

Profil informatică

Teste pentru licență

14-MAR-2003

1 Programare în Pascal

- Un comentariu între acolade:
 - ajută calculatorul să înțeleagă funcția pe care o realizează programul
 - ajută cititorul să înțeleagă funcția pe care o realizează programul
 - trebuie scris după fiecare instrucțiune
 - nu este folositor
- Cuvântul BEGIN este:
 - un identificator al utilizatorului
 - un identificator standard
 - o variabilă
 - un cuvânt rezervat
- La afișarea string-ului 'O'HARA' pe ecran va apare:
 - OHARA
 - O'HARA
 - O'HARA
 - 'O'HARA'
- Care din următoarele aserțiuni sunt adevărate?
 - simbolurile SIN,COS nu pot fi redefinite de utilizator
 - simbolul END poate fi folosit ca identificator
 - simbolul BEGIN poate fi înlocuit cu simbolul START
 - simbolul BEGIN apare cel puțin odată în program
- Numărul $2,3 \times 10^{-5}$ se scrie în PASCAL:
 - 2.310-5
 - 2.3E-5
 - 2.3XE-5
 - 2.3X10-5
- Care din următoarele instrucțiuni sunt corecte pentru expresia:

$$ax^3 + bx^2 - cx + d$$

- `a(x*x*x*)+b(x*x)-c*x+d`
- `a*xxx+b*xx-c*x+d`
- `a*x*x*x+b*x*x-c*x+d`

d) $a*x^3+b*x^2-c*x+d$

7. În instrucțiunea de atribuire: $a:=((b+5) \text{ Div } c) * 13$ ordinea efectuării operațiilor este:
- a) Div * +
 - b) + Div *
 - c) * Div +
 - d) * + Div
8. Funcția TRUNC:
- a) extrage partea fracționară dintr-un număr real
 - b) convertește un număr întreg în număr real
 - c) convertește caracterele în întregi
 - d) convertește un real într-un întreg ignorând partea zecimală
9. O expresie cu operanzi întregi și reali produce:
- a) un rezultat întreg
 - b) un rezultat real
 - c) nu poate fi evaluată
 - d) tipul expresiei depinde de tipul variabilei căreia urmează să i se atribuie
10. A doua operație ce se efectuează în următoarea expresie: $3 \text{ Mod } 2 > (6-2*4) \text{ Div } 7$ este:
- a) Div
 - b) >
 - c) *
 - d) -
11. Care din următoarele constante sunt scrise greșit în PASCAL:
- a) 0005
 - b) -0
 - c) 8,3
 - d) 1EO
12. Indicați care din următoarele aserțiuni sunt adevărate:
- a) tipul INTEGER este identic cu mulțimea Z din matematică
 - b) tipul REAL este identic cu mulțimea R din matematică
 - c) pe tipul REAL sunt permise operațiile: +, -, *, /, Mod, Div
 - d) mulțimea numerelor care pot fi reprezentate cu ajutorul tipului REAL este finită
13. Expresia SQRT(2) în Pascal are:
- a) valoarea 1
 - b) un număr infinit de zecimale
 - c) un număr de zecimale ce depinde de calculatorul utilizat
 - d) o valoare întreagă
14. Care din următoarele aserțiuni sunt adevărate?
- a) valoarea expresiei $77 \text{ Or } 62$ este 139
 - b) valoarea expresiei $77 \text{ Xor } 62$ este 15
 - c) valoarea expresiei $15 \text{ Shl } 2$ este 30

d) valoarea expresiei $21 \text{ Shr } 2$ este 5

15. Expresia matematică:

$$\frac{xy}{z} - \frac{z}{xy}$$

se scrie în PASCAL astfel:

- a) `xy/z-z/xy`
- b) `x*y/z-z/x*y`
- c) `x*y/z-z/x/y`
- d) `xy/z-z/x/y`

16. Expresia matematică:

$$\sqrt[3]{|1 - x^{100}|}$$

se scrie în PASCAL astfel:

- a) `Sqrt (Abs(1-x^100))`
- b) `Sqrt3 (Abs(1-x**100))`
- c) `Abs (1-x**100)**(1/3)`
- d) `Exp (Ln(Abs(1-Exp(100*Ln(x)))))/3`

17. Care este ordinea operațiilor în evaluarea expresiei: $(x-z) \text{ Mod } y + x \text{ Div } y * 3$

- a) - Mod Div * +
- b) - Mod + Div *
- c) Mod - * Div +
- d) - Mod * Div +

18. Care din următoarele aserțiuni sunt adevărate?

- a) tipul `SINGLE` definește un real pe 2 octeți
- b) tipul `REAL` definește un real pe 4 octeți
- c) tipul `DOUBLE` definește un real pe 6 octeți
- d) tipul `EXTENDED` definește un real pe 10 octeți

19. Care din următoarele aserțiuni sunt adevărate?

- a) expresia `Round(-0.5)` are valoarea 0
- b) expresia `-10 Div 3` are valoarea -4
- c) expresia `-10 Mod 3` are valoarea -1
- d) expresia `Sqrt(16)` are tipul întreg

20. Care din următoarele aserțiuni sunt adevărate?

- a) expresia `1*-3` este greșită în Turbo-Pascal
- b) expresia `1*-3` are valoarea -3 în Turbo-Pascal
- c) rezultatul expresiei `Ln(Exp(0))` este de tip întreg
- d) expresia `Trunc (-0.5)` are valoarea -1

21. După execuția secvenței :

```
x :=1;  
y :=2;  
x :=y;  
y :=x;
```

valorile variabilelor x și y vor fi:

- a) $x=2$ $y=1$
 b) $x=1$ $y=2$
 c) $x=2$ $y=2$
 d) $x=1$ $y=1$
22. Dacă x este real și n întreg, specificați care din următoarele instrucțiuni sunt corecte:
 a) $n := n \text{ Div } 2$
 b) $x := x \text{ Div } 2$
 c) $n := n/2$
 d) $n := x \text{ Div } 2$
23. Valoarea lui x după execuția următoarei instrucțiuni: $x := 3+2 \text{ } \leq 35 \text{ Div } 7$ este:
 a) **False**
 b) 5
 c) **True**
 d) 3
24. A treia operație ce se efectuează în următoarea expresie: $(7 \text{ Mod } 2 \text{ } \geq 9 \text{ Div } 8) \text{ And } (3*8 \text{ } \geq 16-5)$ este:
 a) **Div**
 b) \geq
 c) **And**
 d) $*$
25. Care din următoarele aserțiuni sunt adevărate?
 a) expresia $\text{Uppcase}('a')$ are valoarea 'a'
 b) expresia $\text{Sqr}(1) + \text{Sqr}(2) \leq 5$ are valoarea **False**
 c) expresia $'n' = \text{Uppcase}('n')$ are valoarea **False**
 d) expresia $(2 < 0) \text{ or } (1 > 2)$ are valoarea **False**
26. Care din următoarele aserțiuni sunt adevărate?
 a) expresia $\text{Odd}(\text{Round}(10*3.14))$ are valoarea **True**
 b) expresia $\text{True Xor Not False}$ are valoarea **True**
 c) expresia $3*7 \text{ Div } 2 \text{ Mod } 7 \text{ Div } 3 = \text{Round}(\text{Sin}(1))$ are valoarea 1
 d) expresia $\text{Odd}(k) \text{ Or } \text{Odd}(k+1)$ are valoarea **True**
27. Care este ordinea operațiilor în următoarea expresie: $x \text{ Or } y \text{ And Not } z \text{ And } x$
 a) **Or And Not And**
 b) **And Not And Or**
 c) **Not And And Or**
 d) **Not And Or And**
28. Ce tipărește instrucțiunea: $\text{WRITE}(' ' ' ')$
 a) nimic
 b) ''
 c) '
 d) un spațiu
29. Ce tipărește secvența de program:

```
Write ('Suma dintre');  
Write (5:2,' si ',4:2);  
Writeln('este',5+4:2);
```

- a) suma dintre 5 și 4 este 9
- b) suma dintre 5:2 și 4:2 este 5+4:2
- c) suma dintre 5 și 4:2 este 5+4:2
- d) suma dintre 5:2 și 4:2 este 9

30. Care vor fi valorile lui x,y și z după execuția instrucțiunilor:

```
Read(x);  
Readln(y);  
Read(z);
```

dacă la intrare se dau datele:

```
1 2 3 4 5  
6  
7 8
```

- a) 1 2 6
- b) 1 2 3
- c) 1 2 7
- d) 1 6 7

31. Determinați rezultatele următorului program:

```
Program Test;  
Var a,b: Integer;  
Begin  
  Readln(a,b,a);  
  Writeln(a,b,a);  
End.
```

dacă la intrare se vor da valorile 5 2 -3

- a) 5 2 -3
- b) 5 2 5
- c) -3 2 -3
- d) 5 2 -3

32. Ce se va afișa pe ecran în urma execuției secvenței:

```
x:= 5.7;  
Writeln(x);  
Writeln(x:0:2);  
Writeln(x:5:2);
```

- a) 5.7000000000E+00
5.70
5.70
- b) 5.7
5.7
5.7

- c) 5.7
5.7:0:2
5.7:5:2
- d) 5.7
5.7:2
5.7:2

33. Fie declarația: `Var x,y : Integer; Op : Char;` Se dorește a se citi de la intrare o linie cu următoarea structură: `<întreg><spațiu><operație><spațiu><întreg>` unde: `operație` e unul din caracterele '+', '-', '*', sau '/', `<spațiu>` este caracterul ' ' `<întreg>` este un număr întreg. Care din următoarele instrucțiuni realizează acest lucru?

- a) `Readln(x,op,op,y)`
- b) `Readln(x,op,y)`
- c) `Readln(x,op,op,op,y)`
- d) `Readln(x,y,op)`

34. Ce va tipări următorul program:

```
Program Test;  
Var x: Integer; y : Real;  
Begin Readln (x,y);  
Writeln(x*y);  
End.
```

dacă la intrare se furnizează valorile: 3.5 4?

- a) 14.0
- b) 15.0
- c) 12.9
- d) apare eroare la execuție

35. Ce va tipări următorul program?

```
Program Test;  
Var n : Integer;  
Begin n := 1;  
Writeln('n-1'); n := n+2;  
Writeln(n+2);  
End.
```

- a) 0
5
- b) n-1
5
- c) 0
4
- d) n-1
3

36. Următoarele programe intenționează să citească un întreg de 3 cifre și să tipărească răsturnatul lui. Care realizează acest lucru?

- a) `Program Test;`
`Var x,y,z:Integer;`
`Begin`
`Readln(xyz);`
`Writeln(zyx);`
`End.`

- b) Program Test;
 Var x,y,z:Integer;
 Begin
 Readln(x,y,z);
 Writeln(z,y,x);
 End.
- c) Program Test;
 Var x:Integer;
 Begin
 Readln(x);
 Writeln(x Mod 10, x Div 10 Mod 10,x Div 100);
 End.
- d) Program Test;
 Var x:Integer;
 Begin
 Readln(x);
 Writeln(x Div 10 Mod 10, x Mod 10, x Div 100);
 End.

37. Ce va apare pe ecran după execuția secvenței:

```
Write(1,2); Writeln(3);
Writeln(4,5); Write(6);
Writeln ; Writeln(7,8,9);
```

- a) 123
 45
 6
 789
- b) 12
 3
 45
 6
 789
- c) 12
 3
 456
 789
- d) 12
 34
 56
 789

38. Care dintre instrucțiunile următoare este incorectă?

- a) Writeln (a Div b :5)
- b) Writeln (a:4, b:5)
- c) Readln (a/b)
- d) Writeln ('a/b=', a/b)

39. Ce tipărește secvența de program:

```
x:=3
If x>-1 Then Writeln(1)
  Else If x=3 Then Writeln(2)
    Else Writeln(3)
```

- a) 2
- b) 3
- c) 1
- d) 1
- 2

40. Ce va tipări secvența următoare de program:

```
Readln(a,b,c);  
If a=1 Then  
  If b=2 Then  
    If c=3 Then  
      Writeln(3)  
    Else Writeln(2)  
  Else Writeln(1)  
Else Writeln(0)
```

dacă la intrare se dau valorile 1 4 3?

- a) 3
- b) 1
- c) 2
- d) 0

Test	Raspuns
1	b
2	d
3	c
4	d
5	b
6	c
7	b
8	d
9	b
10	c
11	c
12	d
13	c
14	d
15	c
16	d
17	a
18	d
19	c
20	b
21	c
22	a
23	c
24	b
25	c
26	d
27	c
28	c
29	a
30	a
31	c
32	a
33	a
34	d
35	b
36	c
37	a
38	c
39	c
40	b

Test	Raspuns
------	---------

2 Programare în C

1. Care dintre următoarele valori sunt considerate flotante scrise corect?

1) 2307.98 2) +54.3 3) -20.07 4) -198.
5) .13 6) 1.9E4 7) -2.7E-3 8) 2.e+4

- a) 1), 2), 3), 6) și 7)
- b) Toate mai puțin 5)
- c) Toate
- d) Toate mai puțin 8)
- e) Primele cinci

2. Urmăriți secvența de mai jos și precizați valoarea variabilei y ? (Valabila folosind compilatorul pentru C++)

```
int a, b = 3;
int x = 2;
int y = 2 * b - x;
```

- a) 2
- b) 3
- c) 4
- d) 5
- e) Secvența este eronată

3. Care dintre programele următoare sunt eronate?

- a)

```
void main ()
{
;
}
```
- b)

```
void main ()
{
;
};
```
- c)

```
void main (void)
{
;
}
```
- d)

```
void main ();
{
;
}
```
- e)

```
main ()
{
;
}
```

4. Fie programul:

```
#include <iostream.h>
#include <math.h>
#define m 3
void main ()
{
int x, y, z, a, b;
```

```

cin >> x >> y;      //(1)
a = b = 2;          //(2)
z = b * abs(y) - x + m; //(3)
cout<< "Valorile sunt \n" << z << " " << a; //(4)
}

```

Considerand ca la executie se introduc de la tastatura numerele 3 si -6 in aceasta ordine,precizati care dintre afirmatiile de mai jos sunt adevarate.

- Definirea constantei simbolice m este corecta.
- Citirea de tastatura din linia (1) este eronata.
- Atribuirea din linia (2) este eronata
- In urma executiei liniei (3), valoarea variabilei z va fi -12.
- In urma executiei liniei (4) se va afisa:

Valorile sunt 12 2

5. Se consideră variabilele întregi x, y, și z fiind cunoscute valorile x=4 și y=2. Care dintre expresiile de mai jos are valoarea 0?

- $x+y > x\%y+1$
- $z=(x-y!=0)$
- $x-2*y==0$
- $!x$
- $x\&\&y$

6. Fie declarațiile de variabile:

```

int x = 4, y = 13;
float z;

```

Care dintre instrucțiunile de mai jos nu atribuie corect valoarea 8.5 variabilei z?

- $z=(x+y)/2.;$
- $z=((float)x+y)/2;$
- $z=(x+y.)/2;$
- $z=(x+y)/(float)2;$
- $z=(float)(x+y)/2;$

7. Pentru programul următor,precizați care din cele cinci instrucțiuni de afișare sunt eronate:

```

#include <stdio.h>
#include<math.h>
void main ()
{
int a = 73, b = 15;
float x = 1.76;
printf("Am %d kg greutate \nsi %f m inaltime\n", a, x); //(1)
printf("a=%f,x=%f\n", (float)a, x); //(2)
printf("%d %d\n%d", a>0 && b>0, b=2*b-1, abs(a)); //(3)
}

```

- Instrucțiunile (1) și (2), pentru că în parametrul de control nu putem avea texte și secvențe Escape.
- Instrucțiunea (2), pentru că variabila a este de tipul int și nu se poate afișa cu specificatorul %f, iar construcția (float)a este eronată.
- Instrucțiunea (3), pentru că în funcția printf nu putem avea o atribuire, în speță $b=2*b-1$

- d) Instrucțiunea (3), pentru că nu putem afișa o expresie logică (în cazul de față "a>0 && b>0").
e) Nici una din instrucțiuni nu este eronată.

8. Ce afișează programul următor, dacă valoarea citită de la tastatură este 2?

```
#include <stdio.h>
void main()
{
    int x, y, z;
    scanf("%d", &x);
    y = --x;
    y += 3;
    z = x - 2 * y++;
    printf("%d", z++);
}
```

- a) -9
b) -8
c) -7
d) -6
e) -5

9. Ce valoare afișează programul următor?

```
#include <iostream.h>
void main ()
{
    int x = 5, y;
    y = (sizeof(x-1) == sizeof(int)) ? sizeof('x') : sizeof(3);
    cout << y;
}
```

- a) 3
b) 1
c) 2
d) 4
e) Programul este eronat

10. Ce valori va afișa programul următor?

```
#include <stdio.h>
void main ()
{
    int a = 10, b = 6, c = 4, d;
    d = (c = a-6, a = b%c, b += a, a/2);
    printf("\n%d %d %d %d", a, b, c, d);
}
```

- a) 0 16 -6 5
b) 2 8 4 1
c) 4 2 8 1
d) -6 0 16 5
e) Alte valori decât cele prezente

11. Fie instrucțiunile:

```
int a = 34; float x = 6.25;
```

Precizați care dintre instrucțiunile de afișare următoare trebuie executată astfel încât să se afișeze pe ecran rîndul ilustrat în desenul de mai jos.

34 6.250

(prin " " am simbolizat caracterul spațiu).

- a) `printf("\n%4d:%-10f", a, x);`
- b) `printf("\n%-4d:%6.3f", a, x);`
- c) `printf("\n%6d:%10f", a, x);`
- d) `printf("n%-d:%-.3f", a, x);`
- e) `printf("\n%d:%f", a, x);`

12. Dacă la tastatură se introduce caracterul 'a', iar codurile literelor mici sunt succesive, începînd cu 97, ce afișează programul următor?

```
#include <iostream.h>
#include <stdio.h>
void main ()
{
    char c, p;
    p = getchar();
    int n = p+259;
    c = n;
    putchar(c);
}
```

- a) 356
- b) 'a'
- c) 'd'
- d) 100
- e) programul este greșit

13. Ce valoare afișează programul de mai jos?

```
#include <iostream.h>
void main ()
{
    int a = 3, b = 2, n = 4, x;
    x = (a<<n) + (a&b) + (n|b);
    cout << x;
}
```

- a) 2
- b) 8
- c) 51
- d) 56
- e) programul este greșit

14. Care dintre construcțiile de mai jos reprezintă constante caracter?

1) " " 2) '\ ' 3) 'a' 4) '''
5) '\\\ ' 6) '\13' 7) "a" 8) ''

- a) 2), 3) și 8)

- b) toate
- c) toate mai puțin 5) și 6)
- d) 2), 3), 4) și 8)
- e) 3) 4) 5) 6) și 8)

15. Pentru fiecare dintre constantele aflate în coloana A), alegeți din coloana B) tipul său:

Coloana A)	Coloana B)
A1) 5.0	B1)constant"a "intreag"a \
A2) 5	B2)constant"a real"a
A3) '5'	B3)constant"a hexazecimal"a
A4) 05	B4)constant"a octal"a
A5) "5"	B5)constant"a caracter
A6) 0X5	B6)constant"a "sir de caractere

- a) A1→B2, A2→B1, A3→B5, A4→B1, A5→B6, A6→B3
- b) A1→B2, A2→B1, A3→B5, A4→B4, A5→B5, A6→B3
- c) A1→B2, A2→B1, A3→B5, A4→B4, A5→B6, A6→B3
- d) A1→B2, A2→B1, A3→B5, A4→B4, A5→B6, A6→ eronată
- e) A1→B2, A2→B1, A3→B5, A4→B1, A5→B6, A6→ eronată

16. Care dintre liniile de program de mai jos realizează inițializarea corectă a variabilei x la declararea sa?

- a) int==2;
- b) x:int=2;
- c) int x=2;
- d) int x 2;
- e) x=2:int;

17. Definiți o constantă simbolică PI cu valoarea 3.14, folosind directiva-preprocesor "#define".

- a) #define 3.14 PI;
- b) #define PI 3.14;
- c) #define float PI 3.14;
- d) #define PI=3.14;
- e) #define float PI=3.14;

18. Ce valoare afișează programul următor?

```
#include <iostream.h>
#include <math.h>
void main ()
{
    int x = 4, y = -9, z = (x+y)/2, u;
    z--;
    u = (sqrt(abs(x)) + sqrt(abs(y))) / (x+y);
    z -= u%2 - x - y;
    cout << "\n" << z;
}
```

- a) -7
- b) -8
- c) -9
- d) 4

e) 5

19. Care dintre următoarele afirmații sunt adevărate?

- a) Operatorul de atribuire este "==".
- b) Operatorul care realizează "SAU logic" între două expresii este "&&".
- c) "!=" este un operator logic
- d) $a\%b$ reprezintă restul împărțirii întregi a lui a la b.
- e) Într-o expresie, operatorii relaționali se execută înaintea celor aritmetici.

20. Care dintre următoarele expresii au valoarea 1 dacă și numai dacă valorile variabilelor întregi x și y sunt numere pare?

- a) $x-y==2$
- b) $x*y\%4==4$
- c) $(x+y)\%2==1$
- d) $y\%x==2$
- e) $(x\%2==0) \&\& (y\%2==0)$

21. Care dintre următoarele expresii sunt adevărate dacă și numai dacă valorile variabilelor x și y sunt numere naturale consecutive?

- a) $x-y==2$
- b) $(x==1) \&\& (y==2)$
- c) $(x-y==1) \&\& (y-x==1)$
- d) $y==x \pm 1$
- e) $(x-y==1) \|\ (y-x==1)$

22. Pentru programul următor, precizați care dintre afirmațiile de mai jos sunt adevărate:

```
#include <stdio.h>
void main ()
{
    int a,b; float c;
    scanf("%d %d", &a, &b);
    printf("%d\n", !a);          //(1)
    printf("%d\n", b\%2==0 && b>0);  //(2)
    printf("%-8.2f\n", float(2*b+a)); //(3)
    printf("a=%3d*b=%-3d\n", a, b);  //(4)
}
```

- a) Dacă valoarea citită în variabila a este diferită de zero atunci linia (1) afișează valoarea 0.
- b) Dacă valoarea citită în variabila b este pară și pozitivă, atunci linia (2) afișează valoarea 0.
- c) Instrucțiunea (3) afișează corect valoarea expresiei $2*b+a$, pe opt caractere din care două zecimale, cu aliniere la dreapta.
- d) Dacă de la tastatură se introduc valorile 2 și 6 pentru variabilele a, respectiv b, atunci linia (4) va afișa

$$a = 2_{\square} * b =_{\square} 6$$

(prin " \square " am simbolizat caracterul "spatiu").

- e) Nici una dintre afirmațiile anterioare nu este adevărată, deoarece citirea cu funcția scanf este eronată.

23. Funcțiile getchar(), getch() și getche() citesc de la tastatură un caracter. Ce deosebiri există între cele trei funcții?

- a) Funcțiile getchar și getche realizează citirea cu ecou, iar getch citește caracterul fără ecou.
- b) Funcția getchar citește caracterul cu ecou, iar funcțiile getche și getch realizează citirea fără ecou.

- c) Funcțiile getchar și getch preiau caracterul numai după apăsarea tastei ENTER.
- d) Funcțiile getchar și getch preiau caracterul de îndată ce a fost tastat, fără să mai aștepte "confirmarea" ENTER.
- e) Toate cele trei funcții au prototipul în header-ul conio.h
24. Știind că în standardul ASCII caracterele literă mare au codurile succesive începând cu 65 ('A' ← 65, 'B' ← 66, 'C' ← 67, etc), deduceți ce valoare va afișa programul următor.

```
#include <stdio.h>
void main()
{
    int x , y, z, p; char m, n;
    m = 'C'; n = 'A';
    x = m; y = 2*m-n; z = 3;
    p = x<y ? (y<z?z:y) : (z<x?x:z);
    printf("\n%d", p);
}
```

- a) 1
- b) 3
- c) 69
- d) 67
- e) 0
25. Presupunem că rulăm programul următor sub o versiune a limbajului C++, în care valorile de tipul int se memorează pe doi octeți, iar cele de tipul float pe patru octeți. De câte ori va afișa programul valoarea 2?

```
#include <iostream.h>
#include <stdio.h>
void main(void)
{
    int x; char c;
    cout << "\n";
    x = 'A'; cout << sizeof(x);
    c = 'A'; printf("%d", sizeof(c));
    printf("%d", sizeof(float)-2);
    x = sizeof(int); x = ++x/2; cout << (x==2);
}
```

- a) nici o dată
- b) o dată
- c) de două ori
- d) de trei ori
- e) de patru ori
26. Ce valori afișează programul următor?

```
#include <stdio.h>
void main ()
{
    int x = 10, y = 6, m, n, p;
    n = (m = x++, y++, p = x+y);
    printf("\n%d %d %d", m, n, p);
}
```

- a) 10 18 16

- b) 11 18 18
- c) 10 18 18
- d) 11 18 17
- e) 10 18 17

27. În programul următor, care dintre secvențele de instrucțiuni notate cu (1), (2) și (3) nu vor produce erori la execuție?

```
#include <stdio.h>
void main ()
{
    int x, y; char m;
    m = 'A'; putchar(m+1); // (1)
    x = getchar(); m = 'A'; x -= m; printf("%c %d ", x, x); // (2)
    x = 300; m = x; y = m; printf("\n%c %d\n", y, y); // (3)
}
```

- a) Numai secvențele (1) și (3)
- b) Toate trei
- c) Numai secvența (1)
- d) Numai secvențele (1) și (2)
- e) Nici una

28. Precizați valoarea lui n, rezultată în urma execuției programului:

```
#include <iostream.h>
void main()
{
    char c; int n = 97;
    n = c = n;
}
```

- a) Programul este greșit
- b) 97
- c) 99
- d) 79
- e) NULL

29. Precizați valoarea pe care o va avea variabila c în urma execuției programului de mai jos:

```
#include <iostream.h>
void main()
{
    char c = 'd';
    int n = 99;
    c = n+1 = c-1;
}
```

- a) 'd'
- b) 'c'
- c) 'b'
- d) NULL
- e) atribuirea este greșită

30. În urma execuției secvenței de program alăturate, pentru care dintre tripletele de valori ale variabilelor a,b,c, date mai jos, se va afișa valoarea 1?

```
x=1;
if (!(a<=b) || !(a<=c))
{
    if (b>=c) printf("%d\n",-x);
}
else
    if (b<c) printf("%d\n",x);
```

- a) a=3, b=2, c=4
- b) a=2, b=3, c=4
- c) a=4, b=3, c=2
- d) a=2, b=4, c=3
- e) a=4, b=2, c=3

31. Se consideră secvența de mai jos, în care x este o variabilă de tipul “caracter”, iar m este de tip întreg.

```
if (x=='a') m=0; else if ((x=='b')||(x=='c')) m=1; else
if ((x>'c')&&(x<'z')) m=0; else if (x=='z') m=1; else m=-1;
```

Care dintre cele trei secvențe S1 S2 S3 de mai jos este echivalenta cu secvența dată?

```
//Secvența S1
if (x=='a') m=0;
else
    if (x=='b' || x=='c') m=1;
    else
        if (x>'c' && x<'z') m=0;
        else
            if (x=='z') m=1;
            else m=-1;
```

```
//Secvența S2
if (x=='b' || x=='c' || x=='z' ) m=1;
else
    if ((x=='a') || (x>'c' && x<'z')) m=0;
    else
        m=-1;
```

```
//Secvența S3
if (x<'a' || x>'z') m=-1;
else
    if (x=='b' || x=='c' || x=='z') m=1;
    else m=0;
```

- a) numai S1
- b) numai S2
- c) numai S3
- d) toate
- e) nici una

32. Ce va afișa programul următor, dacă de la tastatură se introduc în ordine numerele 5, 7 și 8?

```

#include <iostream.h>
void main()
{
    int x, y, z, m;
    cin >> x >> y >> z;
    m = (x+y+z) / 3;
    switch (m) {
        case 1, 2, 3, 4:
            { cout << "Corigent";
              break; }
        case 5, 6:
            { cout << "Mediocru";
              break; }
        case 7, 8, 9:
            { cout << "Bine";
              break; }
        case 10:
            { cout << "Foarte bine";
              break; }
        default:
            cout << "EROARE !";
    }
}

```

- a) 'Corigent'
- b) 'Mediocru'
- c) 'Satisfăcător'
- d) 'Foarte bine'
- e) 'EROARE !'

33. Precizați ce va afișa în urma execuției secvenței de program de mai jos pentru $n=5$ (S, n și k sunt variabile întregi)

```

S = 0; k = 1;
while (k <= n)
{
    S += k;
    k += 2;
}
printf("S=%d", S);

```

- a) S=4
- b) S=16
- c) S=9
- d) S=15
- e) S=0

34. Care dintre secvențele de program de mai jos calculează corect factorialul numărului natural n ? (Reamintim: $n! = 1 \cdot 2 \cdot \dots \cdot n$)

```

//Secventa S1
p = 1;
for(i = 1; i <= n; i++)
    p = p * i;

//Secventa S2
p = 1; i = 1;

```

```

while (i <= n)
    p = p * i++;

//Secventa S3
p = 1; i = 1;
do
{
    p *= i; i = i + 1;
}while (i <= n);

```

- a) numai S1
- b) numai S2
- c) numai S3
- d) S1 și S3
- e) toate

35. Se consideră programul următor:

```

#include <iostream.h>
void main(0)
{
    int a, b, c, d, i;
    cin >> a >> b;
    if (a > b)
    {
        c = a; a = b; b = c;
    }
    d = 0;
    for (i = a; i < b; i++)
        if (i%2 == 0) d++;
    cout << d;
}

```

Ce valoare se afișează, conform algoritmului dat, pentru a=33 și b=18?

- a) 8
- b) 7
- c) 0
- d) 16
- e) 33

36. Se consideră secvența de program de mai jos, în care toate variabilele sunt întregi. Pentru n=3, care va fi valoarea variabilei p după execuția secvenței?

```

p = 1;
for (i = 1; i <= n; i++)
{
    S = 0;
    for (j = 1; j <= i; j++)
        S += j;
    p *= S;
}

```

- a) 180
- b) 18
- c) 9

d) 216

e) 1

37. Precizați ce va afiș în urma execuției programului următor, presupunând că valoarea lui x, care se citește de la tastatură, este 179.

```
#include <stdio.h>
void main(void)
{
    int c, S;
    long d, x;
    scanf("%ld", &x);
    d = x; S = 0;
    while (d)
    {
        c = d%10; S += c; d = d/10;
    }
    printf("%d", S);
}
```

a) 16

b) 18

c) 17

d) 0

e) 971

38. Fie o variabilă x de tip întreg, diferită de 0 și 1. În urma execuției secvențelor de program de mai jos, valoarea variabilei ok ar trebui să fie: 1, dacă numărul x este prim, respectiv, 0, în caz contrar (variabilele ok și x sunt de tip întreg, iar valoarea lui x se presupune citită anterior). Care dintre ele funcționează corect în sensul realizării efectului dorit?

a) ok = 1;
for (i = 2; i < x; i++)
if (x % i == 0)
ok = 0;
printf("%d", ok);

b) ok = 1;
for (i = 2; i < x; i++)
if (x % i == 0)
ok = 0;
else
ok = 1;

c) ok = 0;
for (i = 2; i < x; i++)
if (x % i == 0)
ok = 1;

d) toate

e) nici una

39. Considerăm programul următor:

```
#include <stdio.h>
#include <conio.h>
void main()
{
    short int m,x;
    m = -1;
```

```

while ((scanf("%d", &x) == 1) && x)
    if (x > m) m = x;
printf("%d", m);
}

```

Precizați ce valoare va afișa programul, dacă șirul de numere citit de la tastatură în timpul execuției acestuia este 2,5,-32000,33000,0.

- a) -1
- b) 0
- c) 33000
- d) 2
- e) 5

40. Dacă de la tastatură se introduce numărul 22, câte valori distincte va afișa programul următor?

```

#include <iostream.h>
#include <math.h>
void main(void)
{
    int x, n, i;
    for (cin >> n, i = 1; ; x = sqrt(i), cout << x, i++)
        if (i > n) break;
}

```

- a) nici una
- b) una
- c) două
- d) trei
- e) patru

41. Fie secvența de program următoare, care calculează și afișează produsul p al primelor n numere naturale $1*2*\dots*n$ (unde valoarea lui n este cunoscută):

```

// Secventa S)
p = 1;
for (i = 1; i <= n; i++)
    p = p * i;
cout << p;

```

Pentru cele doua secvente S1) și S2) de mai jos, precizati care dintre afirmatiile urmatoare sunt corecte.

```

Secventa S1)
for (p = 1, i = 1; i <= n; p *= i++);
    cout << p;

```

```

Secventa S2)
i = 1; p = 1;
while(i <= n ? p *= i++ : 0);
    cout << p;

```

- a) Ambele secvențe S1) și S2) sunt corecte sintactic dar nici una nu este echivalentă cu S).
- b) Ambele secvențe S1) și S2) sunt corecte sintactic și echivalente cu S).
- c) Ambele secvențe S1) și S2) sunt corecte sintactic, dar numai secvența S1) este echivalentă cu S).
- d) Secvența S1) conține erori, iar secvența S2) este corectă sintactic și echivalentă cu S).
- e) Ambele secvențe S1) și S2) sunt eronate.

42. Precizați de câte ori se va afișa valoarea 1 în timpul execuției programului următor, dacă prin citirea de la tastatură variabilele primesc valorile a=3, b=4, x=5.

```
#include<iostream.h>
void main()
{
    int a, b, x;

    cin >> a >> b >> x;
    if (!(x<=a) && (x>=b)) cout << 1 << "\n";
    if (!(x<=a || x>=b)) cout << 1 << "\n";
    if (!(x<=a) && !(x>=b)) cout << 1 << "\n";
    if (!(x<=a) || !(x>=b)) cout << 1 << "\n";
}
```

- a) niciodată
b) o dată
c) de două ori
d) de trei ori
e) de patru ori
43. Dacă în timpul execuției programului de mai jos n va primi prin citire de la tastatură valoarea 232213 care vor fi în final valorile variabilelor f1,f2 și f3?

```
#include<iostream.h>
void main()
{
    long n;
    unsigned int f1, f2, f3, c;

    cin >> n;
    f1 = f2 = f3 = 0;
    do
    {
        c = n % 10;
        n = n / 10;
        switch (c) {
            case 1: {f1++; break;}
            case 2: {f2++; break;}
            case 3: {f3++; break;}
        }
    }while (n != 0);
    cout << f1 << f2 << f3;
}
```

- a) f1=1, f2=1, f3=1
b) f1=1, f2=2, f3=2
c) f1=1, f2=3, f3=2
d) f1=2, f2=1, f3=3
e) f1=3, f2=2, f3=1
44. Deduceți ce valoare va afișa în urma execuției secvenței de program de mai jos, dacă valorile variabilei x citite de tastatură sunt în ordine 3, 2, 4, 3, 5, 10, 20, 0.

```
cin >> x;
nr = 0;
```

```

do
{
    y = x;
    cin >> x;
    if (x == 2 * y)
        nr++;
}while (x != 0);
cout << nr;

```

- a) 0
- b) 1
- c) 2
- d) 3
- e) 4

45. Care va fi valoarea variabilei c afișată de către programul următor, dacă de la tastatură se citesc valorile a = b = 3?

```

#include<iostream.h>
void main()
{
    long a, b, c, z, i;

    cout << "Dati a si b: ";
    cin >> a >> b;
    c = 0; z = 1;
    for (i = 1; i <= a; i++)
    {
        c += z; z *= b;
    }
    cout << "c=" << c;
}

```

- a) 8
- b) 32
- c) 27
- d) 13
- e) 1

46. Care dintre secvențele de program S1,S2,S3, date mai jos este echivalentă cu secvența următoare? (Două secvențe de program se consideră echivalente, dacă produc același efect în orice situație). Toate variabilele folosite sunt întregi.

```

p = 1;
for (i = 1; i <= n; i++)
{
    S = 0;
    for (j = 1; j <= i; j++)
        S = S + j;
    p = p * S;
}

```

```

//=====
//Secventa S1
//=====
p = 1; S = 0;

```



```

for (i = 1; i <=n; i++)
{
    p = p * i; S = S + p;
}

//=====
//Secventa S2
//=====
p = 1; S = 0;
for (i = 1; i <= n; i++)
{
    S = S + i; p = p * S;
}

//=====
//Secven"ta S3
//=====
p = 1; S = 0;
for (i = 1; i <= n; i++)
{
    p = p * S; S = S + i;
}

```

- a) numai S1
- b) numai S2
- c) numai S3
- d) toate
- e) nici una

47. Se consideră secvențele de program de mai jos. Pentru $n=4$ precizați care dintre secvențe afișează, în urma execuției, șirul de numere: 1 2 2 3 3 3 4 4 4 4.

- a)

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= n; j++)
        printf("%2d", i);
```
- b)

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= i; j++)
        printf("%2d", i);
```
- c)

```
for (i = 1; i <= n; i++)
    for(j = 1; j <= n; j++)
        printf("%2d", j);
```
- d)

```
for (i = 1; i <= n; i++)
    for (j = 1; j <= i; j++)
        printf("%2d", j);
```
- e)

```
for (j = 1; j <= n; j++)
    for (i = 1; i <= n; i++)
        printf("%2d", i);
```

48. Pentru afișarea numerelor naturale $1, 2, \dots, n$ (unde n se presupune cunoscut), propunem următoarele două secvențe de program:

S1)

```
for (i = 1; i <= n; i++)
    printf("%2d", i);
```

S2)

```
for (i = 1; i <= n; i++);
    printf("%2d", i);
```

Care dintre afirmatiile de mai jos este adevarata?

- a) Nici una dintre secvențe nu îndeplinește cerința problemei.
- b) Ambele secvențe îndeplinesc cerința problemei.
- c) Numai secvența S2 conține erori de sintaxă.
- d) Numai secvența S1 îndeplinește cerința problemei.
- e) Numai secvența S2 îndeplinește cerința problemei

49. Considerînd că toate variabilele sunt întregi, ce valoare se afișează după execuția secvenței de mai jos?

```
s = 0; t = 0; x = 3;
i = 1; y = 1; z = 1;
do
{
    if (x>0)
        if (y>1)
            if (z>2) t = x;
            else t = x + y;
            else t = x + y + z;
    s += i + t;
    i++;
}while (i > 7);
printf("%d", s);
```

- a) 1
- b) 5
- c) 6
- d) 51
- e) 63

50. Pentru ce valoare a variabilei M, secvența de program de mai jos reprezintă o buclă infinită?

```
int n = 10, m;
do
{
    while (n>0) n--;
}while (n != m);
```

- a) 10
- b) orice valoare diferită de 10
- c) 0
- d) orice valoare diferită de 0
- e) pentru orice valoare întregă

51. Definim oglinditul unui număr natural x ca fiind numărul obținut prin citirea numărului x în ordine inversă (de la dreapta la stînga).

Exemplu: Oglinditul numărului 1534 este 4351. Care dintre programele de mai jos afișează corect oglinditul lui x?

```
//Programul P1
#include <stdio.h>
void main()
{
    int c;
    long d, x, y;
```

```

scanf("%ld", &x);
d = x; y = 0;
while (d)
{
    c = d % 10;
    y = y + c * 10;
    d = d / 10;
}
printf("%ld", y);
}

```

```

//Programul P2
#include <stdio.h>
void main()
{
    int c;
    long d, x, y;

    scanf("%ld", &x);
    d = x; y = 0;
    do
    {
        c = d % 10;
        y = y + c * 10;
        d = d / 10;
    }while (d);
    printf("%ld", y);
}

```

```

//Programul P3
#include <stdio.h>
void main()
{
    int c;
    long d, x, y;

    scanf("%ld", &x);
    d = x; y = 0;
    while (d)
    {
        c = d % 10;
        y = y * 10 + c;
        d = d / 10;
    }
    printf("%ld", y);
}

```

```

//Programul P4
#include <stdio.h>
void main()
{
    int c;
    long d, x, y;

    scanf("%ld", &x);
    d = x; y = 0;
    do
    {

```

```

    c = d % 10;
    y = y * 10 + c;
    d = d / 10;
}while (d);
printf("%ld", y);
}

```

- a) P1 și P2
- b) P1 și P3
- c) P2 și P4
- d) P1 și P4
- e) P3 și P4

52. Precizați ce valori se vor afișa, în ordine, în timpul execuției programului următor:

```

#include <iostream.h>
void main()
{
    int x, y, m, n, a, b = 5;

    x = (m = b = n = 3, b + 4);
    y = a = ((b == 5) ? b-- : --b);

    if (!(y == a) && (m == n)) cout << 1; else cout << 0;
    if (!a && b && !m) cout << 1; else cout << 0;
    if ((n = (a + b--)) == -(m - x)) cout << 1; else cout << 0;
}

```

- a) 0,0,0
- b) 0,0,1
- c) 0,1,1
- d) 1,1,1
- e) 1,0,1

53. Ce valoare va afișa programul următor pentru n=12?

```

#include <iostream.h>
void main ()
{
    int i, n, S;

    cin >> n;
    for (S = 0, i = 2; i < n / 2; !(n % i) ? S += i++ : i++);
    cout << S;
}

```

- a) 0
- b) 9
- c) 12
- d) 78
- e) programul conține erori

54. Pentru programul următor care dintre afirmațiile de mai jos sunt adevărate?

```

#include <iostream.h>
void main ()
{
    int S, x;

    for (S = 0, x = 1; ; S += x, cin >> x)
        if (!x) break;
    cout << S;
}

```

- Dacă de la tastatură se introduc, în ordine, numerele 2,3,4 și 5 atunci programul va afișa suma numerelor citite, adică 14.
- Dacă prima valoare introdusă de la tastatură este 0, atunci ciclul se încheie și se afișează valoarea 1.
- Ciclul este eronat. Nu se poate face o citire în linia for.
- Instrucțiunea if este eronată.
- Din cauză că lipsește expresia care dă condiția de continuare ciclul for se va executa la infinit.

55. Dacă de la tastatură se introduc, în ordine numerele 2,7,3,8,5,5, ce valoare va afișa secvența următoare?

```

int a, b, nr = 0;
do
{
    cin >> a >> b;
}while ((b != a) ? ++nr : 0);
cout << nr;

```

- 0
- 1
- 2
- 3
- 4

56. Care dintre secvențele de mai jos afișează corect șirul cifrelor zecimale impare 97531 în această ordine?

- for (i = 9; i >= 1; i--)
cout << i--,i++;
- for (i = 0; i <= 9; i++)
cout << 9 - i;
- for (i = 9; i-- >= 1;)
cout << i,i--;
- i = 10;
while (i--)
cout << --i;
- i = 1;
do
{
cout << 10 - i++;
}while (i <= 9 ? i++ : 0);

57. Care dintre următoarele variante reprezintă o declarare corectă a unei variabile x de tipul "adresă a unei variabile de tip întreg"?

- int x*;
- int *x;
- int x;

- d) `int &x;`
- e) `int x&;`

58. Fie declarațiile de variabile:

```
int a = 2, b;  
int *x, *y;
```

Precizați ce valori se vor afișa, în ordine, în urma execuției secvenței de program de mai jos.

```
x = &a;  
a = 5;  
cout << *x;  
b = a-2;  
y = &b;  
b += (*y) + 4;  
cout << b;  
*y = *x;  
cout << *y;  
if (x == y)  
    cout << 1;  
else  
    cout << 0;
```

- a) 2,10,2,1
- b) 2,10,2,0
- c) 5,7,5,0
- d) 5,10,5,0
- e) 5,10,5,1

59. Se consideră următoarea secvență de program:

```
int *q, **p, a = 5, b = 3;  
*p = &a; // (1)  
q = *p; // (2)  
b += *(&(**p)); // (3)  
cout << *q << b;
```

Ce puteți spune despre atribuiri (1),(2) și (3)?

- a) Nici una dintre atribuiri nu este corectă.
- b) Numai atribuirea (1) este corectă.
- c) Numai atribuiri (1) și (2) sunt corecte.
- d) Toate cele trei atribuiri sunt corecte, iar secvența afișează de două ori numărul 5.
- e) Toate cele trei atribuiri sunt corecte, iar secvența afișează numerele 5 și 8.

60. Fie atribuirea: `*y = &(*(&z))`; Cum trebuie scrise corect declarațiile de variabile, astfel încât atribuirea să fie corectă?

- a) `int *y, z;`
- b) `int y, *z;`
- c) `int y, **z;`
- d) `int **y, z;`
- e) `int **y, *z;`

61. Fie programul următor:

```

#include <iostream.h>
#include <stdlib.h>

void main()
{
    int a, *x;

    x = (int*)malloc(sizeof(int));
    a = 2; *x = 3;
    int &s1 = a;
    int *&s2 = x;
    int &s3 = 5;
    .....
    free(x);
}

```

In loc de "...", trebuie scrisa o linie de program, care sa afiseze numerele 2, 3 si 5, pe acelasi rind, separate prin cite un spatiu. Care este acesta?

- a) cout << "\n " << s1 << " " << s2 << " " << &s3;
- b) cout << "\n " << s1 << " " << *s2 << " " << s3;
- c) cout << "\n " << &s1 << " " << s2 << " " << &s3;
- d) cout << "\n " << *s1 << " " << s2 << " " << *s3;
- e) Programul conține erori în instrucțiunile de atribuire.

62. Care dintre instrucțiunile (I), (II), (III), (IV) din programul următor sunt eronate?

```

#include <stdio.h>
void main()
{
    const int x = 3;
    int u, v;

    x = 4;                // (I)
    *(int*)&x = 8;        // (II)
    u = x;                // (III)
    v = *(int*)&x;        // (IV)
}

```

- a) I
- b) II
- c) III
- d) IV
- e) nici una

63. Fie declarația de variabile:

```

int a = 2, b, c = 5;
int *x, *y;

```

Precizați ce valori se vor afișa, în ordine, în urma executiei secvenței de program de mai jos.

```

x = &c;
a += *x;
cout << a;
b = ++a;
y = &b;
cout << *y;
x = y;
cout << (*x)++;

```

- a) 7,7,7
- b) 7,8,9
- c) 7,8,8
- d) 7,7,8
- e) 8,8,9

64. Fie declarațiile de variabile:

```
int a, b, c;
int *x, *y, *z;
```

Precizați ce valori se vor afișa, în ordine, în urma execuției secvenței de program de mai jos.

```
a = 3;
y = &c;
*y = a++;
z = &a;
x = y;
cout << *x++ ;
if (*x == *y) cout << 1; else cout << 0;
*z = *y;
if (z == y) cout << 1; else cout << 0;
```

- a) 5,1,0
- b) 4,1,0
- c) 4,0,0
- d) 3,0,0
- e) 3,0,1

65. Fie un pointer x către un întreg. Care dintre instrucțiunile de mai jos realizează corect alocarea dinamică a memoriei?

- a) `x = (int)malloc(sizeof(int*));`
- b) `x = (int*)malloc(sizeof(int*));`
- c) `x = (int*)malloc(sizeof(int));`
- d) `*x = (int*)malloc(sizeof(int));`
- e) `*x = (int)malloc(sizeof(int*));`

66. Programul următor își propune să afișeze jumătate din produsul a două numere citite, referite prin intermediul pointerilor a și b. Ce erori întâlnim în acest program?

```
#include <stdio.h>
#include <stdlib.h>
void main ()
{
    int *a, *b;
    float x;

    a = (int*)malloc(sizeof(int));
    b = (int*)malloc(sizeof(int));
    scanf("%d %d", &a, &b);
    x = *a * *b/2;
    printf("\n%f", x);
    free(a); free(b);
}
```


- a) Declarațiile de variabile sunt eronate.
- b) Alocarea dinamică a memoriei este eronată
- c) Citirea de la tastatură nu se face corect.
- d) Atribuirea nu calculează valoarea dorită
- e) Instrucțiunea de afișare a rezultatului este eronată.

67. Pentru secvența de program următoare, care dintre afirmațiile de mai jos sunt adevărate?

```
#include <iostream.h>
void main()
{
    int x, y;

    x = y = 0;
    int &n = 2;
    for (int &i = x; i < n; i++)
        cout << y++;
}
```

- a) Linia for este eronată, deoarece controlul ciclului nu poate fi un pointer
- b) Instrucțiunea `int &i=x` este eronată
- c) Secvența este eronată, deoarece într-o instrucțiune for nu se poate scrie o declarație de sinonimie.
- d) Secvența este corectă și vor afișa valorile 0 și 1
- e) Secvența este corectă și vor afișa valorile 1 și 2.

68. Considerăm declarația: `int **p`; și atribuirea `p=&q`; Alegeți declarația potrivită astfel, încât atribuirea să aibă sens.

- a) `int q`;
- b) `int *q`;
- c) `int ***q`;
- d) `int &q`;
- e) nici una

69. Precizați valoarea variabilei a ca urmare a execuției programului următor:

```
void main()
{
    int a;
    char b = 1;

    a = *(int*)&b;
}
```

- a) 1
- b) 97
- c) neprecizată
- d) o adresă
- e) programul este greșit

70. Precizați care dintre instrucțiunile de atribuire de mai jos face ca x să primească valoarea 0:

```

void main ()
{
    int a = 1, b = 2;
    float x;

    x = a / *&b;    //(I)
    x = (float) a / b; //(II)
}

```

- a) I
- b) II
- c) ambele
- d) niciuna
- e) programul este greșit

71. Precizați valoarea pe care o va avea variabila a, ca urmare a execuției instrucțiunilor următoare:

```

int a, &b = a;
b = 1;

```

- a) nedefinită
- b) 0
- c) 1
- d) declarația este eronată
- e) dimensiunea tipului int

72. Care dintre instrucțiunile de tipărire vor afișa aceeași valoare:

```

#include <stdio.h>
void main()
{
    int a = 2, *p = &a;

    printf("%d\n", *p+1);
    printf("%d\n", *&p+1);
    printf("%d\n", *(p+1));
    printf("%d\n", *(&p+1));
}

```

- a) prima și a doua
- b) a doua și a treia
- c) a doua și a patru
- d) nici una
- e) programul este eronat

73. Care dintre tripletele de numere întregi date ca variante de răspuns pot fi introduse de la tastatură la execuția programului următor, astfel încât acesta să afișeze valoarea 1?

```

#include <stdio.h>
void main ()
{
    int a, b, c, m;
    const int x = 0;

    scanf("%d %d %d", &a, &b, &c);
    m = a<b ? a : b;
    printf("\n%d, x = m>c);
}

```

- a) 1,2,3
- b) 1,3,2
- c) 2,3,1
- d) 1,1,1
- e) Instrucțiunea de afișare este eronată, din cauza atribuirii din funcția printf

74. Care dintre variantele de mai jos reprezintă o declarație corectă a unui vector v cu 20 de elemente numere întregi?

- a) `v[20] : integer;`
- b) `v[20] int;`
- c) `int v[20];`
- d) `int : v[20];`
- e) `integer v[20];`

75. Câte erori conține programul de mai jos?

```
void main()
{
    int k, n;                //Linia (1)
    int [n];                //Linia (2)
    n = 4                    //Linia (3)
    for (k = 0; k < n; k++) //Linia (4)
        v[k] = 0;           //Linia (5)
}
```

- a) nici una
- b) una
- c) două
- d) trei
- e) patru

76. Deduceți care vor fi elementele vectorului v după execuția secvenței de program următoare.

```
int n, k, x, v[7] = {5,14,-3,8,-1};
```

```
n = 5; x = v[0];
for (k = 1; k < n; k++)
    v[k - 1] = v[k];
v[n - 1] = x;
```

- a) (-1,5,14,-3,8,0,0)
- b) (14,-3,8,-1,0,0,5)
- c) (14,-3,8,-1,5,0,0)
- d) (0,0,5,-3,14,-1,8)
- e) (0,0,-1,14,-3,8,5)

77. Câte elemente ale vectorului v vor avea valoarea 9 după execuția programului de mai jos?

```
#include <iostream.h>
void main()
{
    int v[] = {0,1,2,0,4,5,6};
    int i = 0, x = 9;

    do {
        v[i++] = x;
    }while (i < 6 && v[i]);
}
```

- a) nici unul
- b) unul
- c) două
- d) trei
- e) toate

78. Se consideră o matrice a cu n linii * n coloane și un vector v cu n elemente. Precizați care vor fi elementele vectorului v , după execuția secvenței următoare:

```
int nr, n, i, j, x, v[3];
int a[3][3] = {{7,1,7},{-7,7,0},{2,4,11}};
```

```
n = 3; x = 7;
for (i = 0; i < n; i++)
{
    nr = 0;
    for (j = 0; j < n; j++)
        if (a[i][j] == x) nr++;
    v[i] = nr;
}
```

- a) Nedefinite
- b) $v=(0,0,0)$
- c) $v=(1,2,3)$
- d) $v=(2,0,1)$
- e) $v=(2,1,0)$

79. Se consideră secvența de program următoare, în care a este o matrice cu n linii * n coloane și elemente numere întregi, iar x este o variabilă de tip întreg.

```
x = 1;
for (i = 1; i <= n; i++)
    for (j = 0; j <= i - 1; j++)
        if (a[i][j] != 0)
            x = 0;
```

În urma execuției secvenței, valoarea variabilei x va fi 1, dacă:

- a) Deasupra diagonalei principale există cel puțin un element egal cu zero.
- b) Toate elementele de deasupra diagonalei principale sunt zero.
- c) Toate elementele de sub diagonala principală sunt diferite de zero.
- d) Toate elementele de sub diagonala principală sunt zero.
- e) Sub diagonala principală există cel puțin un element diferit de zero.

80. Precizați care vor fi elementele matricii a după execuția secvenței de mai jos, pentru $x=1, m=4, n=3$ și

$$a = \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$

```
for(i = x + 1; i <= m-1; i++)
    for(j = 0; j <= n-1; j++)
        a[i - 1][j] = a[i][j];
```

- a)
$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \end{pmatrix}$$
- b)
$$\begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 10 & 11 & 12 \end{pmatrix}$$
- c)
$$\begin{pmatrix} 4 & 5 & 6 \\ 7 & 8 & 9 \\ 10 & 11 & 12 \\ 10 & 11 & 12 \end{pmatrix}$$
- d)
$$\begin{pmatrix} 1 & 3 & 3 \\ 4 & 6 & 6 \\ 7 & 9 & 9 \\ 10 & 12 & 12 \end{pmatrix}$$
- e)
$$\begin{pmatrix} 2 & 3 & 3 \\ 5 & 6 & 6 \\ 8 & 9 & 9 \\ 11 & 12 & 12 \end{pmatrix}$$

81. Fie următorul program:

```
#include <iostream.h>
void main ()
{
    int v[20], i, n, E;

    cin >> n;
    for (i = 0; i < n; i++)
        v[i] = i%2 ? i : -i;
    for (E = 1, i = 0; i < n; E *= v[i++]);
        E++;
    cout << E ;
}
```

În urma execuției sale sunt posibile următoarele situații:

- Expresia condițională din primul ciclu for este eronată din punct de vedere sintactic.
- Dacă variabila n primește prin citire valoarea 6, atunci elementele vectorului v vor fi, în ordine(0,1,-2,3,-4,5).
- Prezența caracterului “;” după al doilea ciclu for constituie o eroare.
- Dacă variabila n primește prin citire valoarea 5, atunci programul afișează 32
- Programul funcționează corect pentru orice valoare întreagă a lui n mai mică sau egală cu MAXINT.

82. Ce va afișa programul următor:

```
#include <iostream.h>
void main()
{
```

```

int (*v)[3]; int u[] = {10, 11, 12};

v = &u;
cout << (*v)[1];
}

```

- Programul este eronat
- O adresă de memorie oarecare, fără nici o semnificație.
- Valoarea întregă 11.
- Adresa de memorie la care este depozitată valoarea întregă 11.
- Adresa începînd cu care se găsește vectorul v în memorie.

83. Care dintre secvențele de program de mai jos afișează corect produsul elementelor pare ale unui vector $v=(v[0], v[1], \dots, v[n])$ cu n elemente variabile de tip întreg?

- ```

p = 1;
for (i = 1; i <= n; i++)
 if (v[i] % 2 == 0)
 p = p * v[i];
cout << p;

```
- ```

p = 1;
for (i = 0; i < n; i++)
    if (v[i] / 2 == 0)
        p = p * v[i];
cout << p;

```
- ```

p = 0;
for (i = 0; i < n; i++)
 if (v[i] % 2 != 0)
 cout << p;

```
- ```

p = 1;
for (i = 0; i < n; i++)
    if (v[i] % 2 == 0)
        p = p * v[i];
cout << p;

```
- ```

p = 0;
for (i = 0; i < n; i++)
 if (v[i] % 2 == 0)
 p = p * v[i];
cout << p;

```

84. Ce valoare va fi afișată în urma execuției programului următor?

```

#include <iostream.h>
void main()
{
 int v[] = {0, 1, 2, 0, 4, 5, 6};
 int i = 0, nr = 0;

 do {
 if (i == v[i]) nr++;
 }while (i < 6 && v[i++]);
 cout << nr;
}

```

- 0
- 1

- c) 3
- d) 5
- e) programul va intra într-un ciclu infinit

85. Deduceți care vor fi în ordine, de la stînga la dreapta elementele nenule ale vectorului a la sfîrșitul execuției secvenței de program următoare:

```
int i, j = 0;
int v[7] = {0, 2, 7, 3, 4, 8, 5};
int a[7] = {0, 0, 0, 0, 0, 0, 0};

for (i = 0; i < 7; i++)
 if ((v[i] % 2 == 0) && (i % 2 != 0))
 {
 a[j] = v[i]; j++;
 }
```

- a) 2,4,8
- b) 7,5
- c) 2,8
- d) 2,3,8
- e) 7,3,5

86. Se consideră secvența următoare, în care valorile lui n și x se presupun cunoscute, iar este un vector cu elementele (v[0], v[1], ..., v[n-1])

```
p = n;
for (i = 0; i < n; i++)
 if (v[i] == x)
 p = i;
for (i = p + 1; i < n; i++)
 v[i - 1] = v[i];
for (i = 0; i < n - 1; i++)
 printf("%2d", v[i]);
```

Precizați care dintre următoarele afirmații sunt adevărate:

- a) Pentru n=5, x=3 și v=(5,6,2,7,1), se afișează ultimele patru elemente nemodificate ale vectorului: 6 2 7 1.
- b) Pentru n=5, x=1 și v=(2,1,3,1,4), se afișează 2 3 1 4;
- c) Secvența conține erori de sintaxă.
- d) Algoritmul șterge un vector elementul cu valoarea x, prin mutarea cu o poziție mai la dreapta a elementelor aflate înaintea lui.
- e) Algoritmul șterge din vector elementul cu valoarea x, prin mutarea cu o poziție mai la stînga a elementelor aflate după el.

87. Declarați o matrice a cu 15 linii\* 10 coloane și elemente de tip întreg.

- a) `int a[15][10];`
- b) `int a[15,10];`
- c) `a[15][10] int;`
- d) `int a[10][15];`
- e) `a[10,15] int;`

88. În secvența de program de mai jos (anterior acesteia se consideră inclus header-ul values.h), a este o matrice pătratică cu n linii și coloane iar vectorul v are n elemente. Toate celelalte variabile sunt întregi. Atît elementele matricei cît și cele ale vectorului se presupun citite anterior.

```

for (i = 0; i < n; i++)
{
 x = -MAXINT;
 for (j = 0; j < n; j++)
 if ((v[j] != -MAXINT) && (v[j] > x))
 {
 x = v[j];
 p = j;
 }
 a[i][p] = v[p];
 v[p] = -MAXINT;
}

```

Precizați care dintre următoarele afirmații sunt false:

- a) Dacă  $n=3$ , vectorul  $v$  este  $(0,0,0)$  (are toate elementele nule) iar matricea este

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

atunci, după execuția secvenței, toate elementele deasupra diagonalei principale în matrice vor fi 0.

- b) Dacă  $n=3$ , vectorul  $v$  este  $(-1,-1,-1)$  iar matricea este

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

atunci după execuția secvenței, toate elementele de pe diagonala secundară în matrice rămân nemodificate.

- c) Dacă  $n=3$ , vectorul  $v$  este  $(-6,11,10)$ , iar matricea este

$$\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$$

atunci, după execuția secvenței, matricea  $a$  va avea elementele

$$\begin{pmatrix} 1 & 11 & 3 \\ 4 & 5 & 10 \\ -6 & 8 & 9 \end{pmatrix}$$

- d) Indiferent care ar fi inițial, elementele vectorului  $v$  și ale matricei  $a$ , după execuția secvenței, toate elementele vectorului vor avea valoarea  $-MAXINT$ .
- e) Secvența plasează elementele vectorului pe rând în matrice, vizitându-le în vector în ordine descrescătoare.

89. Precizați ce valoare va afișa programul următor:

```

#include<stdio.h>
void main()
{
 int x = 12;
 int *a[10], *b = a[2];

 b = &x;
 printf("%d", *a[2]);
}

```

- a) Adresa lui  $b$   
b) Adresa lui  $a[2]$



- c) 12
- d) NULL
- e) Programul are erori de sintaxă

90. Fie vectorul `y` cu patru elemente numere întregi: `int y[4] = {0, 1, 2, 3}`; Care dintre următoarele instrucțiuni declară și inițializează corect un pointer `ptr` către vectorul `y`?

- a) `int *(ptr[4]) = &y`
- b) `int (ptr*)[4] = &y`
- c) `int (*ptr)[4] = &y`
- d) `int ptr*[4] = &y`
- e) `int *ptr[4] = &y`

91. De câte ori va afișa valoarea 30 programul următor?

```
#include <iostream.h>
void main()
{
 int *a[5], v[5], i = 0, x = 30;

 while (i++ < 5) v[i] = x / i;
 for (i = 1; i < 5; i++)
 *(a+i) = &v[i];
 a[0] = &x;
 for (i = 0; i < 5; i++)
 cout << "\n" << *a[i];
}
```

- a) nici o dată
- b) o dată
- c) de două ori
- d) de trei ori
- e) Programul generează eroarea "Divide error" din cauza unei împărțiri la 0.

92. Ce valoare va afișa programul următor?

```
#include <iostream.h>
void main()
{
 int a[20][20], i, j, n = 4;

 for (i = 0; i < n; i++)
 for (j = 0; j < n; j++)
 *((a+i)+j) = (i>j) ? (j-i) : (j+i);

 int m = 10;
 for (i = 0; i < n; i++)
 for(j = 0; j < n; j++)
 if (m > (*(a+i))[j])
 m = a[i][j];
 cout << m << "\n";
}
```

- a) 10
- b) 6
- c) 3

- d) -3
- e) programul este eronat

93. Fie următorul program:

```
#include <iostream.h>
void main()
{
 int x[4] = {1, 2, 3}, y[4] = {4, 5, 6, 7}, z[7], i, j;
 for (i = 0; i < 4; i++)
 *(z + i) = *(y + i);
 for (j = 0; j < 3; j++)
 *(z + i + j) = *(x + j);
 for (i = 0; i < 7; i++)
 cout << * (z+i);
}
```

Care vor fi valorile afișate în urma execuției sale?

- a) 1,2,3,4,5,6,7
  - b) 7,6,5,4,3,2,1
  - c) 3,2,1,7,6,5,4
  - d) 4,5,6,7,1,2,3
  - e) programul este eronat
94. Câte erori conține programul următor?

```
#include <iostream.h>
void main()
{
 char a[5] = "abcd";
 char b[10] = "efghij";

 b[3] = '$';
 a = "klm";
 b = a;
}
```

- a) nici una
  - b) una
  - c) două
  - d) trei
  - e) patru
95. În programul următor, care dintre secvențele de instrucțiuni (I), (II), (III) realizează corect citirea unui șir de caractere de la tastatură și afișarea acestuia?

```
#include <iostream.h>
#include <stdio.h>
void main()
{
 char s1[10],s2[10],s3[10];

 cin >> s1; cout << s1; // (I)
 scanf("%s", s2); printf("s2=%s", s2); // (II)
 scanf("%s", &s3); printf("%s", s3[10]); // (III)
}
```

- a) numai (I)
- b) numai (II)
- c) (I) și (II)
- d) (I) și (III)
- e) toate

96. Analizati programul următor și alegeți răspunsul corect:

```
#include <iostream.h>
void main()
{
 char b[11], a[11] = "abcdefghij";
 int i = 0;

 while (a[i]%2)
 b[i++] = a[i];
 b[i] = 0;
}
```

Observație: Literele mici au codurile ASCII succesive, începînd cu 97 ('a'← 97, 'b'← 98, etc.).

- a) Programul are erori.
- b) Șirul b conține numai caracterul 'a'.
- c) În urma execuției, șirurile a și b coincid.
- d) Șirul b conține numai caracterele din șirul a ale căror coduri ASCII sunt numere pare.
- e) Șirul b conține numai caracterele de rang par din a (al doilea, al patrulea etc.).

97. Cîte erori conține corpul funcției main din programul de mai jos?

```
#include <stdio.h>
void main()
{
 char v[2][10] = {"Andrei", "Vasilescu"};

 printf("%s", v[1]);
 printf("%s", v[1][2]);
 printf("%s", v[0]+v[1]);
}
```

- a) nici una
- b) una
- c) două
- d) trei
- e) patru

98. Care dintre variantele de mai jos reprezintă o declarație corectă a unui șir de caractere?

- a) `char s[20];`
- b) `char *s[20];`
- c) `char *s;`
- d) `char s;`
- e) `char s[];`

99. Pentru programul următor, analizați corectitudinea afirmațiilor de mai jos:

```

#include <stdlib.h>
#include <string.h>
#include <iostream.h>
void main()
{
 char s1[4], s2[4];
 long x;

 cin >> s1 >> s2;
 if (strcmp(s1, s2) < 0) x = atol(s1);
 else if (strcmp(s1, s2) == 0) x = 0;
 else x = atol(s2);
 cout << x;
}

```

- a) Condițiile din cele două linii if sunt greșite.
- b) Apelurile funcției atol sunt corecte.
- c) Dacă de la tastatură se introduc șirurile "98" și "123" atunci se va afișa 98.
- d) Dacă de la tastatură se introduc șirurile "123" și "121", atunci programul va afișa șirul "123".
- e) Dacă de la tastatură se introduc șirurile "ab" și "ac", atunci se va semnala un mesaj de eroare.

100. Știind că în conformitate cu standardul ASCII, codurile literelor mari sunt succesive începînd cu 65, ce va afișa programul de mai jos?

```

#include <stdlib.h>
#include <string.h>
#include <iostream.h>

void main()
{
 int x = 20, e;
 char s[15] = "ABC", t[15], u[15];

 e = s[1] + s[2];
 itoa(e, t, 10);
 strcpy(u, t);
 strcat(s, u);
 cout << s;
}

```

- a) Nimic, șirul s fiind vid
- b) ABC13
- c) AB13
- d) ABC133
- e) ABC131

101. Ce șir de caractere va afișa secvența următoare?

```

char *s = "abcdefg", *ptr;
ptr = s;
ptr += 4;
cout << ptr;

```

- a) "fg"
- b) "efg"
- c) "defg"

- d) "cdefg"
- e) secvența este eronată

102. Ce va afișa programul următor?

```
#include <stdio.h>
void main()
{
 char *a[3] = {"abc", "def", "ghi"};
 char *p = &a[0][0];

 printf("%s %c %c", a[1], a[2][1], *(p+5));
}
```

- a) abc d NULL
- b) abc d e
- c) def h NULL
- d) def h e
- e) Programul va semnaliza eroare de compilare.

103. Ce va afișa cea de-a doua instrucțiune de tipărire din programul de mai jos:

```
#include <stdio.h>
void main()
{
 char a[12] = "abcdefghij";
 char *p = a;
 int i;

 for (i = 0; i < 12; i++) *(p + i) = a[i]^i;
 printf("%s\n", p);
 for(i = 0; i < 12; i++) *(a + i) = p[i]^i;
 printf("%s", p);
}
```

- a) Nimic
- b) Textul "abcdefghij"
- c) Textul "jihgfedcba"
- d) O succesiune de caractere imprevizibilă
- e) Programul contine erori

104. Care dintre secvențele de program de mai jos afișează corect textul "C++"?

- a) `char s[4]="C++";  
printf("%s", s);`
- b) `char s[3]="C++";  
cout << s;`
- c) `char s = "C++";  
puts(s);`
- d) `char s[4];  
printf("%s", s="C++");`
- e) `char s[3];  
cout << (s="C++");`

105. Care dintre instrucțiunile programului de mai jos sunt eronate?

```

#include <stdio.h>
#include <string.h>

void main()
{
 char a[10], b[10]; int k; //(1)
 scanf("%s %s", a, b); //(2)
 k = strlen(a) / 2; //(3)
 a[k] = '*'; //(4)
 printf("%d", strlen(a)<strlen(b)); //(5)
 b = a; //(6)
}

```

- a) Declarațiile de variabile din linia (1).
- b) Citirea șirurilor din linia (2).
- c) Atribuirile din liniile (3) și (4).
- d) Afișarea din linia (5).
- e) Atribuirea din linia (6).

106. Precizați ce șir de caractere se va afișa în urma execuției programului următor:

```

#include <iostream.h>
#include <string.h>

void main()
{
 char s[20] = "BorLanD C++ 3.1";
 int i;

 for (i = 0; i < strlen(s); i++)
 if((s[i] >= 'A') && (s[i] <= 'Z')) s[i] -= ('A'-'a');
 cout << s;
}

```

- a) "BorLanD C++ 3.1"
- b) "bORlANd c++ 3.1"
- c) "BORLAND C++ 3.1"
- d) "borland c++ 3.1"
- e) "Borland C++ 3.1"

107. Se consideră un șir a definit ca un vector de caractere. Care dintre cele două secvențe S1) și S2) de mai jos afișează prima literă mare din șir?

```

S1)
i = 0;
while (a[i] && (a[i]<'A' || a[i]>'Z'))
 i++;
printf("%c", a[i]);

```

```

S2)
i = 0;
do{
 i++;
}while (a[i] && !(a[i]>='A' && a[i]<='Z'));
putchar(a[i]);

```

- a) numai S1)

- b) numai S2)
- c) nici una
- d) ambele
- e) secvențele conțin erori.

108. Precizați ce text va tipări programul de mai jos:

```
#include <stdio.h>

void main()
{
 char v[3][100] = {"Popescu","Ionescu","Vasilescu"};

 if (v[0] < v[1]) printf("%s", v[0]);
 else printf("%s", v[1]);
}
```

- a) Popescu
- b) Ionescu
- c) Vasilescu
- d) Nimic
- e) Programul conține erori.

109. Care dintre cele trei instrucțiuni printf ale funcției main de mai jos tipăresc succesiunea de caractere "bd"?

```
#include <stdio.h>

void main()
{
 char s[6][2] = {"ab", "ac", "ad", "bc", "bd", "cd"};

 printf("%c%c", s[3][0], s[2][1]);
 printf("%s", s[3][0]+s[2][1]);
 printf("%s", s[5]);
}
```

- a) Toate
- b) Numai prima
- c) Numai primele două
- d) Numai prima și a treia
- e) Nici una

110. Ce va afișa programul de mai jos?

```
#include <iostream.h>

void main()
{
 char s[10] = "AB6X92P3M", b[10];
 int i = 0, k = 0;

 while (s[i])
 {
 if (i%2) b[k] = s[i];
 i++;
 }
}
```

```

 b[k] = 0;
 cout << b;
}

```

- a) BX23
- b) A69PM
- c) B
- d) 3
- e) nimic

111. Fie s și t două variabile de tipul “vector de caractere”. Scrieți o secvență de program pentru enunțul “dacă șirul s conține cel puțin n caractere, atunci copiază în t primele n caractere ale lui s”.

- a) if (strlen(s) >= n) strncpy(s, t, n); t[n] = 0;
- b) if (strlen(s) >= n) strncpy(t, s, n); t[n] = 0;
- c) if (strlen(s) >= n) strncpy(n, s, t); t[n] = 0;
- d) if (n <= strlen(s)) strcpy(s, t, n); t[n] = '0';
- e) if (strlen(s) >= n) strcpy(t, s, n); t[n] = '0';

112. Ce va afișa programul următor?

```

#include <stdlib.h>
#include <string.h>
#include <iostream.h>

void main ()
{
 char s[12] = "6789", t[12] = "6", u[12] = "89";
 long x = 0;

 strcat(t, u);
 if (strcmp(s, t)) x = atol(t);
 else x = atol(s);
 if (strcmp(s,u)>0) x = atol(u);
 cout << "\n" << x;
}

```

- a) 0.
- b) 6.
- c) 89.
- d) 689.
- e) 6789.

113. Ce text afișează programul următor?

```

#include <stdio.h>

void main()
{
 char *s1 = "EXEMPLU SIMPLU ", *s2 = "SIMPLU";

 printf("\n%.8s%.6s", s1, s2);
}

```

- a) “EXEMPLU”
- b) “EXEMPLU SIMPLU”



- c) "EXEMPLU SIMPLU SIMPLU"
- d) "EXEMPLUSIMPLU"
- e) "SIMPLU"

114. Ce afișează programul următor?

```
#include <stdio.h>

void main()
{
 char *s = "123456789", *t, *u;

 u = &s[4], s += 3, t = &s[1];
 printf("%d%d%d", u == s, u == t, s == t);
}
```

- a) 000
- b) 001
- c) 010
- d) 100
- e) 111

115. Care dintre instrucțiunile (1), (2),..., (5) din funcția main de mai jos sunt eronate?

```
#include <stdio.h>
#include <string.h>
#include <iostream.h>

void main()
{
 char *s1, *s2, *s3;
 int x;

 s1 = "test"; // (1)
 scanf("%s", s2); // (2)
 s3 = &s1; // (3)
 cout << s1+s2; // (4)
 x = strlen(*s2); // (5)
}
```

- a) (2), (3) și (4)
- b) (2), (3), (4) și (5)
- c) (4) și (5)
- d) (3) și (5)
- e) (3), (4) și (5)

116. Fie programul:

```
#include <iostream.h>

void main()
{
 char s[20], *t, *u;
 int i, x;

 cin >> s;
 for (x = 0, i = 0; s[i]; t = &s[i], u = t + 1, u[0]==t[0] ? x=1 : 0, i++);
 cout << x;
}
```

În urma executiei programului, se va afisa valoarea 0 daca:

- a) Toate caracterele șirului s sunt identice.
- b) În șirul s există cel puțin două caractere succesive diferite.
- c) În șirul s există cel mult două caractere succesive identice.
- d) În șirul s există cel puțin două caractere succesive identice
- e) În șirul s nu există două caractere succesive identice

117. Considerăm următoarele noțiuni:

- A) vector de doi pointeri catre caracter
- B) pointer catre sir de doua caractere

si urmatoarele declaratii de variabile:

- I) `char *a[2];`
- II) `char (*b)[2];`

Precizati corespondentele corecte:

- a) A) cu I) și B) cu II).
- b) A) cu II) și B) cu I)
- c) Nu există corespondențe.
- d) B) nu are corespondent
- e) Cele două declarații semnifică același lucru

118. Ce afișează programul de mai jos?

```
#include <iostream.h>

void main()
{
 char *s[5] = {"012", "345", "678", "9AB", "CDE"};
 char *t, *u; int i;

 t = &s[1][0];
 cout << (*(t+5)==s[2][1]);
 u = &s[3][0] + 1;
 i = 0;
 while (u[i]) cout << u[i++];
}
```

- a) 178
- b) 1AB
- c) 078
- d) 0AB
- e) 067

119. Ce afișează programul de mai jos?

```
#include <string.h>
#include <iostream.h>

void main()
{
 char *s[10] = {"10", "00", "10", "10", "01", "11"};
```

```

char *t = "10";
int i = 0, j = i-1;

while (s[i])
{
 if (!strcmp(s[i],t)) j = i;
 i++;
}
cout << j;
}

```

- a) -1
- b) 0
- c) 1
- d) 3
- e) 4

120. Cîte erori conține programul următor:

```

void main()
{
 char *(a[4][6]);
 char b;

 a[2][3] = &(b + 2);
 a[3][2] = &b + 3;
 *(4 + a[2]) = &b + 1;
 *a[1][3] = b + 3;
}

```

- a) nici una
- b) una
- c) două
- d) trei
- e) patru

121. Se consideră un fișier definit prin pointerul f, și care conține următoarele valori pe primele două rînduri:

```

4 7 2.5 -6.23
8

```

Fie următoarele declarații de variabile:

```

FILE *f;
int x, y;
float a, b, d;
char c;

```

Care dintre secvențele de instrucțiuni de mai jos trebuie executate astfel încît toate variabilele declarate să primească valori citite din fișierul f?

- a) `fscanf(f, "%d %f %d %f\n", &x, &a, &y, &b);`  
`fscanf(f, "%c %f", &c, &d);`
- b) `fscanf(f, "%d %d %f %f\n", &x, &y, &a, &b);`  
`fscanf(f, "%c %f", &c, &d);`
- c) `fscanf(f, "%d %d %f %f\n", &x, &y, &b, &a);`  
`fscanf(f, "%f %c", &d, &c);`

- d) `fscanf("%f %f %d %d\n", &b, &a, &y, &x, f);`  
`fscanf("%c %f", &c, &d, f);`
- e) `fscanf("%d %f %d %f\n", &x, &a, &y, &b, f);`  
`fscanf("%f %c", &d, &c, f);`

122. În timpul execuției programului următor sunt posibile următoarele situații:

```
#include <stdio.h>

void main()
{
 FILE *f;
 int x = 1, S = 0;

 f = fopen("suma.txt", "r");
 while (!feof(f) && x)
 {
 fscanf(f, "%d", &x);
 if (x%2) S += x;
 }
 fclose(f);
 printf("\nS=%d", S);
}
```

- a) Programul este corect sintactic
- b) Pentru a funcționa citirea din fișier, acesta trebuie deschis în alt mod.
- c) Programul va intra într-un ciclu infinit.
- d) Dacă în fișier se găsesc, pe același rând separate prin câte un spațiu, numerele 2 5 4 3 6 1 0 7, atunci programul va afișa S=16.
- e) Modul în care este închis fișierul nu corespunde cu modul în care a fost deschis.

123. Fie fișierul identificat prin descriptorul f, avînd următorul conținut:

```
5
2 3 4 6 7 8
```

Care dintre secvențele următoare de program S1, S2 și S3 poate fi executată, astfel, încît în vectorul v să se citească corect toate numerele din fișier?

```
//Secventa S1
fscanf(f, "%d, &n);
for (i = 0; i < n; i++)
 fscanf(f, "%d", &v[i]);

//Secventa S2
i = 0;
while(!feof(f))
{
 fscanf(f, "%d", &v[i]); i++;
}
n = i;

//Secventa S3
i = 0;
do{
 fscanf(f, "%d", &v[i]); i++;
}while(!feof(f));
n = i-1;
```

- a) toate
- b) nici una
- c) numai S1 și S2
- d) numai S2 și S3
- e) numai S1

124. Ce număr se va găsi pe al patrulea rând al fișierului "4.txt" după execuția programului următor?

```
#include <stdio.h>

void main()
{
 FILE *f;
 int n = 8, i = 0, v[8] = {1, 3, 8, 5, 0, 6, 7, 4};

 f = fopen("4.txt", "w");
 while (v[i] % 2)
 i++;
 while (i < n)
 {
 if (v[i]) fprintf(f, "%d\n", v[i]);
 i++;
 }
 fclose(f);
}
```

- a) 5
- b) 0
- c) 6
- d) 7
- e) 4

125. Precizați care va fi conținutul fișierului g după execuția programului următor, dacă fișierul f conține textul ilustrat mai jos:

```
#include <stdio.h>
#include <math.h>

void main()
{
 FILE *f, *g; int i = 1;
 char s[11], c1, c2;

 f = fopen("7.txt", "r");
 g = fopen("7_2.txt", "w");
 while (i++ < 4) fgets(s, 10, f);
 fprintf(g, "%d ", i-1);
 fputs(s, g);
 fclose(f); fclose(g);
}
```

Textul din fișierul fișierul f:

Luni  
Marti  
Miercuri

Joi  
Vineri  
Sambata  
Duminica

- a) 3 Miercuri
- b) 3 Joi
- c) 4 Miercuri
- d) 4 Joi
- e) 5 Joi

126. Ce valori va afișa programul următor, dacă fișierul f arată ca mai jos?

1 11 4 12 8 9 5

(prin " " am notat caracterul "spațiu")

```
#include <stdio.h>

void main()
{
 FILE *f; int a, b, c, x, p;

 f = fopen("valori.txt", "r");
 fscanf(f, "%d %d %d", &a, &b, &c);
 p = ftell(f);
 fseek(f, 7, 1);
 fscanf(f, "%d", &x);
 printf("\n%d %d %d", p, !feof(f), x);
}
```

- a) 6 0 9
- b) 6 1 -9
- c) 7 1 9
- d) 7 0 -9
- e) 6 1 5

127. Fie un fișier identificat prin descriptorul f și deschis cu atributul "w". Fie de asemenea, două variabile întregi x și y, ale căror valori sunt cunoscute. Care dintre instrucțiunile de mai jos pot fi executate astfel încât valorile celor două variabile să fie scrise în fișier fiecare pe alt rând?

- a) fprintf(f, "%d\n%d", x, y);
- b) fprintf("%d\n%d", x, y, f);
- c) fprintf(f, "\n%d%d\n", x, y);
- d) fprintf("\n%d%d\n", x, y, f);
- e) fprintf("\n%d%d\n", x, y, \*f);

128. Fie fișierul cu descriptorul f cu următorul conținut:

50 2  
1.4 3  
8

Ce va afișa pe ecran în urma execuției programul următor?

```

#include <stdio.h>
void main ()
{
 FILE *f;
 int a, b;
 float x, y;

 f = fopen("intrare.txt", "r");
 fscanf(f, "%d %f\n", &a, &x);
 fscanf(f, "%f\n", &y);
 fscanf(f, "%d", &b);
 printf("\na=%d x=%f y=%f b=%d", a, x, y, b);
 fclose(f);
}

```

- a) a=50 x=2.000000 y=1.400000 b=8
- b) a=50 x=2.000000 y=1.400000 b=3
- c) a=50.000000 x=2.000000 y=1.4 b=3.000000
- d) a=50.000000 x=2.000000 y=1.4 b=8.000000
- e) Citirile din fișier vor eșua.

129. Fie programul următor:

```

#include <stdio.h>

void main()
{
 FILE *f, *g;
 int a, x, S;

 f = fopen("in.txt", "r");
 g = fopen("out.txt", "w");
 scanf("%d", &a);
 while (!feof(f))
 {
 S = 0;
 while (S < a && !feof(f))
 {
 fscanf(f, "%d", &x); S += x;
 }
 fprintf(g, "%d ", S);
 }
 fclose(f); fclose(g);
 printf("\nS=%d", S);
}

```

Dacă de la tastatură se introduce valoarea 10, iar conținutul fișierului 'in.txt' este cel de mai jos, câte numere va scrie programul în fișierul 'out.txt'?

4 6 3 2 6 15 1

- a) nici unul
- b) unul
- c) două
- d) trei
- e) patru

130. Cite numere se vor găsi în fișierul "nr.txt" după execuția programului următor?

```
#include <stdio.h>

void main()
{
 int v[9] = {0, 1, 0, 0, 2, 3, 0, 4, 5}, i;
 FILE *f;

 f = fopen("nr.txt", "w");
 i = 0;
 while (i < 9)
 {
 while (i < 9 && v[i])
 {
 fprintf(f, "%3d", v[i]); i++;
 }
 fprintf(f, "%3d", 99);
 i++;
 }
 fclose(f);
}
```

- a) 4
- b) 5
- c) 8
- d) 9
- e) 10

131. Presupunem că pe hard-disk există fișierul "car.txt" cu următorul conținut:

AbCdEfGhIj

Știind că în conformitate cu standardul ASCII literele mari au coduri succesive începând cu 65 ('A'← 65,'B'← 66, etc.), iar literele mici au coduri succesive începând cu 97 ('a'← 97,'b'← 98, etc.), precizați care va fi conținutul fișierului "car2.txt" după execuția programului următor:

```
#include <stdio.h>

void main()
{
 char c1, c2;
 FILE *f, *g;

 f = fopen("car.txt", "r");
 g = fopen("car2.txt", "w");
 c1 = 96;
 while (!feof(f))
 {
 c2 = fgetc(f);
 printf("%c", c2);
 if (c2 > c1) fputc(c2 - 32, g);
 c1 = c2;
 }
 fclose(f); fclose(g);
}
```

- a) bdfhj



- b) BDFHJ
- c) acegi
- d) ACEGI
- e) abcdefghij

132. Deduceți ce valoare va afișa programul următor, presupunând că în fișierul cu descriptorul f se găsesc pe rând, separate prin spații, numerele 1 3 0 0 2 -3 0 -4 -1.

```
#include <stdio.h>
#include <math.h>

void main()
{
 FILE *f;
 int s = 1, i = 0, a[20];

 f = fopen("numere.txt", "r");
 while (!feof(f))
 {
 i++;
 fscanf(f, "%d ", &a[i]);
 if (a[i]) s *= abs(a[i]);
 }
 printf("\n%d", s);
 fclose(f);
}
```

- a) 1
- b) 72
- c) -72
- d) Programul conține erori de sintaxă
- e) Nu se pot citi corect numerele din fișier

133. Deduceți ce valoare va afișa programul de mai jos, dacă fișierul text are următorul conținut:

Conținutul fișierului:

```
3 3
1 2 3
4 5 6
7 8 9
```

Programul sursa:

```
#include <stdio.h>

void main()
{
 FILE *f;
 int i, j, m, n, S = 0, a[20][20];

 f = fopen("continut.txt", "r");
 fscanf(f, "%d %d", &m, &n);
 for (i = 0; i < m; i++)
 {
 for (j = 0; j < n; j++)
```

```

 {
 fscanf(f, "%d ", &a[i][j]);
 if ((i + j) % 2) S += a[i][j];
 }
}
fclose(f);
printf("%d", S);
}

```

- a) 0
- b) 8
- c) 20
- d) 25
- e) programul este eronat

134. Precizați ce nume se va găsi pe al cincilea rând în fișierul "propus.txt", după execuția programului de mai jos?

```

#include <stdio.h>
#include <string.h>

void main()
{
 FILE *f;
 int i = 0, j, k;
 char *aux;
 char *a[9] = {"Marius", "Claudiu", "3rei-Sud-Est", "Daniel", "Vasile",
 "Dan", "Sinacdu", "2Pac"};

 while (a[i]) i++;
 for (j = 0; j < i - 1; j++)
 for(k = j + 1; k < i; k++)
 if (strcmp(a[j], a[k]) > 0)
 {
 aux = a[j]; a[j] = a[k]; a[k] = aux;
 }
 k = 0; f = fopen("propus.txt", "w");
 while (a[k])
 fprintf(f, "%s\n", a[k++]);
 fclose(f);
}

```

- a) 2Pac
- b) Claudiu
- c) Dan
- d) Daniel
- e) Marius

135. Precizați care dintre următoarele linii de program reprezintă corect, din punct de vedere sintactic, lista de parametri și valoarea returnată de o funcție cu numele test, dacă cerem ca parametrii formali să fie două variabile de tip întreg, x,y și o variabilă de tip real z, și să returneze un rezultat întreg.

- a) int test (int x,y,float z)
- b) int test (int x,y;float z)
- c) int test (int x,int y,float z)
- d) test (int x;int y;float z) int

e) test (int x,int y,float z) int

136. Deduceți șirul de valori care se afișează în urma execuției programului de mai jos:

```
#include <stdio.h>

int i, j, k;

int test(int x, int y)
{
 return (x - y);
}

void calcul(int p, int q)
{
 int u, v;

 u = p - i; v = q + j;
 i = test(u, q);
 j = test(v, p);
}

void main()
{
 i = 2; j = 3;
 calcul(i, j);
 printf("%3d%3d", i, j);
 calcul(j, i);
 printf("%3d%3d\n",i, j);
}
```

- a) 2 3 2 3
- b) 3 2 3 2
- c) 2 3 -3 4
- d) -3 4 4 -3
- e) -3 4 10 -3

137. Câte erori va produce execuția programului următor?

```
#include <stdio.h>
#include <math.h>

int calcul(int p, int q)
{
 return(sqrt(p * q)); //(2)
}

void main()
{
 printf("%d", calcul(1, 2)); //(1)
}
```

- a) Una
- b) Două
- c) Trei
- d) Patru
- e) Nici una

138. Ce valori va afișa programul următor?

```

#include <iostream.h>

int n, m;

void T(int n, int &m)
{
 n += 2; m--;
}

void main()
{
 n = 2;
 m = 5;
 T(n, m);
 cout << n << " ";
 cout << m << " ";
 n = 10; m = 20;
 T(n, m); cout << n << " ";
 cout << m << " ";
}

```

- a) 4 4 12 19
- b) 4 5 12 20
- c) 2 4 10 19
- d) 2 5 10 20
- e) 7 2 22 10

139. Precizați ce se va afișa în urma execuției programului de mai jos:

```

#include<stdio.h>

int v[2];
int i;

void test (int x[], int k)
{
 x[k] = 0;
}

void main()
{
 v[0] = 1; v[1] = 2;
 for (i = 0; i <= 1; i++)
 {
 test(v, i);
 printf("%2d%2d", v[0], v[1]);
 }
}

```

- a) 0 2 0 0
- b) 0 2 1 2
- c) 1 2 1 0
- d) 1 2 1 2
- e) 1 2 2 2

140. Fie programul:

```

#include <iostream.h>

int i,x;

int test(int, int);
void main()
{
 x = 2;
 cout << test(x, x + 3); //(I)
 cout << x; //(II)
}
int test(int a, int b)
{
 int x = a + b;

 for (i = 1; i <= x; i++)
 if (i>a && i<b)
 return i;
 return 0;
}

```

Precizati care dintre afirmatiile de mai jos sunt adevarate:

- a) Instrucțiunea (I) afișează valoarea 3.
- b) Instrucțiunea (II) afișează valoarea 7.
- c) Funcția test returnează 0, indiferent care ar fi parametrii actuali dați la apel.
- d) Corpul funcției test este eronat, deoarece conține două instrucțiuni return.
- e) Programul este eronat, deoarece variabila x a fost declarată de două ori

141. Deduceți ce valori va afișa, în ordine, programul următor?

```

#include <iostream.h>

void calcul(int *u, int *v)
{
 for (int i = 0; i < 3; i++)
 {
 v[i] = 1 - u[i]; u[i] = 0;
 }
}
void main()
{
 int u[5], v[5], i;

 for (i = 0; i < 3; i++)
 {
 u[i] = i % 2; v[i] = 0;
 }
 calcul(u, v);
 for (i = 0; i < 3; i++) cout << " " << u[i];
 for (i = 0; i < 3; i++) cout << " " << v[i];
}

```

- a) 1 0 1 0 0 0
- b) 0 0 0 1 0 1
- c) 1 0 1 0 1 0
- d) 0 1 0 1 0 1

e) 0 1 0 0 0 0

142. Se consideră programul următor:

```
#include <stdio.h>

int f(int a = 1, int b = 2, int c = 3)
{
 return a+b+c;
}

void main()
{
 int m, n, p, q;
 m = f(); n = f(10); p = f(12, 19); q = f(11, 23, 18);
 printf("\n%5d%5d%5d%5d", m, n, p, q);
}
```

Precizați valoarea de adevăr a următoarelor afirmații:

- a) Toate apelurile funcției sunt eronate.
- b) Singurul apel corect al funcției este ultimul, iar valoarea lui q va fi 52.
- c) Singurul apel corect al funcției este primul, iar valoarea lui m va fi 6
- d) Toate apelurile sunt corecte și toate variabilele m,n,p,q vor avea valoarea 6
- e) Toate apelurile sunt corecte, iar valorile variabilelor m,n,p,q vor fi m=6,n=15,p=34,q=52.

143. Se consideră programul următor:

```
int f()
{
 static int a = 1;
 return a++;
}

void main()
{
 int m;
 m = f() + f();
 cout << m;
}
```

În urma executiei programului, variabila m va avea valoarea:

- a) 0
- b) 2
- c) 3
- d) 4
- e) 5

144. Ce valori se vor afișa în urma execuției programului următor?

```
#include <stdio.h>

int f1(int *p)
{
 *p = 43;
 return 1;
}

int *f2(int *q)
```

```

{
 int m = f1(q);
 return &m;
}
void main()
{
 int x, y;
 y = *f2(&x);
 printf("\n%d %d", x, y);
}

```

- a) programul este eronat
- b) 0 și 0
- c) 43 și 43
- d) 1 și 43
- e) 43 și 1

145. Ce valoare se va afișa în urma execuției programului următor?

```

#include <iostream.h>

int f(int n,...)
{
 int *x = &n + 1, m = x[0], i;

 for (i = 1; i < n; i++)
 x[i]>m ? m = x[i] : m = m;
 return m;
}
void main()
{
 cout << f(4, 12, 11, 23, 2);
}

```

- a) 12
- b) 11
- c) 23
- d) 2
- e) antetul funcției este greșit

146. Deduceți ce numere va afișa programul următor.

```

#include <iostream.h>

int& f(int a, int b, int &c)
{
 c = a + b;
 return c;
}
void main()
{
 int x = 2, y = 3, z = 11;
 cout << f(x, y, z);
 cout << z;
 f(x, y, z) = 9;
 cout << z;
}

```

- a) 5,11,5
- b) 5,11,9
- c) 5,11,11
- d) 11,5,9
- e) 5,5,9

147. Considerăm o funcție demo, de tipul void (nu returnează nimic), pentru care se definesc ca parametri trei variabile întregi. Cum realizăm apelul funcției, astfel încât la apel să dăm ca parametri variabilele întregi a,b și c?

- a) demo(int a, int b,int c)
- b) demo(int a; int b;int c)
- c) demo(a,b,c)
- d) demo(a;b;c)
- e) demo(int a, b, c)

148. Deduceți șirul de valori care se afișează în urma execuției programului următor.

```
#include <stdio.h>

int s, f;
int sf(int a)
{
 f -= a;
 return (a * a);
}
void main()
{
 f = 10; s = sf(f);
 printf("%6d%6d", s, f);
 f = 10; s = sf(10); s *= sf(f);
 printf("%6d%6d", s, f);
 f = 10; s = sf(f); s *= sf(10);
 printf("%6d%6d", s, f);
}
```

- a) 100,10,10000,10,10000,10
- b) 100,0,100,0,100,0
- c) 0,0,0,0,0,-10
- d) 100,0,10000,0,16960,0
- e) 100,0,0,0,10000,-10

149. Determinați valorile pe care le afișează programul de mai jos:

```
#include <stdio.h>

int x, y;

int T(int m, int n)
{
 m = n + x;
 n += 1;
 return (n + y + m);
}
void main()
{
 y = 10; x = 12;
```



```

printf("%3d ", T(x, y));
printf("%3d%3d ", x, y);
}

```

- a) 43,22,10
- b) 43,12,10
- c) 47, 10, 12
- d) 44,22,11
- e) 44,12,11

150. Fie programul:

```

#include <stdio.h>

void F (.....)
{
 a += 2;
 b--;
 c = a + b;
}

void main()
{
 int x, y, z;

 x = 2; y = 4;
 F(x, y, z);
 printf("%3d%3d%3d", x, y, z);
}

```

Functia F primeste ca parametri trei numere intregi a,b,si c. Cum trebuie scris antetul complet al functiei, astfel incit programul sa afiseze, in ordine, valorile 4 4 7?

- a) void F(int a, int b, int c)
- b) void F(int a, int &b, int &c)
- c) void F(int &a, int b, int &c)
- d) void F (int &a, int b, int c)
- e) void F (int &a, int &b, int &c)

151. Se consideră următoarea structură de program:

```

#include <iostream.h>

void main()
{
 int v[20], i, n;

 calcul(v, 2.5, 10);
}

void calcul (int v[], float x, int a)
{

}

```

Functia calcul, fiind scrisa dupa main, are nevoie de un prototip care trebuie plasat inaintea lui main. Cum poate arata acest prototip?

- a) void calcul (int v, float x, int a);

- b) void calcul (int v[], float x, int a)
- c) void calcul (int, float, int);
- d) void calcul (int [], float, int);
- e) void calcul (int \*, float, int)

152. Deduceți ce valori va afișa, în ordine, programul următor?

```
#include <iostream.h>

int u[5], v[5], w[5], t[5], i;

void calcul(int *w, int *t)
{
 for (i = 0; i < 3; i++)
 {
 w[i] = 0; t[i] = v[i] - u[i];
 }
}

void main()
{
 for (i = 0; i < 3; i++)
 {
 u[i] = 2 * i - 1; v[i] = 3 * i - 2;
 w[i] = u[i] + v[i]; t[i] = 0;
 }
 calcul(w, t);
 for (i = 0; i < 3; i++) cout << " " << w[i];
 for (i = 0; i < 3; i++) cout << " " << t[i];
}
```

- a) -3 2 7 -1 0 1
- b) 0 0 0 -1 0 1
- c) -3 2 7 0 0 0
- d) -1 0 1 -3 2 7
- e) 0 0 0 -3 2 7

153. Ce valoare va afișa în urma execuției programul următor?

```
#include <iostream.h>

int f(int a = 5, int b = 2)
{
 return --a + 2 * b++;
}

void main()
{
 cout << f();
}
```

- a) 7
- b) 8
- c) 9
- d) 10
- e) 11

154. Se consideră următorul program:

```

int f(int x, int y = 5)
{
 static int m = 3;
 m += x;
 return m>=y ? m : -1;
}
void main()
{
 int a, b, c;
 a = f(4); b = f(1, 7); c = f();
}

```

Care dintre cele trei apeluri realizate in main sunt corecte?

- Numai primele două apeluri sunt corecte iar valorile variabilelor a și b vor fi a=7,b=8.
- Numai primele două apeluri sunt corecte iar valorile variabilelor a și b vor fi a=7,b=-1.
- Numai al doilea apel este corect, și în urma execuției lui se va obține b=8.
- Numai al doilea apel este corect, și în urma execuției lui se va obține b=-1.
- Toate cele trei apeluri sunt corecte.

155. Fie funcția F de mai jos:

```

int* F(int *a, int *b, int c)
{
 d = (*a + *b + c) / 3;
 return &d;
}

```

Presupunem ca sunt declarate urmatoarele variabile:

```
int *x, *y, z, *m, d;
```

Care dintre secvențele de program de mai jos nu afișează corect valoarea returnată de către funcția F?

- scanf("%d %d %d", x, y, &z);  
printf("\n%d",\*F(x, y, z));
- cin >> \*x >> \*y >> z;  
cout << \*F(x, y, z);
- \*m = \*F(x, y, z);  
cout << \*m;
- n = \*F(x, y, z);  
cout << n;
- cout << \*F( 2, 3, 4);

156. Fie funcția:

```

void afis1(int v[])
{
 int i = 0;

 while (i < 4)
 cout << v[i++] << " ";
}

```

Funcțiile afis2 și afis3 de mai jos sunt corecte și echivalente cu funcția dată afis1? (adică afișează același șir de valori ca funcția afis1)

```

void afis2(int v[4])
{
 int i = 0;

 do{
 cout << *(v + i++) << " ";
 }while (i < 4);
}
void afis3(int v[])
{
 int i, *x[4];

 for (i = 0; i < 4; i++)
 {
 x[i] = &v[i];
 cout << *x[i] << " ";
 }
}

```

- Ambele funcții `afis2` și `afis3` conțin erori.
- Ambele funcții `afis2` și `afis3` sunt corecte și echivalente cu `afis1`.
- Ambele funcții `afis2` și `afis3` sunt corecte, dar nici una dintre ele nu este echivalentă cu `afis1`.
- Funcția `afis2` este corectă și echivalentă cu `afis1` iar `afis3` conține erori.
- Funcția `afis3` este corectă și echivalentă cu `afis1` iar `afis2` conține erori.

157. Care va fi valoarea lui `n` în urma execuției programului următor:

```

#include <iostream.h>

int* intreg(float *p)
{
 return (int *)p;
}
void main()
{
 int n;
 float x = -3.56;

 n = x;
 n = *intreg(&x);
}

```

- `n` va avea valoarea -3.
- `n` va avea valoarea -4.
- `n` nu va avea în nici un caz valorile de la a) sau b).
- `n` va avea valoarea -3.56
- Programul conține erori de sintaxă

158. Ce valori se vor afișa în urma execuției programului următor?

```

#include <iostream.h>

int q;

int *A(int *m, int *n)
{

```

```

 *n = *m + 1;
 *m = *n + 1;
 q = *m - 1;
 return &q;
}
void B(int *p)
{
 int y = 4;

 y = *A(&y, p);
}
void main()
{
 int y = 9, x;
 B(&x);
 cout << x << " " << y;
}

```

- a) programul este eronat
- b) 5 și 9
- c) 5 și 0
- d) 0 și 9
- e) 0 și 0

159. Fie funcțiile f1 și f2 avînd prototipurile:

```

void f1(int (*p) [8]);
void f2(int p[4] [8]);

```

Analizati corectitudinea apelurilor celor doua functii in urmatorul program:

```

void main()
{
 int m[4] [8], (*q) [8], *r [8];

 f1(m); f2(m); f1(q); f2(q); f1(r); f2(r);
}

```

- a) Numai primele patru apeluri sunt corecte.
- b) Numai primele două apeluri sunt corecte.
- c) Numai primele două apeluri sunt greșite.
- d) Toate apelurile sunt greșite.
- e) Toate apelurile sunt corecte.

160. Fie programul:

```

#include<iostream.h>

int plus(int x)
{
 return x++;
}
int minus(int x)
{
 return --x;
}

```

```

void test(int p, int q, int (*F1) (int), int (*F2) (int))
{
 if (p && q) cout << F1(p);
 else cout << F2(q);
}
void main()
{
 int a = 2, b = 5;

 test(a, b, plus, minus);
 test(3, 0, plus, minus);
 test(a, b, plus(a), minus(b));
 test(3, 0, plus(3), minus(0));
}

```

Ce se poate spune despre apelurile funcției test realizate din main?

- Primele două apeluri sunt corecte, determinînd afișarea valorilor 2, respectiv, -1, iar ultimile două sunt eronate.
- Primele două apeluri sunt corecte, determinînd afișarea valorilor 3, respectiv, -1, iar ultimele două sunt eronate.
- Ultimele două apeluri sunt corecte, determinînd afișarea valorilor 2, respectiv, -1, iar primele două sunt eronate.
- Ultimele două apeluri sunt corecte, determinînd afișarea valorilor 3, respectiv, -1, iar primele două sunt eronate.
- Toate cele patru apeluri sunt eronate.

161. Ce valoare se va afișa în urma execuției programului următor?

```

#include <iostream.h>

int f(int n,...)
{
 int *a = &n + 1, *b = a + 2;

 return n * (*a + *b);
}
void main()
{
 cout << f(3, 2, 11, 5, 8);
}

```

- 21
- 39
- 18
- 30
- antetul funcției este greșit

162. Precizați ce va afișa programul de mai jos:

```

#include <stdio.h>

char& f(char *p, int n)
{
 return *(p + n);
}
void main()

```

```

{
 char *s = "abc";
 f(s,1) = 'a';
 printf("%s", s);
}

```

- a) Programul nu va afișa nimic pentru că este greșit.
- b) "aac"
- c) "abc"
- d) "aba"
- e) "aabc"

163. Care dintre apelurile de funcție de mai jos va schimba valoarea lui n?

```

#include <stdio.h>

int* f(int& n)
{
 return &n;
}

void main()
{
 int n = 7;
 f(n) = 2; //(I)
 *f(n) = 3; //(II)
}

```

- a) Ambele
- b) Nici una
- c) I
- d) II
- e) Antetul funcției este greșit

164. Analizați programul de mai jos și precizați care dintre afirmațiile date sunt false.

```

int& f1(int& x)
{
 return x = 1;
}

int* f2(int* p)
{
 int a = 2;

 f1(a); *p = 3;
 return p;
}

void main()
{
 int b = 4;

 f1(b);
 f1(b) = 5;
 f2(&b);
 *f2(&b) = 6;
}

```

- a) În urma execuției liniei f1(b), valoarea variabilei b va fi 4.

- b) În urma execuției liniei f1(b)=5, valoarea variabilei b va fi 5.
- c) În urma execuției liniei f2(&b),valoarea variabilei b va fi 3.
- d) În urma execuției liniei \*f2(&b),valoarea variabilei b va fi 6
- e) Valoarea variabilei locale a din funcția f2 nu poate fi modificată prin nici un mecanism în funcția principală main.

165. Se dau următoarele programe:

```
//(I)
char& f(char &a)
{
 if (a > 96) return a = a - 32;
 else return a = a + 32;
}
void main()
{
 char x = 'x';
 f(x) = x;
}
```

```
//(II)
char& f(char &a)
{
 if (a > 96) return a - 32;
 else return a + 32;
}
void main()
{
 char x = 'x';
 f(x) = x;
}
```

Alegeti raspunsurile corecte:

- a) Ambele programe conțin erori.
- b) Primul program este corect și în final x va avea valoarea 'X'.
- c) Al doilea program este corect și în final x va avea valoarea 'x'.
- d) Ambele programe fac același lucru.
- e) Ambele programe sunt corecte, dar nu fac același lucru.

166. Precizați valoarea variabilei n rezultată în urma execuției programului:

```
int f(char a[2])
{
 int i=0;
 while (a[i++]); return i;
}
void main()
{
 int n=f("abcdefgh");
}
```

- a) 0
- b) 2
- c) 9
- d) Corpul funcției conține o buclă infinită.



e) Programul are erori de sintaxă.

167. Precizați șirul de numere afișat de programul următor:

```
#include <stdio.h>

int n = 4;

void ex (int k)
{
 if (k < n)
 {
 ex(k + 1);
 printf("%d ", k);
 }
}

void main()
{
 ex(1);
}
```

- a) 4 3 2 1
- b) 3 2 1
- c) 1 1 1
- d) 1 2 3 4
- e) 1 2 3

168. Deduceți ce valoare va afișa programul următor:

```
#include<stdio.h>

int func(int i)
{
 if (i)
 return 3 * i + 2 - func(i - 1);
 else
 return 0;
}

void main()
{
 printf("%d", func(4));
}
```

- a) 0
- b) 9
- c) 6
- d) 25
- e) 36

169. Deduceți valoarea pe care o afișează programul de mai jos:

```
#include<stdio.h>

int calcul(int n)
{
 if ((n == 0) || (n == 1)) return n;
 else return 3 * calcul(n - 1) - 2 * calcul(n - 2);
}
```

```

}
void main()
{
 printf("%d", calcul(5));
}

```

- a) 0
- b) 31
- c) 5
- d) 15
- e) 1

170. Precizați ce va afișa programul următor:

```

#include <iostream.h>

void binar(int x)
{
 if (x > 1) binar(x / 2);
 cout << (x % 2);
}
void main()
{
 binar(11);
}

```

- a) 1011
- b) 1010
- c) 1101
- d) 1001
- e) 1000

171. Precizați ce valoare afișează următorul program:

```

#include <stdio.h>

int v[8] = {2, 4, 1, 3, 11, 5, 1, 3};

int impar(int n, int m)
{
 if (n == m)
 return v[n] % 2;
 else
 return impar(n, (n + m) / 2) + impar((n + m) / 2 + 1, m);
}
void main()
{
 printf("%d", impar(0, 7));
}

```

- a) 3
- b) 1
- c) 5
- d) 6
- e) 11

172. Precizați de câte ori se execută auto-apelul funcției recursive alternativ și ce valoare afișează următorul program:

```
#include <iostream.h>

int u[6] = {3, -5, 6, -4, 7, -4}, n = 6;

int alternativ(int k, int i)
{
 if (i > n - 1) return 1;
 else
 if ((u[i] >= 0) ^ k)
 return 0;
 else
 return alternativ(!k, ++i);
}

void main()
{
 cout << alternativ(1, 0);
}
```

- a) De patru ori;întoarce 0
- b) De șase ori;întoarce 1
- c) De patru ori;întoarce 1
- d) De cinci ori;întoarce 0
- e) De cinci ori;întoarce 1

173. Precizați ce șir va afișa programul următor, dacă se introduce secvența de cifre separate prin spații: 0 2 4 6 8 urmată de Enter?

```
#include <iostream.h>

void cifre(int n)
{
 cin >> n;
 if (n != 6) cifre(n);
 if (n != 2) cout << n;
}

void main()
{
 cifre(9);
}
```

- a) 29
- b) 4
- c) 640
- d) 40
- e) 60

174. Ce se salvează pe stivă în urma apelului unui subprogram recursiv? Selectați enunțul complet.

- a) Adresa de revenire, variabilele locale și parametrii transmiși prin referința (adresă).
- b) Adresa de revenire, variabilele locale, parametrii transmiși prin valoare și prin referință.
- c) Variabilele locale și parametrii transmiși prin valoare.
- d) Adresa de revenire și variabilele locale.

e) Adresa de revenire, variabilele locale și parametrii transmiși prin valoare.

175. Precizați ce valori se vor afișa, în ordine, în urma execuției funcției recursive afișare, dacă primul apel(cele din main) este afișare(2).

```
void afisare (int k)
{
 if (k < 6)
 {
 printf("%d ", 2 * k - 1);
 afisare(k + 1);
 }
}
```

- a) 3,5,7,9
- b) 3,5,7
- c) 3,5
- d) 9,7,5,3
- e) 7,5,3

176. Se consideră următoarea funcție:

```
int ce(int i)
{
 if (i <= 1) return i;
 else
 ce(i - 2) + i;
}
```

Ce valoare va returna ce(5)?

- a) 1
- b) 5
- c) 8
- d) 9
- e) 15

177. Care este valoarea returnată de funcția următoare, la apelul f(4)?

```
int f(int x)
{
 if (x <= 0) return 3;
 else
 f(x - 1) * 2;
}
```

- a) 16
- b) 24
- c) 48
- d) 3
- e) 4

178. Se consideră funcția definită recursiv:

```

int ce(int i)
{
 if (i == 0) return 0;
 else ce(i - 1) + 2 * i;
}

```

Ce valoare va returna ce(5)?

- a) 0
- b) 10
- c) 15
- d) 20
- e) 30

179. Se consideră funcția f de mai jos definită recursiv. Ce va returna f(7)?

```

int f(int i)
{
 if (i == 0) return 0;
 else
 if (i % 2 == 0)
 f(i - 1) + i;
 else
 f(i - 1) - i;
}

```

- a) -28
- b) 28
- c) -4
- d) 4
- e) 0

180. Fie următoarele funcții recursive, în care parametrul k este un număr natural:

```

int F1(int k)
{
 if (k > 0)
 return k * F1(k - 1);
 else
 return 1;
}

```

```

int F2(int k)
{
 if (k == 1)
 return 1;
 else
 return k * F2(k - 1);
}

```

```

int F3(int k)
{
 if (k == n)
 return 1;
 else
 return k * F3(k + 1);
}

```

În funcția main citim de la tastatură un număr natural  $n$ , apoi facem un prim apel al uneia dintre funcțiile de mai sus, memorînd valoarea returnată într-o variabilă întregă  $x$ . Cum trebuie să arate acest apel, astfel încît valoarea lui  $x$  să fie factorialul lui  $n$ ? (Reamintim: produsul  $1 \cdot 2 \cdot \dots \cdot n$ , notat  $n!$ , se numește factorialul numărului natural  $n$ ).

- a)  $x=F1(1)$
- b)  $x=F1(n)$
- c)  $x=F2(1)$
- d)  $x=F2(n)$
- e)  $x=F3(1)$

181. Deduceți șirul de numere afișate în urma execuției programului:

```
#include <stdio.h>

int x;

void A(int i)
{
 if (i < 4)
 {
 printf("%2d", x - 2);
 A(i + 1);
 printf("%2d", x + 3);
 }
}

void main()
{
 x = 5; A(1);
}
```

- a) 3 8 3 8 3 8
- b) 3 3 3 3 8 8 8 8
- c) 3 3 3 8 8 8
- d) 8 8 8 3 3 3
- e) 8 8 8 8 3 3 3 3

182. Pentru funcția recursivă

```
void exemplu (int m, int n, int &x, int &t)
{
 float y; char c;

 if (m < n) exemplu(m, n, x, t);

}
```

precizați identificatorii de variabile care ocupă într-o formă sau alta spațiu pe stivă, ca urmare a execuției fiecărui apel (autoapel);

- a)  $m, n, x, t$
- b)  $y, c$
- c)  $y, c, x, t$
- d)  $y, c, m, n$
- e)  $y, c, m, n, x, t$

183. Precizați expresia care lipsește pe ramura else în funcția E, astfel încât în urma execuției instrucțiunii `printf("%d", F(6, 27))` să se afișeze cîtu întreg al împărțirii lui 27 la 6 (adică 4).

```
int E(int a, int b)
{
 if (a > b) return 0;
 else return
}
```

- a)  $1 + E(b, b - a)$ ;
- b)  $1 + E(a, b - a)$ ;
- c)  $1 + E(a - b, b)$ ;
- d)  $1 + E(a - b, a)$ ;
- e)  $1 + E(a - b, b)$

184. Analizați programul de mai jos și dați răspunsul corect:

```
#include <iostream.h>

int f(int a)
{
 if (a > 0)
 {
 a--; f(a - 1);
 }
 else return a;
}

void main()
{
 int a;
 cout << "\n" << f(3);
 cout << "\n" << f(4);
}
```

- a) Afîșează valoarea 0 de două ori.
- b) Afîșează valoarea -1 de două ori.
- c) Afîșează valorile -1 și 0.
- d) Execuția se soldează cu depășirea stivei, deoarece conține o ramură infinită de apeluri ale funcției recursive f.
- e) Programul este eronat, deoarece variabila a este declarată de două ori, o dată global, iar a doua oară ca parametru formal al funcției f.

185. Fie funcția recursivă de mai jos:

```
void ex(int a, int &b)
{
 int c = a + 2;
 if (a>0) ex(a / 2, c);
 printf("%d %d %d\n", a, b, c);
 b += 1;
}
```

Ce valori se vor afișa în urma celei de-a treia execuții a instrucțiunii `printf`, dacă în `main` avem instrucțiunile: `x = 0; ex(12, x);`

- a) 1 5 4

- b) 3 8 5
- c) 6 14 9
- d) 6 14 10
- e) 3 8 6

186. Pentru programul următor, care dintre afirmațiile date sunt adevărate?

```
#include <iostream.h>

int n = 5; u[] = {8, 3, 1, 7, 5};

int calcul (int i, int &h)
{
 if (i >= n) return h;
 else
 {
 if (u[i] > h) h = u[i];
 return calcul(i + 1, h);
 }
}

void main ()
{
 int h = 0;

 cout << calcul(0, h);
}
```

- a) Programul afișează valoarea 1.
- b) Programul afișează valoarea 8.
- c) Funcția conține erori de sintaxă
- d) Execuția programului va genera un lanț infinit de autoapeluri ale funcției.
- e) În cazul în care în main am fi avut numai `cout << calcul(0,0)`, în loc de `int h=0` și `cout << calcul(0,h)` atunci funcția ar fi avut același efect.

187. Care dintre secvențele de mai jos definesc corect un pointer p către un nod al unei liste liniare simplu înlănțuite de numere întregi ?

- a) 

```
typedef struct un_nod{
 int info;
 un_nod urm;
}TNOD;
TNOD* p;
```
- b) 

```
typedef struct un_nod{
 int info;
 un_nod& urm;
}TNOD;
TNOD* p;
```
- c) 

```
typedef struct un_nod{
 int info;
 un_nod* urm;
}TNOD;
TNOD * p;
```
- d) 

```
typedef struct un_nod{
 int info;
 un_nod* urm;
}TNOD;
TNOD p;
```



```
e) typedef un_nod{
 int info;
 un_nod* urm;
}TNOD;
TNOD* p;
```

188. Fie p un pointer către primul nod al listei liniare simplu înlănțuite de numere întregi definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Care dintre secvențele de program următoare mută corect pointerul p la nodul al șaselea

```
//Secventa S1
for (i = 1; i <= 5; i++)
 p = p->urm;
```

```
//Secventa S2
i = 1;
while (i <= 4)
{
 p = p->urm;
 i++;
}
```

```
//Secventa S3
i = 1;
do{
 p = p->urm;
 i++;
}while (i <= 5);
```

- a) numai S1
- b) numai S2
- c) S1 și S3
- d) S1 și S2
- e) toate

189. Se consideră o listă liniară simplu înlănțuită de numere întregi, definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Dacă p este un pointer către primul nod, iar numerele din noduri sunt (5,4,7,6) în această ordine, de la stînga la dreapta, ce valoare va returna funcția F ?

```
int F(TNOD *p)
{
 int S = 0;

 while (p->urm)
 {
 if (!(p->info % 2)) S += p->info;
 p = p->urm;
 }
}
```

```

 }
 return S;
}

```

- a) 4
- b) 10
- c) 16
- d) 12
- e) 22

190. Fie secvența de program următoare, în care p este un pointer către primul nod al unei liste liniare simplu înlănțuite de numere întregi definită astfel:

```

typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;

```

Pentru care dintre listele date mai jos secvența va afișa valoarea 1? În fiecare listă din variantele de răspuns, cheile din noduri sunt enumerate prin parcurgerea de la stînga la dreapta.

```

int x = 1;

while (p->urm)
{
 if (p->urm->info >= p->info)
 x = 0;
 p = p->urm;
}
cout << x;

```

- a) (5,4,3,2,1)
- b) (1,2,3,4,5)
- c) (1,1,1,1,1)
- d) (5,4,3,3,1)
- e) (1,3,3,4,5)

191. În secvența următoare, q este un pointer către primul nod al unei liste liniare simplu înlănțuite de numere întregi definită astfel:

```

typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;

```

Precizați ce va afișa secvența, dacă numerele din listă sunt, în ordine, de la stînga la dreapta, (-5,8,6,8,8,-8)

```

int a = 0, z = 8;

while (q)
{
 if (q->info == z) a++;
 q = q->urm;
}
cout << a;

```

- a) 0

- b) 1
- c) 2
- d) 3
- e) 4

192. În secvența de program,

```
TNOD *p = prim->urm;
int max = MAXINT;

while (p != ultim)
{
 if (max > p->info)
 p->info = max;
 p = p->urm;
}
```

prim și ultim sunt pointeri către primul, respectiv, ultimul nod al unei liste liniare dublu înlănțuite de numere întregi definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * prec;
 un_nod * urm;
}TNOD;
```

Precizați câte erori de sintaxă și de concepție trebuie corectate (fără a adăuga sau elimina instrucțiuni) astfel, încât, după execuția secvenței date, valoarea variabilei max să reprezinte cel mai mare dintre elementele (numerele) memorate în listă. Observație: presupunem că în program este inclus header-ul values.h care conține definiția constantei MAXINT.

- a) una
- b) două
- c) nici una
- d) trei
- e) patru

193. Se consideră o listă liniară dublu înlănțuită definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * prec;
 un_nod * urm;
}TNOD;
```

în care Ls și Ld sunt pointeri către primul, respectiv, ultimul nod al listei. Dacă inițial numerele din nodurile listei sunt, în ordine de la stînga la dreapta, 0,1,2,3,4,5,6 deduceți care vor fi aceste numere după execuția secvenței următoare (în variantele de răspuns, ele sunt date tot prin parcurgerea listei de la stînga la dreapta).

```
void algoritm()
{
 int x; TNOD *p, *q;

 p = Ls->urm; q = Ld->prec;
 do{
 x = q->info; p->info = x + 2;
 x = p->info; q->info = x - 2;
 p = p->urm; q = q->prec;
 }while ((p != q) && (q != Ls) && (p != Ld));
}
```

- a) 0,-1,0,3,6,7,6
- b) 0,7,6,3,4,5,6
- c) 0,7,6,3,2,3,6
- d) 0,3,4,3,2,3,6
- e) Secvența va intra într-un ciclu infinit

194. Fie p un pointer către primul nod al listei liniare simplu înlănțuite de numere întregi definite astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Ce nod va avea adresa pointerului p în urma execuției secvenței de program de mai jos?

```
i = 0;
do{
 i++;
 p = p->urm;
} while (i <= 6);
```

- a) al cincilea
- b) al șaselea
- c) al șaptelea
- d) al optulea
- e) al nouălea

195. Fie secvența de program următoare,

```
nr = 0;
while (p->urm)
{
 if (p->urm->info - p->info == 1) nr++;
 p = p->urm;
}
cout << nr;
```

în care p un pointer către primul nod al unei liste liniare simplu înlănțuite de numere întregi, definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Precizați pentru care dintre listele Lista\_1 ... Lista\_5 de mai jos, se va afișa valoarea 3 în urma execuției secvenței (între paranteze enumerat, în ordine, numerele din nodurile listei).

- a) Lista\_1: (2,4,6,7,8)
- b) Lista\_2: (2,3,4,5,6)
- c) Lista\_3: (2,3,6,7,8)
- d) Lista\_4: (2,3,5,6,8)
- e) Lista\_5: (2,4,5,7,8)

196. Fie p un pointer către primul nod al listei liniare simplu înlănțuite de numere întregi definite astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Ce instrucțiune trebuie executată pentru a se tipări numărul memorat în al treilea nod?

- a) `cout << p->urm->info->info;`
- b) `cout << p->urm->info->urm;`
- c) `cout << p->urm->urm->info;`
- d) `cout << p->urm->urm;`
- e) `cout << p->urm->info;`

197. Declarați o funcție test, care să primească drept parametru un pointer b către primul nod al unei liste liniare simplu înlănțuite definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

și care nu returnează nimic.

- a) `void test(b TNOD *);`
- b) `void test(b TNOD);`
- c) `void test(TNOD * b);`
- d) `void test(TNOD b);`
- e) `void test(b * TNOD);`

198. Funcțiile următoare au ca parametru un pointer q către primul nod al unei liste liniare simplu înlănțuite de numere întregi, definită astfel:

```
typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;
```

Funcțiile test definite în continuare, trebuie să întoarcă 1 dacă lista conține numai chei (valori) nenule sau 0 dacă lista are cel puțin o cheie (valoare) nulă. Care dintre aceste funcții realizează acest lucru?

- a) `int test1(TNOD *q)`

```
{
 int b = 1;
 while (q)
 {
 if (q->info == 0)
 b = 0;
 q = q->urm;
 }
 return b;
}
```
- b) `int test2(TNOD *q)`

```
{
 int b = 0;
 while (q)
 {
 if (q->info == 0)
```

```

 b = 1;
 q = q->urm;
 }
 return b;
}
c) int test3(TNOD *q)
{
 int b = 1;
 while (q)
 {
 q = q->urm;
 if (q->info == 0)
 b = 0;
 }
 return b;
}
d) int test4(TNOD *q)
{
 int b;
 while (q)
 {
 if (q->info == 0) b = 1;
 else b = 0;
 q = q->urm;
 }
 return b;
}
e) int test5(TNOD *q)
{
 int b;
 while (q)
 {
 if (q->info == 0) b = 0;
 else b = 1;
 q = q->urm;
 }
 return b;
}

```

199. Definim o listă liniară simplu înlănțuită de numere întregi astfel:

```

typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;

```

Funcțiile de mai jos au ca parametru un pointer p către primul nod al listei.

```

TNOD* F1(TNOD *p)
{
 TNOD *r;

 r = (TNOD *)malloc(sizeof(TNOD));
 cin >> r->info; p->urm = r;
 return p;
}

```

```

TNOD* F2(TNOD *p)

```

```

{
 TNOD *r;

 r = (TNOD *)malloc(sizeof(TNOD));
 cin >> r->info; r->urm = p;
 return r;
}

```

```

TNOD* F3(TNOD *p)
{
 TNOD *r;

 *r = (TNOD *) malloc(sizeof(TNOD));
 cin >> r->info; r->urm = p;
 return *r;
}

```

Precizați care dintre aceste funcții realizează corect inserarea unui nod nou, adresat de pointerul r, la începutul listei rezultate după inserare

- a) numai F1
- b) numai F2
- c) numai F3
- d) F2 și F3
- e) nici una

200. Fie o listă liniară simplu înlănțuită de numere întregi, definită astfel:

```

typedef struct un_nod{
 int info;
 un_nod * urm;
}TNOD;

```

ale cărei elemente (numerele memorate în câmpurile info ale nodurilor) sunt (2,6,5,9,7). Lista este referită prin intermediul unui pointer a către primul nod. Deduceți care vor fi elementele listei referite de pointerul b, după execuția secvenței de mai jos:

```

TNOD *b, *r;
int x = 11, m = 3, i;

b = a;
for (i = 1; i < m; i++) a = a->urm;
r = (TNOD *)malloc(sizeof(TNOD));
r->info = x; r->urm = a->urm; a->urm = r;

```

- a) (2)
- b) (2,6,5,9,7)
- c) lista este vidă
- d) (2,6,11,5,9,7)
- e) (2,6,5,11,9,7)

201. Fie o listă liniară dublu înlănțuită de numere întregi, definită astfel:

```

typedef struct un_nod{
 int info;
 un_nod * prec;
 un_nod * urm;
}TNOD;

```

ale cărei elemente (numerele memorate în câmpurile info ale nodurilor) sunt de la stînga la dreapta, (0,-2,3,-4,5,-6,7,0). Ce valoare va returna funcția F ?

```
int F(TNOD *prim, TNOD *ultim)
{
 int S = 0, i = 1;
 TNOD *p = prim->urm;

 while (p != ultim)
 {
 i++;
 if ((p->info > 0) && (i % 2 != 0)) S += p->info;
 p = p->urm;
 }
 return S;
}
```

- a) 3
- b) 0
- c) 8
- d) 15
- e) 10

202. Pentru programul următor, precizați care dintre afirmațiile date sunt adevărate.

```
#include <iostream.h>
#include <math.h>

void main()
{
 cout << sqrt(9);
}
```

- a) Este greșit, deoarece instrucțiunea cout afișează conținutul unei variabile.
- b) Este greșit, deoarece funcția sqrt nu este recunoscută în program.
- c) Afișează valoarea 3.
- d) Afișează valoarea 81.
- e) Fiind o singură instrucțiune de afișare în program, nu mai era nevoie de "{" si "}"

203. Ce valoare afișează următoarea secvență de program ?

```
int n = 10;

while (n > 0)
 do{
 n--;
 }while (n != -2);
cout << n;
```

- a) 10
- b) 2
- c) -2
- d) -3
- e) programul conține un ciclu infinit



204. Știm că, în conformitate cu standardul ASCII, literele mari au codurile succesive începînd cu 65 ('A'← 65,'B'← 66, etc.), iar literele mici au codurile succesive începînd cu 97 ('a'← 97, 'b'← 98, etc.). Care dintre următoarele funcții realizează transformarea reciprocă a caracterelor, din litere mari în litere mici și invers?

- a) 

```
char reciproc(char a)
{
 if (a > 96) return a - 32;
 else return a + 32;
}
```
- b) 

```
char reciproc(char a)
{
 if (a > 96) return 'a - 32';
 else return 'a + 32';
}
```
- c) 

```
char reciproc(char a)
{
 if ('a' > 96) return a - 32;
 else return a + 32;
}
```
- d) 

```
char reciproc(char a)
{
 if ('a' > 96) return 'a - 32';
 else return 'a + 32';
}
```
- e) 

```
char reciproc(char a)
{
 if (a > 96) return 'a' - 32;
 else return 'a' + 32;
}
```

205. Precizați ce valoare se va afișa în urma execuției programului următor.

```
#include <iostream.h>

void main()
{
 int a = 17, b = (a / 2) % a;
 int *x = &b;

 cout << "\n" << *x;
}
```

- a) 8
- b) 17
- c) 0
- d) o adresă de memorie
- e) secvența este eronată

206. Se dă programul următor:

```
void main()
{
 ; ;
}
```

- a) Programul este corect.
- b) Programul este greșit pentru că nu are nici o instrucțiune.

- c) Programul este greșit pentru că după "{" nu poate urma semnul ";"
- d) Programul este greșit pentru că nu există nici o declarație de variabilă
- e) Programul este greșit pentru că nu conține nici un header inclus.

207. Fie variabilele x, y, z, toate de tipul int, cu valorile x = 2, y = 3, z = 1. Expresia x - (y < 3) < z + x && y are valoarea:

- a) -1
- b) 0
- c) 1
- d) o valoare imprevizibilă
- e) expresia este eronată

208. Ce valoare afișează în urma execuției secvenței de mai jos?

```
i = 4; S = 0;
while (i <= 10)
 i++; S += i;
cout << "\n" << S;
```

- a) 0
- b) 10
- c) 11
- d) 56
- e) secvența conține erori

209. Care dintre cele trei instrucțiuni printf ale funcției main de mai jos tipăresc succesiunea de caractere "bd"?

```
void main()
{
 char s[6][2] = {"ab", "ac", "ad", "bc", "bd", "cd"};
 printf("%c%c", s[3][0], s[2][1]);
 printf("%s", s[3][0] + s[2][1]);
 printf("%s", s[5]);
}
```

- a) Toate
- b) Numai prima
- c) Numai primele două
- d) Numai prima și a treia
- e) Nici una

210. Ce afișează programul următor?

```
#include <iostream.h>

int f(int &a)
{
 int i;

 a++;
 for(i = 14; i < 20; i++)
 if (i % a == 0)
 return i;
 return 0;
}
```

```

}

void main()
{
 int x = 2;

 cout << f(x) << "\n" << x;
}

```

- a) 0 și 2
- b) 15 și 3
- c) 15 și 2
- d) 18 și 2
- e) 18 și 3

211. Știind că fișierul `date.dat` conține două linii, precizați răspunsul corect analizând programele următoare.

```

//Programul P1
#include <stdio.h>

void main()
{
 FILE *f;
 int n = 0;

 f = fopen("date.dat", "r");
 while (!feof(f))
 {
 fscanf(f, "\n");
 n++;
 }
 fclose(f);
 printf("%d\n", n);
}

```

```

//Programul P2
#include <stdio.h>

void main()
{
 FILE *f;
 int n = 0;

 f = fopen("date.dat", "r");
 do{
 n++;
 fscanf(f, "\n");
 }while(!feof(f));
 fclose(f);
 printf("%d\n", n);
}

```

- a) Programul P1 este greșit pentru că citește din fișier după ce s-a trecut de ultima linie.
- b) Ambele programe afișează valoarea 2.
- c) Programul P1 afișează valoarea 1, iar programul P2 afișează valoarea 2.
- d) Ambele programe sunt greșite, deoarece conduc la un ciclu infinit, în urma faptului că instrucțiunea de citire nu conține nici o variabilă.

- e) Ambele programe sunt greșite, deoarece nu s-au folosit operatori relaționali în condițiile ciclurilor de citire din fișierul text.

212. Pentru programul următor, care dintre afirmațiile date sunt adevărate ?

```
#include <iostream.h>

int n = 5; u[] = {8, 3, 1, 7, 5};

int calcul(int i, int &h)
{
 if (i >= n) return h;
 else
 {
 if (u[i] > h) h = u[i];
 return calcul(i + 1, h);
 }
}

void main()
{
 int h = 0;

 cout << calcul(0, h);
}
```

- a) Programul afișează valoarea 1  
b) Programul afișează valoarea 8  
c) Funcția conține erori de sintaxă  
d) Execuția programului va genera un lanț infinit de autoapeluri ale funcției  
e) Programul afișează valoarea 5
213. Dacă m, n și p sunt variabile întregi, iar x, y și z sunt variabile reale, identificați care dintre instrucțiunile de mai jos reprezintă atribuiri corecte.
- a)  $x+=2$ ;  
b)  $2*x+y$   
c)  $m*n=p$ ;  
d)  $n=m+z+p*x$ ;  
e)  $p:=y$ ;
214. Dacă x, y, z sunt trei variabile de tipul int, cu valorile inițiale  $x=1$ ,  $y=0$ ,  $z=1$ , precizați care dintre afirmațiile de mai jos sunt adevărate.

- a) Expresia  $x<y$  are valoarea 1  
b) Instrucțiunea `scanf("%d %d", x, y)` este corectă  
c) În urma execuției instrucțiunii `z = (abs(-2) - 11 % 3 == 0)`, variabila z va primi valoarea 1  
d) Instrucțiunea `printf("%d", !x - y)` va afișa valoarea 1  
e) Instrucțiunea `cout << (x+y)/2` va afișa valoarea 0.5

215. Ce valoare afișează secvența de program:

```
#include <iostream.h>

void main()
{
```

```

int n = 1008, m = 416;

while (n != m)
 if (n < m)
 m = m % n;
 else
 n = n % m;
cout << n;
}

```

- a) 416
- b) 0
- c) 1008
- d) 36
- e) programul este greșit

216. Ce valoare va afișa programul următor:

```

#include <iostream.h>

int c(int x)
{
 if (x > 0) c(x - 1);
 return x;
}

void main()
{
 int x = 10;

 cout << c(x);
}

```

- a) 0
- b) 1
- c) 9
- d) 10
- e) programul este greșit

217. Pentru programul următor, care dintre afirmațiile date sunt adevărate ?

```

#include <iostream.h>

int X(int a;int b)
{
 int m = a + b;
 return m;
}

void main()
{
 cout << X(0.5, 1.5); //(A)
 cout << m; //(B)
}

```

- a) Linia (A) afișează valoarea 1.

- b) Linia (A) afișează valoarea 2.
- c) Apelul funcției din linia (A) este eronat.
- d) Afișarea din linia (B) este eronată
- e) În program există și alte erori pe lângă cele de mai sus.

218. Se consideră două mulțimi, cu  $m$ , respectiv,  $n$  elemente numere întregi memorate în vectorii  $u$  și  $v$ . Presupunem că la citirea vectorilor s-a asigurat condiția ca elementele fiecărei mulțimi să fie distincte. Ce mulțime va memora vectorul  $w$  după execuția secvenței de mai jos?

```
k = 0;
for (i = 0; i < m; i++)
{
 g = 0;
 for (j = 0; j < n; j++)
 if (u[i] == v[j]) g = 1;
 if (!g) w[k++] = u[i];
}
```

- a)  $u \cup v$
- b)  $u \cap v$
- c)  $v \cap u$
- d)  $u - v$
- e)  $v - u$

219. Ce afișează programul următor?

```
#include <iostream.h>

void main()
{
 int a = 2, b = 3, x, m, n;
 x = ((m = a - b) > 0, b++, n = (!a + b == 4));
 cout << "\n" << m << " " << n << " " << x;
}
```

- a) 0 0 0
- b) 0 0 1
- c) 0 1 1
- d) -1 1 1
- e) -1 1 -1

220. Fie variabilele  $a$  și  $b$ , de tipul float, cu valorile  $a=2.75$  și  $b=3.25$ . Precizați care dintre instrucțiunile următoare afișează valoarea 3.25 în formatul de mai jos:

□□3.2500

(Prin "□" am simbolizat caracterul spațiu)

- a) `printf("\n%-8.4f", x = a < b ? b : a);`
- b) `printf("\n%8.4f", x = a > b ? a : b);`
- c) `printf("\n%4.4f", x = a > b ? a : b);`
- d) `printf("\n%-4.4f", x = a < b ? a : b);`
- e) `printf("\n%8.4f", x = a > b ? b : a);`

221. Care dintre secvențele de program de mai jos afișează 420 ?

```

//Secventa S1
for (i = 0; i <= 4; i++)
 cout << 4 - i++;

//Secventa S2
for (i = 5; i-- >= 1;)
 cout << i,i--;

//Secventa S3
i = 5;
while (i--)
 cout << --i;

```

- a) numai S1
- b) nici una
- c) S1 și S2
- d) S1 și S3
- e) toate

222. Ce afișează programul următor?

```

#include <iostream.h>

void main()
{
 char *s[3] = {"123", "456", "789"};
 char **v;

 v = &s[1];
 cout << (*v)[1];
}

```

- a) 1
- b) 5
- c) 56
- d) 456
- e) o adresă de memorie

223. Știind că a este o matrice cu 4 linii \* 4 coloane, care dintre următoarele secvențe de program sunt echivalente?

I)

```

int i, j, k, S = 0;
for (i = 0; i < 4; i++)
 for (j = 0; j < 4; j++)
 S += a[i][j];
cout << S;

```

II)

```

int i, j, k, S = 0;
for (i = 0; i < 4; i++)
 for(j = 3; j >= 0; j--)
 S += a[j][i]
cout << S;

```

III)

```

int i, j, k, S = 0;

```

```

for (i = 0; i < 4; i++)
{
 for (j = 0; j <= i; j++)
 S += a[i][j];
 for (k = 0; k <= i - 1; k++)
 S += a[k][i];
}
cout << S

```

- a) I și II
- b) I și III
- c) II și III
- d) toate
- e) nici una

224. Ce valoare returnează funcția f la apelul f(4)?

```

int f(int x)
{
 if (x <= 0) return 3;
 else return f(x - 1) * 2;
}

```

- a) 16
- b) 24
- c) 48
- d) 3
- e) 4

225. Precizați răspunsul corect, analizând programul următor.

```

#include <iostream.h>

int* f(int &n)
{
 int x = n++;
 return &n;
}

void main()
{
 int n = 3;
 cout << *f(n);
}

```

- a) Instrucțiunea cout este eronată.
- b) Declarația funcției este greșită.
- c) Programul afișează valoarea 3.
- d) Programul afișează valoarea 4.
- e) Programul afișează adresa variabilei n.

226. Fie programul următor:



```

#include <stdio.h>

void main()
{
 FILE *f, *g;
 int a, x, S;

 f = fopen("in.txt", "r");
 g = fopen("out.txt", "w");
 scanf("%d", &a);
 while (!feof(f))
 {
 S = 0;
 while (S < a && !feof(f))
 { fscanf(f, "%d", &x); S += x; }
 fprintf(g, "%d ", S);
 }
 fclose(f); fclose(g);
}

```

Dacă de la tastatură se introduce valoarea 10, iar conținutul fișierului 'in.txt' este cel de mai jos, câte numere va scrie programul în fișierul 'out.txt'? 4 6 3 2 6 15 1

- a) nici unul
- b) unul
- c) două
- d) trei
- e) patru

227. Se consideră o listă liniară simplu înlănțuită de numere întregi, definită astfel:

```

typedef struct nod {
 int info; //informatia utila din nod
 nod *urm; //pointer c"atre nodul urm"ator
} TNOD;
TNOD *p;

```

Dacă p este un pointer către primul nod al listei, iar numerele memorate în câmpurile info ale nodurilor sunt, în ordine, de la stînga la dreapta (1,2,3,4,5,6,7), deduceți ce va returna funcția F în urma apelului F(1,2,3,4,5,6,7).

```

int F(int n, ...)
{
 int *a = &n + 1, *b = a + 2;

 for (i = 1; i <= *b; i++)
 p = p->urm;
 return p->info;
}

```

- a) 2
- b) 3
- c) 4
- d) 5
- e) 6

228. Care dintre variantele de mai jos reprezintă o declarație corectă a unui șir de caractere?

- a) `char s[20];`
- b) `char *s[20];`
- c) `char *s;`
- d) `char s;`
- e) `char s[];`

229. Cîte elemente ale matricii a vor avea valoarea 0 după execuția secvenței de program de mai jos?

```
int i, a[3][3] = {{0, 1, 2}, {0, 1, 2}, {0, 1, 2}};
for (i = 0; i < 3; i++)
 !(i%2) ? a[i][2 - i] = *a[i] : 0;
```

- a) două
- b) trei
- c) patru
- d) cinci
- e) șase

230. Precizați valoarea variabilei n în urma execuției programului următor.

```
int F(char a[2])
{
 int i = 0;

 while (a[i++]);
 return i;
}

void main()
{
 int n = F("abcdefgh");
}
```

- a) 9
- b) 8
- c) 2
- d) Corpul ciclului conține o buclă infinită.
- e) Programul este greșit, deoarece șirul "abcdefgh", dat ca parametru actual, nu corespunde ca lungime cu parametrul formal a al funcției F.

| Test | Raspuns |
|------|---------|
| 1    | c       |
| 2    | c       |
| 3    | d       |
| 4    | a       |
| 5    | d       |
| 6    | c       |
| 7    | e       |
| 8    | c       |
| 9    | b       |
| 10   | b       |
| 11   | b       |
| 12   | c       |
| 13   | d       |
| 14   | e       |
| 15   | c       |
| 16   | c       |
| 17   | b       |
| 18   | a       |
| 19   | d       |
| 20   | e       |
| 21   | e       |
| 22   | a       |
| 23   | a       |
| 24   | c       |
| 25   | c       |
| 26   | c       |
| 27   | e       |
| 28   | b       |
| 29   | e       |
| 30   | b       |
| 31   | d       |
| 32   | b       |
| 33   | c       |
| 34   | e       |
| 35   | a       |
| 36   | b       |
| 37   | c       |
| 38   | a       |
| 39   | e       |
| 40   | e       |

| Test | Raspuns |
|------|---------|
| 41   | b       |
| 42   | c       |
| 43   | c       |
| 44   | d       |
| 45   | d       |
| 46   | b       |
| 47   | b       |
| 48   | d       |
| 49   | c       |
| 50   | d       |
| 51   | e       |
| 52   | b       |
| 53   | b       |
| 54   | b       |
| 55   | c       |
| 56   | e       |
| 57   | b       |
| 58   | d       |
| 59   | e       |
| 60   | d       |
| 61   | b       |
| 62   | a       |
| 63   | c       |
| 64   | d       |
| 65   | c       |
| 66   | c       |
| 67   | d       |
| 68   | b       |
| 69   | c       |
| 70   | a       |
| 71   | c       |
| 72   | d       |
| 73   | e       |
| 74   | c       |
| 75   | d       |
| 76   | c       |
| 77   | d       |
| 78   | e       |
| 79   | d       |
| 80   | b       |

| Test | Raspuns |
|------|---------|
| 81   | b       |
| 82   | c       |
| 83   | d       |
| 84   | b       |
| 85   | c       |
| 86   | e       |
| 87   | a       |
| 88   | c       |
| 89   | c       |
| 90   | c       |
| 91   | c       |
| 92   | d       |
| 93   | d       |
| 94   | c       |
| 95   | c       |
| 96   | b       |
| 97   | b       |
| 98   | a       |
| 99   | b       |
| 100  | d       |
| 101  | b       |
| 102  | d       |
| 103  | b       |
| 104  | a       |
| 105  | e       |
| 106  | d       |
| 107  | a       |
| 108  | a       |
| 109  | b       |
| 110  | e       |
| 111  | b       |
| 112  | d       |
| 113  | b       |
| 114  | c       |
| 115  | e       |
| 116  | e       |
| 117  | a       |
| 118  | b       |
| 119  | d       |
| 120  | b       |

| Test | Raspuns |
|------|---------|
| 121  | b       |
| 122  | a       |
| 123  | d       |
| 124  | d       |
| 125  | c       |
| 126  | c       |
| 127  | a       |
| 128  | b       |
| 129  | e       |
| 130  | e       |
| 131  | b       |
| 132  | b       |
| 133  | c       |
| 134  | d       |
| 135  | c       |
| 136  | e       |
| 137  | e       |
| 138  | c       |
| 139  | a       |
| 140  | a       |
| 141  | b       |
| 142  | e       |
| 143  | c       |
| 144  | e       |
| 145  | c       |
| 146  | e       |
| 147  | c       |
| 148  | e       |
| 149  | b       |
| 150  | c       |
| 151  | d       |
| 152  | b       |
| 153  | b       |
| 154  | a       |
| 155  | e       |
| 156  | b       |
| 157  | c       |
| 158  | b       |
| 159  | a       |
| 160  | a       |

| Test | Raspuns |
|------|---------|
| 161  | a       |
| 162  | b       |
| 163  | d       |
| 164  | a       |
| 165  | b       |
| 166  | c       |
| 167  | b       |
| 168  | c       |
| 169  | b       |
| 170  | a       |
| 171  | d       |
| 172  | b       |
| 173  | c       |
| 174  | e       |
| 175  | a       |
| 176  | d       |
| 177  | c       |
| 178  | e       |
| 179  | c       |
| 180  | b       |
| 181  | c       |
| 182  | e       |
| 183  | b       |
| 184  | c       |
| 185  | e       |
| 186  | b       |
| 187  | c       |
| 188  | c       |
| 189  | a       |
| 190  | a       |
| 191  | d       |
| 192  | d       |
| 193  | b       |
| 194  | d       |
| 195  | c       |
| 196  | c       |
| 197  | c       |
| 198  | a       |
| 199  | b       |
| 200  | e       |

| Test | Raspuns |
|------|---------|
| 201  | d       |
| 202  | c       |
| 203  | c       |
| 204  | a       |
| 205  | a       |
| 206  | a       |
| 207  | c       |
| 208  | c       |
| 209  | b       |
| 210  | c       |
| 211  | d       |
| 212  | b       |
| 213  | a       |
| 214  | c       |
| 215  | e       |
| 216  | d       |
| 217  | e       |
| 218  | d       |
| 219  | d       |
| 220  | b       |
| 221  | c       |
| 222  | b       |
| 223  | d       |
| 224  | c       |
| 225  | d       |
| 226  | e       |
| 227  | d       |
| 228  | a       |
| 229  | c       |
| 230  | a       |

### 3 Algoritmi, logica și programare

1. Pentru  $n$  și  $k$  întregi pozitivi, coeficientul lui  $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$  din dezvoltarea  $(x_1 + x_2 + \dots + x_k)^n$  este  $\frac{n!}{n_1! \dots n_k!}$  unde  $\forall 1 \leq i \leq k, 0 \leq n_i \leq n$  și  $n_1 + n_2 + \dots + n_k = n$ . Considerând dezvoltarea  $(a + 2b - 3c)^5$  să se determine coeficientul lui  $ab^2c^2$ .
  - a) 128
  - b) 270
  - c) 720
  - d) 120
  - e) 1080
2. Pentru  $n$  și  $k$  întregi pozitivi, coeficientul lui  $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$  din dezvoltarea  $(x_1 + x_2 + \dots + x_k)^n$  este  $\frac{n!}{n_1! \dots n_k!}$  unde  $\forall 1 \leq i \leq k, 0 \leq n_i \leq n$  și  $n_1 + n_2 + \dots + n_k = n$ . Considerând dezvoltarea  $(a + 2b - 3c)^5$  să se determine coeficientul lui  $b^3c^2$ .
  - a) -270
  - b) 720
  - c) 1080
  - d) 120
  - e) -540
3. Pentru  $n$  și  $k$  întregi pozitivi, coeficientul lui  $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$  din dezvoltarea  $(x_1 + x_2 + \dots + x_k)^n$  este  $\frac{n!}{n_1! \dots n_k!}$  unde  $\forall 1 \leq i \leq k, 0 \leq n_i \leq n$  și  $n_1 + n_2 + \dots + n_k = n$ . Considerând dezvoltarea  $(a + 2b - 3c)^5$  să se determine coeficientul lui  $a^2c^3$ .
  - a) -270
  - b) -1080
  - c) 720
  - d) 120
  - e) 270
4. Pentru  $n$  și  $k$  întregi pozitivi, coeficientul lui  $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$  din dezvoltarea  $(x_1 + x_2 + \dots + x_k)^n$  este  $\frac{n!}{n_1! \dots n_k!}$  unde  $\forall 1 \leq i \leq k, 0 \leq n_i \leq n$  și  $n_1 + n_2 + \dots + n_k = n$ . Considerând dezvoltarea  $(a + 2b - 3c)^5$  să se determine coeficientul lui  $abc^3$ .
  - a) 720
  - b) -120
  - c) 120
  - d) -1080
  - e) 270
5. Pentru  $n$  și  $k$  întregi pozitivi, coeficientul lui  $x_1^{n_1} x_2^{n_2} \dots x_k^{n_k}$  din dezvoltarea  $(x_1 + x_2 + \dots + x_k)^n$  este  $\frac{n!}{n_1! \dots n_k!}$  unde  $\forall 1 \leq i \leq k, 0 \leq n_i \leq n$  și  $n_1 + n_2 + \dots + n_k = n$ . Considerând dezvoltarea  $(a + 2b - 3c)^5$  să se determine coeficientul lui  $a^3bc$ .
  - a) 540
  - b) -540
  - c) -120
  - d) 720
  - e) 1080
6. Care este numărul total de cuvinte diferite ce se pot obține prin permutarea literelor cuvântului "IANINA" ?
  - a) 720

- b) 360  
 c) 45  
 d) 90  
 e) 60
7. Care este numărul total de cuvinte diferite ce se pot obține prin permutarea literelor cuvântului "MARIA" ?
- a) 60  
 b) 120  
 c) 45  
 d) 90  
 e) 12
8. Care este numărul total de cuvinte diferite ce se pot obține prin permutarea literelor cuvântului "PAPPER" ?
- a) 90  
 b) 120  
 c) 720  
 d) 45  
 e) 60
9. Care este numărul total de cuvinte diferite ce se pot obține prin permutarea literelor cuvântului "BRAMBURA" ?
- a) 90  
 b) 5040  
 c) 720  
 d) 40640  
 e) 20320
10. Care este numărul total de cuvinte diferite ce se pot obține prin permutarea literelor cuvântului "HIMALAIA" ?
- a) 180  
 b) 3200  
 c) 40640  
 d) 3360  
 e) 5040

11. Indicați cu cine este expresia logica

$$\neg[\neg[(p \vee q) \wedge r] \vee \neg q]$$

echivalentă ?

- a)  $p \wedge q$   
 b)  $\neg q \wedge r$   
 c)  $p \wedge \neg q$   
 d)  $p \wedge r$   
 e)  $q \wedge r$
12. Indicați cu cine este echivalentă expresia logică?

$$\neg[\neg[(p \wedge q) \vee r] \vee \neg q]$$

- a)  $(p \vee r) \wedge q$   
 b)  $q \wedge q$

- c)  $q \wedge \neg r \wedge p$
- d)  $(p \wedge \neg q) \vee (p \wedge r)$
- e)  $(q \wedge r) \vee (q \wedge \neg r)$

13. Indicați cu cine este echivalentă expresia logică

$$\neg[\neg[(p \vee \neg q) \wedge r] \vee q]$$

- a)  $q \wedge r$
- b)  $p \wedge r$
- c)  $\neg q \wedge r$
- d)  $p \wedge \neg q$
- e)  $p \wedge q$

14. Indicați cu cine este echivalentă expresia logică

$$\neg[\neg[(p \wedge \neg q) \vee r] \vee q]$$

- a)  $p \wedge \neg q$
- b)  $(q \wedge r) \vee (\neg q \wedge p \wedge r)$
- c)  $\neg(q \wedge r) \vee (\neg q \wedge p \wedge r)$
- d)  $(q \wedge \neg r) \vee (\neg q \wedge \neg p \wedge r)$
- e)  $(p \wedge r) \vee (\neg q)$

15. Indicați cu cine este echivalentă expresia logică

$$\neg[\neg[(p \vee q) \wedge r] \vee \neg p]$$

- a)  $q \wedge r$
- b)  $p \wedge q$
- c)  $\neg q \wedge r$
- d)  $p \wedge r$
- e)  $p \wedge \neg q$

16. Considerând secvența de instrucțiuni

```
char x;
.....
x=-25;
x>>=2;
```

care din următoarele numere în baza 16 reprezintă valoarea lui  $x$  după execuție?

- a) e8
- b) f9
- c) d9
- d) 13
- e) e6

17. Considerând secvența de instrucțiuni

```
char x;
.....
x=-32;
x>>=3;
```

care din următoarele numere în baza 16 reprezintă valoarea lui  $x$  după execuție?

- a) e7
- b) f9
- c) fe
- d) d3
- e) fc

18. Considerând secvența de instrucțiuni

```
char x;
.....
x=-105;
x>>=2;
```

care din următoarele numere în baza 16 reprezintă valoarea lui  $x$  după execuție?

- a) f2
- b) e5
- c) d9
- d) 19
- e) f6

19. Considerând secvența de instrucțiuni

```
char x;
.....
x=-9;
x>>=3;
```

care din următoarele numere în baza 16 reprezintă valoarea lui  $x$  după execuție?

- a) fe
- b) 15
- c) de
- d) e5
- e) f6

20. Considerând secvența de instrucțiuni

```
char x;
.....
x=-50;
x>>=2;
```

care din următoarele numere în baza 16 reprezintă valoarea lui  $x$  după execuție?

- a) f3
- b) f9
- c) d9
- d) 13
- e) e6

21. Care este spațiul ocupat de variabila uval?

```
union u{
 unsigned char c[4];
 int ival;
 float fval;
 char *pval;
} uval;
```

- a) 1 octet
- b) 2 octeți
- c) 3 octeți
- d) 4 octeți
- e) 5 octeți

22. Care este spațiul ocupat de variabila uval (model LARGE)?

```
union u{
 unsigned char c[5];
 int ival[2];
 float fval;
 char *pval;
} uval;
```

- a) 8 octeți
- b) 2 octeți
- c) 3 octeți
- d) 4 octeți
- e) 5 octeți

23. Care este spațiul ocupat de variabila uval (model LARGE)?

```
union u{
 unsigned char c[3];
 int ival;
 float fval;
 char *pval[2];
} uval;
```

- a) 1 octet
- b) 3 octeți
- c) 8 octeți
- d) 4 octeți
- e) 12 octeți

24. Care este spațiul ocupat de variabila uval?

```
union u{
 unsigned char c;
 int ival[4];
 float fval[3];
 char *pval;
} uval;
```

- a) 1 octet
- b) 8 octeți
- c) 3 octeți



- d) 4 octeți
- e) 12 octeți

25. Care este spațiul ocupat de variabila uval?

```
union u{
 unsigned char c[8];
 int ival[4];
 float fval;
 char *pval;
} uval;
```

- a) 1 octet
- b) 8 octeți
- c) 3 octeți
- d) 4 octeți
- e) 12 octeți

26. Ce va afișa programul?

```
type long=array[1..2] of byte;
var b:word;
a:^long;
begin
 b:=1;
 a:=@b;
 a^[1]:=a^[1] + a^[2];
 a^[2]:=a^[1] - a^[2];
 a^[1]:=a^[1] - a^[2];
 writeln(b);
end.
```

- a) 231
- b) 16
- c) 256
- d) 1
- e) 257

27. Ce va afișa programul?

```
type long=array[1..2] of byte;
var b:word;
a:^long;
begin
 b:=256;
 a:=@b;
 a^[2]:=a^[1]+a^[2];
 a^[1]:=a^[2]+a^[1];
 writeln(b);
end.
```

- a) 0
- b) 257
- c) 256
- d) 1

e) 65535

28. Ce va afișa programul?

```
type long=array[1..2] of byte;
var b:word;
 a:^long;
begin
 b:=$1010;
 a:=@b;
 a^[2]:=a^[2] div a^[1];
 a^[1]:=a^[1]*a^[2];
 writeln(b);
end.
```

a) 1

b) 16

c) 256

d) 272

e) 1024

29. Ce va afișa programul?

```
type long=array[1..2] of byte;
var b:word;
 a:^long;
begin
 b:=272;
 a:=@b;
 a^[1]:=a^[1] + a^[2];
 b:=b-16;
 a^[2]:=a^[1] + a^[2];
 writeln(b);
end.
```

a) 513

b) 531

c) 256

d) 257

e) 272

30. Ce va afișa programul?

```
type long=array[1..2] of byte;
var b:word;
 a:^long;
begin
 b:=1;
 a:=@b;
 a^[1]:=a^[1] - a^[2];
 a^[2]:=a^[1] - a^[2];
 a^[1]:=a^[1] + a^[2];
 writeln(b);
end.
```

a) 258

- b) 16
- c) 256
- d) 2
- e) 257

31. Un segment de program Pascal conține următoarea buclă Repeat-Until:

```
Repeat
.....
Until ((x<>0) and (y>0)) or (not((w>0) and (t=3)));
```

Pentru care din următoarele seturi de valori ale variabilelor  $x, y, w$  și  $t$ , bucla se termină:

- I.  $x = 7, y = 2, w = 5, t = 3$
- II.  $x = 0, y = -1, w = 1, t = 3$
- III.  $x = 0, y = 2, w = -3, t = 4$
- IV.  $x = 1, y = -1, w = 1, t = 3$ .

- a) I și II
- b) II și III
- c) I și III
- d) II și IV
- e) III și IV

32. Un segment de program Pascal conține următoarea buclă Repeat-Until:

```
Repeat
....
Until ((x>0) or (y>0)) and (not((w>0) and (t=3)));
```

Pentru care din următoarele seturi de valori ale variabilelor  $x, y, w$  și  $t$ , bucla se termină:

- I.  $x = 7, y = 2, w = 5, t = 3$
- II.  $x = 0, y = -1, w = 1, t = 3$
- III.  $x = 0, y = 2, w = -3, t = 4$
- IV.  $x = 1, y = -1, w = 1, t = 3$ .

- a) I și II
- b) II și III
- c) I și III
- d) II
- e) III

33. Un segment de program Pascal conține următoarea buclă Repeat-Until:

```
Repeat
.....
Until ((x+y>=0) or (y-x>0)) and (not((w<0) and (t<>3)));
```

Pentru care din următoarele seturi de valori ale variabilelor  $x, y, w$  și  $t$ , bucla se termină:

- I.  $x = 7, y = 2, w = 5, t = 3$
- II.  $x = 0, y = -1, w = 1, t = 3$
- III.  $x = 0, y = 2, w = -3, t = 4$
- IV.  $x = 1, y = -1, w = 1, t = 3$ .

- a) I și II
- b) II și III
- c) I și IV
- d) II și IV
- e) III și IV

34. Un segment de program Pascal conține următoarea buclă Repeat-Until:

```
Repeat
.....
Until ((x*y>0) and (y+t>0)) or ((w<0) and (t<>3));
```

Pentru care din următoarele seturi de valori ale variabilelor  $x, y, w$  și  $t$ , bucla se termină:

- I.  $x = 0, y = -1, w = 1, t = 3$
- II.  $x = 7, y = 2, w = 5, t = 3$
- III.  $x = 1, y = -1, w = 1, t = 3$
- IV.  $x = 0, y = 2, w = -3, t = 4$ .

- a) I și II
- b) II și III
- c) I și III
- d) II și IV
- e) III și IV

35. Un segment de program Pascal conține următoarea buclă Repeat-Until:

```
Repeat
.....
Until ((x mod w>0) and (y div x>0)) or ((w*t<0) and (t<>3));
```

Pentru care din următoarele seturi de valori ale variabilelor  $x, y, w$  și  $t$ , bucla se termină:

- I.  $x = 7, y = 2, w = 5, t = 3$
- II.  $x = 0, y = -1, w = 1, t = 3$
- III.  $x = 0, y = 2, w = -3, t = 4$
- IV.  $x = 1, y = -1, w = 1, t = 3$ .

- a) I și II
- b) II
- c) IV
- d) III
- e) III și IV

36. Se dă următoarea secvență de program Pascal:

```
n:=99;i:=1;
while i<= n do
begin
i:=i+2;
n:=n-1;
end;
```

Care va fi valoarea variabilei  $n$  la terminarea secvenței, știind că  $i$  și  $n$  sunt de tip integer.

- a) 1

- b) 51
- c) 99
- d) 66
- e) 49

37. Se dă următoarea secvență de program Pascal:

```
n:=99;i:=1;
while i<= n do
begin
 i:=i+3;
 n:=n-1;
end;
```

Care va fi valoarea variabilei  $n$  la terminarea secvenței, știind că  $i$  și  $n$  sunt de tip integer.

- a) 33
- b) 74
- c) 32
- d) 66
- e) 49

38. Se dă următoarea secvență de program Pascal:

```
n:=99;i:=1;
while i<= n do
begin
 i:=i+2;
 n:=n-2;
end;
```

Care va fi valoarea variabilei  $n$  la terminarea secvenței, știind că  $i$  și  $n$  sunt de tip integer.

- a) 74
- b) 51
- c) 33
- d) 66
- e) 49

39. Se dă următoarea secvență de program Pascal:

```
n:=99;i:=1;
while i<= n do
begin
 i:=i+1;
 n:=n-2;
end;
```

Care va fi valoarea variabilei  $n$  la terminarea secvenței, știind că  $i$  și  $n$  sunt de tip integer.

- a) 33
- b) 55
- c) 69
- d) 66
- e) 49

40. Se dă următoarea secvență de program Pascal:

```
n:=99;i:=1;
while i<= n do
begin
 i:=i+5;
 n:=n-2;
end;
```

Care va fi valoarea variabilei  $n$  la terminarea secvenței, știind că  $i$  și  $n$  sunt de tip integer.

- a) 74
- b) 69
- c) 33
- d) 66
- e) 49

41. Se dă următorul program :

```
#include <stdio.h>
int x,y,z;
void p(int *x,int y)
{
 int z;
 ::z=*x;
 *x=z=y+1;
 y=2*z;
}
void main(){
 x=3;y=6;z=2;
 p(&y,x);
 printf("%d %d %d",x,y,z);
}
```

Ce va afișa?

- a) 7 4 6
- b) 3 4 2
- c) 3 4 6
- d) 2 6 4
- e) 3 6 4

42. Se dă următorul program:

```
#include <stdio.h>
int x,y,z;
void p(int *x,int y)
{
 int z;
 ::z=*x;
 *x=z=y+1;
 y=2*z;
}
void main(){
 x=6;y=2;z=3;
 p(&y,x);
 printf("%d %d %d",x,y,z);
}
```

Ce va afișa?

- a) 6 7 2
- b) 2 6 7
- c) 7 2 4
- d) 2 6 4
- e) 3 4 6

43. Se dă următorul program :

```
#include <stdio.h>
int x,y,z;
void p(int *x,int y)
{
 int z;
 ::z=*x;
 *x=z=y+1;
 y=2*z;
}
void main(){
 x=5;y=3;z=6;
 p(&y,x);
 printf("%d %d %d",x,y,z);
}
```

Ce va afișa?

- a) 3 5 6
- b) 3 7 2
- c) 3 12 6
- d) 5 6 3
- e) 3 6 4

44. Se dă următorul program:

```
#include <stdio.h>
int x,y,z;
void p(int *x,int y)
{
 int z;
 ::z=*x;
 *x=z=y+1;
 y=2*z;
}
void main(){
 x=3;y=7;z=5;
 p(&y,x);
 printf("%d %d %d",x,y,z);
}
```

Ce va afișa?

- a) 3 14 3
- b) 3 7 4
- c) 3 4 6
- d) 2 6 4

e) 3 4 7

45. Se dă următorul program:

```
#include <stdio.h>
int x,y,z;
void p(int *x,int y)
{
 int z;
 ::z=*x;
 *x=z=y+1;
 y=2*z;
}
void main(){
 x=7;y=8;z=1;
 p(&y,x);
 printf("%d %d %d",x,y,z);
}
```

Ce va afișa?

- a) 7 6 2
- b) 1 8 7
- c) 7 8 8
- d) 2 6 4
- e) 3 6 4

46. Se dă un vector cu  $n$  componente la început nule. Pentru  $1 \leq i < \lfloor \frac{i+j}{2} \rfloor < j \leq n$  definim "secționarea" vectorului corespunzător perechii de indici  $(i, j)$  astfel :

Pas 1. Incrementarea cu  $p$  unități a elementelor aflate între pozițiile  $i$  și  $k$ .

Pas 2. "Secționarea" corespunzătoare perechii  $(i, k)$ .

Pas 3. Incrementarea cu  $q$  unități a elementelor aflate între pozițiile  $k$  și  $j$ .

Pas 4. "Secționarea" corespunzătoare perechii  $(k, j)$ .

unde  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Pentru  $n = 7, p = 1, q = 2$  să se indice care este vectorul rezultat în urma secționării corespunzătoare perechii  $(1, n)$ .

- a) (2, 5, 6, 7, 5, 7, 6)
- b) (1, 5, 6, 8, 6, 7, 5)
- c) (2, 5, 6, 8, 6, 7, 6)
- d) (2, 5, 7, 8, 6, 7, 7)
- e) (2, 4, 6, 7, 5, 6, 5)

47. Se dă un vector cu  $n$  componente la început nule. Pentru  $1 \leq i < \lfloor \frac{i+j}{2} \rfloor < j \leq n$  definim "secționarea" vectorului corespunzător perechii de indici  $(i, j)$  astfel :

Pas 1. Incrementarea cu  $p$  unități a elementelor aflate între pozițiile  $i$  și  $k$ .

Pas 2. "Secționarea" corespunzătoare perechii  $(i, k)$ .

Pas 3. Incrementarea cu  $q$  unități a elementelor aflate între pozițiile  $k$  și  $j$ .

Pas 4. "Secționarea" corespunzătoare perechii  $(k, j)$ .

unde  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Pentru  $n = 6, p = 3, q = 1$  să se indice care este vectorul rezultat în urma secționării corespunzătoare perechii  $(1, n)$ .

- a) (6, 7, 8, 8, 6, 3)



- b) (5, 6, 7, 7, 5, 2)
- c) (6, 7, 6, 6, 7, 4)
- d) (6, 7, 6, 6, 5, 3)
- e) (6, 7, 8, 8, 5, 4)

48. Se dă un vector cu  $n$  componente la început nule. Pentru  $1 \leq i < \lfloor \frac{i+j}{2} \rfloor < j \leq n$  definim "secționarea" vectorului corespunzător perechii de indici  $(i, j)$  astfel :

Pas 1. Incrementarea cu  $p$  unități a elementelor aflate între pozițiile  $i$  și  $k$ .

Pas 2. "Secționarea" corespunzătoare perechii  $(i, k)$ .

Pas 3. Incrementarea cu  $q$  unități a elementelor aflate între pozițiile  $k$  și  $j$ .

Pas 4. "Secționarea" corespunzătoare perechii  $(k, j)$ .

unde  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Pentru  $n = 6, p = 1, q = 2$  să se indice care este vectorul rezultat în urma secționării corespunzătoare perechii  $(1, n)$ .

- a) (2, 4, 5, 5, 7, 6)
- b) (2, 4, 6, 6, 7, 6)
- c) (1, 4, 6, 7, 6, 5)
- d) (2, 5, 6, 6, 8, 7)
- e) (2, 4, 5, 5, 6, 6)

49. Se dă un vector cu  $n$  componente la început nule. Pentru  $1 \leq i < \lfloor \frac{i+j}{2} \rfloor < j \leq n$  definim "secționarea" vectorului corespunzător perechii de indici  $(i, j)$  astfel :

Pas 1. Incrementarea cu  $p$  unități a elementelor aflate între pozițiile  $i$  și  $k$ .

Pas 2. "Secționarea" corespunzătoare perechii  $(i, k)$ .

Pas 3. Incrementarea cu  $q$  unități a elementelor aflate între pozițiile  $k$  și  $j$ .

Pas 4. "Secționarea" corespunzătoare perechii  $(k, j)$ .

unde  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Pentru  $n = 7, p = 3, q = 1$  să se indice care este vectorul rezultat în urma secționării corespunzătoare perechii  $(1, n)$ .

- a) (6, 8, 8, 9, 8, 6, 4)
- b) (6, 10, 9, 10, 8, 6, 3)
- c) (6, 10, 8, 9, 7, 6, 4)
- d) (6, 10, 8, 9, 8, 6, 3)
- e) (5, 9, 8, 9, 8, 7, 3)

50. Se dă un vector cu  $n$  componente la început nule. Pentru  $1 \leq i < \lfloor \frac{i+j}{2} \rfloor < j \leq n$  definim "secționarea" vectorului corespunzător perechii de indici  $(i, j)$  astfel :

Pas 1. Incrementarea cu  $p$  unități a elementelor aflate între pozițiile  $i$  și  $k$ .

Pas 2. "Secționarea" corespunzătoare perechii  $(i, k)$ .

Pas 3. Incrementarea cu  $q$  unități a elementelor aflate între pozițiile  $k$  și  $j$ .

Pas 4. "Secționarea" corespunzătoare perechii  $(k, j)$ .

unde  $k = \lfloor \frac{i+j}{2} \rfloor$ .

Pentru  $n = 7, p = 2, q = 3$  să se indice care este vectorul rezultat în urma secționării corespunzătoare perechii  $(1, n)$ .

- a) (4, 8, 9, 12, 9, 10, 8)
- b) (4, 7, 9, 11, 9, 8, 7)
- c) (6, 10, 11, 14, 11, 13, 10)

d) (4, 10, 11, 15, 11, 12, 10)

e) (4, 9, 10, 13, 10, 11, 9)

51. Ce va afișa programul?

```
#include <stdio.h>
char s[5],*f,*p,*m=s+5;
void push(int i){
 *p=i+'0';p++;
 if (p>=m) p=s;
}
char pop(){
 char c=*f;
 f++;
 if (f>=m) f=s;
 return c;
}
void main(){
 int i;
 p=f=s;
 for(i=1;i<=6;i++) push(i);
 printf(''%c'',pop());
 push(i);
 for(i=0;i<5;i++) printf(''%c'', s[i]);
}
```

a) 6 6 7 3 4 5

b) 6 7 3 4 5 6

c) 6 7 6 3 4 5

d) 6 7 6 5 4 3

e) 3 4 5 6 7 6

52. Ce va afișa programul?

```
#include <stdio.h>
char s[5],*f,*p,*m=s+5;
void push(int i){
 *p=i+'0';p++;
 if (p>=m) p=s;
}
char pop(){
 char c=*f;
 f++;
 if (f>=m) f=s;
 return c;
}
void main(){
 int i;
 p=f=s;
 for(i=1;i<=8;i++) push(i);
 printf(''%c'',pop());
 push(i);
 for(i=0;i<5;i++) printf(''%c'', s[i]);
}
```

a) 6 6 7 8 5 9

b) 6 6 7 8 9 5

- c) 7 8 9 5 6 6
- d) 5 6 7 8 9 6
- e) 6 7 6 7 8 5

53. Ce va afișa programul?

```
#include <stdio.h>
char s[5],*f,*p,*m=s+5;
void push(int i){
 *p=i+'0';p++;
 if (p>=m) p=s;
}
char pop(){
 char c=*f;
 f++;
 if (f>=m) f=s;
 return c;
}
void main(){
 int i;
 p=f=s;
 for(i=1;i<=3;i++) push(i);
 printf(''%c'',pop());
 push(i);
 for(i=0;i<5;i++) printf(''%c'', s[i]);
}
```

- a) 1 1 2 3 4
- b) 1 2 3 4 1
- c) 4 3 2 1 1
- d) 1 2 3 4 5 6
- e) 1 1 2 3 4 5

54. Ce va afișa programul?

```
#include <stdio.h>
char s[5],*f,*p,*m=s+5;
void push(int i){
 *p=i+'0';p++;
 if (p>=m) p=s;
}
char pop(){
 char c=*f;
 f++;
 if (f>=m) f=s;
 return c;
}
void main(){
 int i;
 p=f=s;
 for(i=1;i<=5;i++) push(i);
 printf(''%c'',pop());
 push(i);
 for(i=0;i<5;i++) printf(''%c'', s[i]);
}
```

- a) 1 2 3 4 5 6

- b) 6 1 2 3 4 5
- c) 6 5 4 3 2 1
- d) 1 6 2 3 4 5
- e) 1 1 2 3 4 5

55. Ce va afișa programul?

```
#include <stdio.h>
char s[5],*f,*p,*m=s+5;
void push(int i){
 *p=i+'0';p++;
 if (p>=m) p=s;
}
char pop(){
 char c=*f;
 f++;
 if (f>=m) f=s;
 return c;
}
void main(){
 int i;
 p=f=s;
 for(i=1;i<=7;i++) push(i);
 printf(''%c'',pop());
 push(i);
 for(i=0;i<5;i++) printf(''%c'', s[i]);
}
```

- a) 6 6 7 3 4 5
- b) 6 6 5 4 7 8
- c) 6 6 7 8 4 5
- d) 1 2 3 4 5 6
- e) 8 7 6 5 4 6

56. Ce va afișa programul?

```
#include <stdio.h>
int s[5],*t,*m=s+5;
void push(int i){
 if (t<m-1){
 t++;*t=i;
 }
}
int pop(void){
 int a;
 if (t>=s){
 a=*t;t--;
 }
 else a=-1;
 return a;
}
void main(void){
 int i;
 t=s-1;
 for (i=1;i<=9;i++) push(i);
 for (;pop(>0););
}
```

```

push(11);push(12);
for (i=0;i<5;i++) printf("%d ",s[i]);
}

```

- a) 1 2 3 4 5 1
- b) 11 12 3 4 5
- c) 11 12 1 2 3
- d) 5 6 7 8 9
- e) 7 8 9 11 12

57. Ce va afișa programul?

```

#include <stdio.h>
int s[5],*t,*m=s+5;
void push(int i){
 if (t<m-1){
 t++;*t=i;
 }
}
int pop(void){
 int a;
 if (t>=s){
 a=*t;t--;
 }
 else a=-1;
 return a;
}
void main(void){
 int i;
 t=s-1;
 for (i=1;i<=9;i++) push(i);
 pop();pop();
 push(1);push(2);
 for (i=0;i<5;i++) printf("%d ",s[i]);
}

```

- a) 1 2 3 2 1
- b) 1 2 1 2 3
- c) 1 2 3 1 2
- d) 3 2 1 2 1
- e) 1 2 3 4 5

58. Ce va afișa programul?

```

#include <stdio.h>
int s[5],*t,*m=s+5;
void push(int i){
 if (t<m-1){
 t++;*t=i;
 }
}
int pop(void){
 int a;
 if (t>=s){
 a=*t;t--;
 }
}

```

```

 else a=-1;
 return a;
}
void main(void){
 int i,x;
 t=s-1;
 for (i=1;i<=9;i++) push(i);
 for(;(x=pop())>0;) printf("%d ",x);
}

```

- a) 1 2 3 4 5
- b) 2 1 3 4 5
- c) 4 5 6 7 8
- d) 5 4 3 2 1
- e) 5 4 3 1 2

59. Ce va afișa programul?

```

#include <stdio.h>
int s[5],*t,*m=s+5;
void push(int i){
 if (t<m-1){
 t++;*t=i;
 }
}
int pop(void){
 int a;
 if (t>=s){
 a=*t;t--;
 }
 else a=-1;
 return a;
}
void main(void){
 int i,x;
 t=s-1;
 for (i=1;i<=3;i++) push(i);
 printf("%d ",pop());
 push(2);push(1);
 for (i=0;i<4;i++) printf("%d ",s[i]);
}

```

- a) 3 1 2 2 1
- b) 1 2 3 2 1
- c) 3 2 1 2 1
- d) 1 2 3 1 2
- e) 2 1 3 2 1

60. Ce va afișa programul?

```

#include <stdio.h>
int s[5],*t,*m=s+5;
void push(int i){
 if (t<m-1){
 t++;*t=i;
 }
}

```

```

}
int pop(void){
 int a;
 if (t>=s){
 a=*t;t--;
 }
 else a=-1;
 return a;
}
void main(void){
 int i,x;
 t=s-1;
 for (i=1;i<=9;i++) push(i);
 for (i=1;i<=9;i++)
 if (i%2==1) printf("%d ",pop());
 else push(i);
}

```

- a) 8 6 4 2 5
- b) 1 2 3 4 5
- c) 4 2 5 6 8
- d) 2 5 4 6 8
- e) 5 2 4 6 8

61. Ce va afișa programul?

```

#include <stdio.h>
#define n 7
int x[n]={7,6,5,4,3,2,1};
void main(void){
 int k=0;
 int i,tmp,ok;
 do{
 ok=0;
 for (i=0;i<n-1;i++)
 if (x[i]>x[i+1]){
 tmp=x[i];x[i]=x[i+1];x[i+1]=tmp;ok=1;
 }
 k++;
 }while (ok && k!=2);
 for (i=0;i<n;i++) printf("%d ",x[i]);
}

```

- a) 1 2 3 4 5 6 7
- b) 6 7 3 4 5 2 1
- c) 5 4 3 2 1 6 7
- d) 4 3 2 1 5 6 7
- e) 2 1 4 3 6 5 7

62. Ce va afișa programul?

```

#include <stdio.h>
#define n 7
int x[n]={7,5,3,1,2,4,6};
void main(void){
 int k=0;

```

```

int i,tmp,ok;
do{
 ok=0;
 for (i=0;i<n-1;i++)
 if (x[i]>x[i+1]){
 tmp=x[i];x[i]=x[i+1];x[i+1]=tmp;ok=1;
 }
 k++;
}while (ok && k!=2);
for (i=0;i<n;i++) printf("%d ",x[i]);
}

```

- a) 3 1 2 4 5 6 7
- b) 6 7 3 4 5 2 1
- c) 5 4 3 2 1 6 7
- d) 4 3 2 1 5 6 7
- e) 2 1 4 3 6 5 7

63. Ce va afișa programul?

```

#include <stdio.h>
#define n 7
int x[n]={1,2,3,7,6,5,4};
void main(void){
 int k=0;
 int i,tmp,ok;
 do{
 ok=0;
 for (i=0;i<n-1;i++)
 if (x[i]>x[i+1]){
 tmp=x[i];x[i]=x[i+1];x[i+1]=tmp;ok=1;
 }
 k++;
 }while (ok && k!=3);
 for (i=0;i<n;i++) printf("%d ",x[i]);
}

```

- a) 3 4 2 1 5 6 7
- b) 1 2 3 4 5 6 7
- c) 5 4 3 2 1 6 7
- d) 4 3 2 1 5 6 7
- e) 2 1 4 3 6 5 7

64. Ce va afișa programul?

```

#include <stdio.h>
#define n 7
int x[n]={2,1,7,6,5,4,3};
void main(void){
 int k=0;
 int i,tmp,ok;
 do{
 ok=0;
 for (i=0;i<n-1;i++)
 if (x[i]>x[i+1]){
 tmp=x[i];x[i]=x[i+1];x[i+1]=tmp;ok=1;
 }
 }
}

```



```

 }
 k++;
}while (ok && k!=2);
for (i=0;i<n;i++) printf("%d ",x[i]);
}

```

- a) 1 2 3 4 5 6 7
- b) 6 7 3 4 5 2 1
- c) 5 4 3 2 1 6 7
- d) 1 2 5 4 3 6 7
- e) 2 1 4 3 6 5 7

65. Ce va afișa programul?

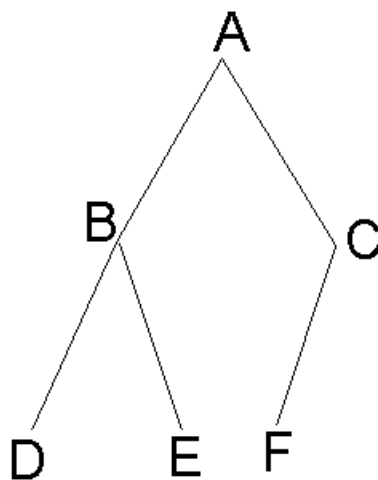
```

#include <stdio.h>
#define n 7
int x[n]={2,1,3,5,6,4,7};
void main(void){
 int k=0;
 int i,tmp,ok;
 do{
 ok=0;
 for (i=0;i<n-1;i++)
 if (x[i]>x[i+1]){
 tmp=x[i];x[i]=x[i+1];x[i+1]=tmp;ok=1;
 }
 k++;
 }while (ok && k!=3);
 for (i=0;i<n;i++) printf("%d ",x[i]);
}

```

- a) 2 1 3 5 6 4 7
- b) 6 7 3 4 5 2 1
- c) 5 4 3 2 1 6 7
- d) 4 3 2 1 5 6 7
- e) 1 2 3 4 5 6 7

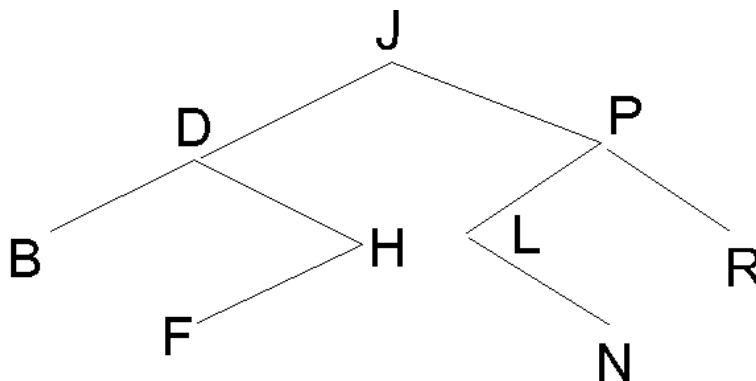
66. O procedură care afișează arborele binar



în postordine va tipări:

- a) A B C D E F
- b) A B D E C F
- c) D B E A F C
- d) D E B F C A
- e) D E F B C A

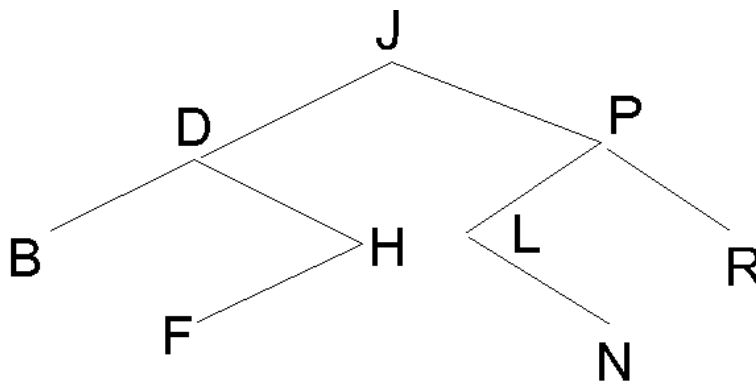
67. O procedură care afișează arborele binar



în preordine va tipări:

- a) J D P B H L R F N
- b) J D B H F P L N R
- c) N F R L H B P D J
- d) B D H F N L P R J
- e) B D F H L J N P R

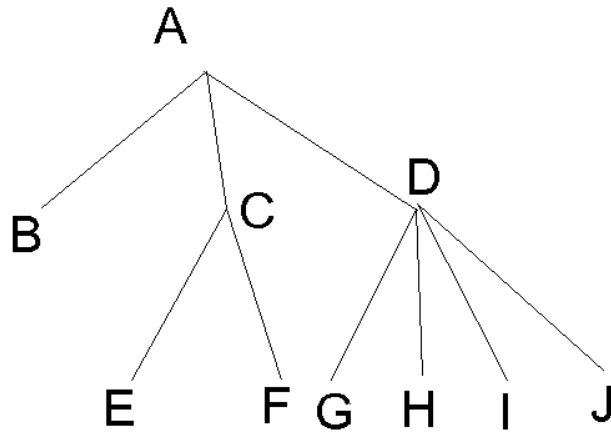
68. O procedură care afișează arborele binar



în inordine va tipări:

- a) B H F D L N P R J
- b) B D H F N L P R J
- c) N F R L H B P D J
- d) J D P B H L R F N
- e) B D F H J L N P R

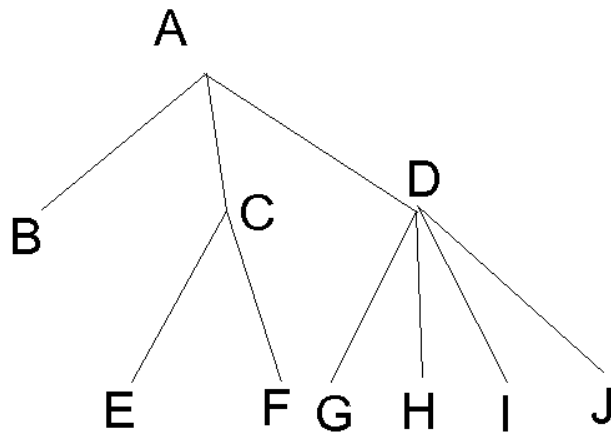
69. O procedură care afișează arborele



în A-preordine va tipări:

- a) A B C E F D G H I J
- b) A B C D E F G H I J
- c) B E F C G H I J D A
- d) B A E C F G D H I J
- e) B E F G H I J C D A

70. O procedură care afișează arborele



în A-postordine va tipări:

- a) A B C D E F G H I J
- b) D C B J I H G F E A
- c) A B C E F G H I J
- d) B E F C G H I J D A
- e) B A E C F G D H I J

71. Un mijloc de transport poate transporta o încărcătură cu greutatea maximă  $G$ . Fie  $O_i, i = 1, \dots, n$  caracterizate de greutatea lor  $g_i$  și câștigurile  $c_i$ ,  $i = 1, \dots, n$  (pentru orice  $i$ , valoarea  $c_i$  reprezintă câștigul obținut prin transportul obiectului  $O_i$  în întregime). Un obiect poate fi transportat în întregime, fracționat sau deloc. Dacă  $G = 11, n = 5, g = (6, 7, 3, 2, 1), c = (4, 2, 9, 1, 1)$  să se indice, pentru acest transport, valoarea câștigului maxim .

- a)  $\frac{27}{2}$
- b)  $\frac{28}{3}$
- c) 17
- d)  $\frac{29}{2}$

- e)  $\frac{25}{6}$
72. Un mijloc de transport poate transporta o încărcătură cu greutatea maximă  $G$ . Fie  $O_i, i = 1, \dots, n$  caracterizate de greutatea lor  $g_i$  și câștigurile  $c_i$ ,  $i = 1, \dots, n$  (pentru orice  $i$ , valoarea  $c_i$  reprezintă câștigul obținut prin transportul obiectului  $O_i$  în întregime). Un obiect poate fi transportat în întregime, fracționat sau deloc. Dacă  $G = 11, n = 5, g = (6, 10, 5, 4, 4), c = (2, 5, 3, 1, 4)$  să se indice, pentru acest transport, valoarea câștigului maxim .
- a)  $\frac{15}{2}$   
 b)  $8\frac{2}{3}$   
 c) 8  
 d)  $\frac{29}{5}$   
 e) 15
73. Un mijloc de transport poate transporta o încărcătură cu greutatea maximă  $G$ . Fie  $O_i, i = 1, \dots, n$  caracterizate de greutatea lor  $g_i$  și câștigurile  $c_i$ ,  $i = 1, \dots, n$  (pentru orice  $i$ , valoarea  $c_i$  reprezintă câștigul obținut prin transportul obiectului  $O_i$  în întregime). Un obiect poate fi transportat în întregime, fracționat sau deloc. Dacă  $G = 7, n = 4, g = (7, 4, 5, 6), c = (5, 9, 2, 3)$  să se indice, pentru acest transport, valoarea câștigului maxim .
- a)  $\frac{78}{7}$   
 b)  $\frac{49}{3}$   
 c) 19  
 d)  $\frac{76}{7}$   
 e)  $\frac{65}{6}$
74. . Un mijloc de transport poate transporta o încărcătură cu greutatea maximă  $G$ . Fie  $O_i, i = 1, \dots, n$  caracterizate de greutatea lor  $g_i$  și câștigurile  $c_i$ ,  $i = 1, \dots, n$  (pentru orice  $i$ , valoarea  $c_i$  reprezintă câștigul obținut prin transportul obiectului  $O_i$  în întregime). Un obiect poate fi transportat în întregime, fracționat sau deloc. Dacă  $G = 15, n = 4, g = (4, 1, 7, 6), c = (2, 1, 3, 5)$  să se indice, pentru acest transport, valoarea câștigului maxim .
- a)  $\frac{79}{7}$   
 b)  $\frac{67}{6}$   
 c) 11  
 d)  $\frac{29}{4}$   
 e)  $\frac{68}{7}$
75. Un mijloc de transport poate transporta o încărcătură cu greutatea maximă  $G$ . Fie  $O_i, i = 1, \dots, n$  caracterizate de greutatea lor  $g_i$  și câștigurile  $c_i$ ,  $i = 1, \dots, n$  (pentru orice  $i$ , valoarea  $c_i$  reprezintă câștigul obținut prin transportul obiectului  $O_i$  în întregime). Un obiect poate fi transportat în întregime, fracționat sau deloc. Dacă  $G = 10, n = 5, g = (6, 8, 18, 2, 2), c = (4, 7, 6, 1, 2)$  să se indice, pentru acest transport, valoarea câștigului maxim .
- a)  $\frac{27}{2}$   
 b) 9  
 c)  $11\frac{1}{8}$   
 d) 13  
 e)  $\frac{45}{6}$

| Test | Raspuns |
|------|---------|
| 1    | e       |
| 2    | b       |
| 3    | a       |
| 4    | d       |
| 5    | c       |
| 6    | d       |
| 7    | a       |
| 8    | b       |
| 9    | b       |
| 10   | d       |
| 11   | e       |
| 12   | a       |
| 13   | c       |
| 14   | e       |
| 15   | d       |
| 16   | b       |
| 17   | e       |
| 18   | b       |
| 19   | a       |
| 20   | a       |
| 21   | d       |
| 22   | e       |
| 23   | c       |
| 24   | e       |
| 25   | b       |
| 26   | c       |
| 27   | b       |
| 28   | d       |
| 29   | a       |
| 30   | a       |
| 31   | c       |
| 32   | e       |
| 33   | c       |
| 34   | d       |
| 35   | d       |
| 36   | d       |
| 37   | b       |
| 38   | e       |
| 39   | a       |
| 40   | b       |

| Test | Raspuns |
|------|---------|
| 41   | c       |
| 42   | a       |
| 43   | d       |
| 44   | e       |
| 45   | c       |
| 46   | c       |
| 47   | a       |
| 48   | b       |
| 49   | d       |
| 50   | e       |
| 51   | a       |
| 52   | b       |
| 53   | a       |
| 54   | d       |
| 55   | c       |
| 56   | b       |
| 57   | c       |
| 58   | d       |
| 59   | a       |
| 60   | e       |
| 61   | c       |
| 62   | a       |
| 63   | b       |
| 64   | d       |
| 65   | e       |
| 66   | d       |
| 67   | b       |
| 68   | e       |
| 69   | a       |
| 70   | d       |
| 71   | d       |
| 72   | c       |
| 73   | a       |
| 74   | e       |
| 75   | b       |

## 4 Tehnici avansate de programare

1. Se consideră următoarea secvență de program:

```
class complex
{
 double real;
 double imag;
public:
 complex(double x=1.0, double y=20.0){real=x; imag=y;}
 complex(const complex &u)
 {
 real=u.real;
 imag=u.imag;
 }

}
```

Precizați în care situație se crează un obiect anonim:

- a) `complex z1(3.42, -12.9);`
- b) `complex z2=z1;`
- c) `complex z3(z1);`
- d) `complex z1= complex(45.0,0.9);`
- e) `complex z(23.25);`

2. Se consideră următoarea secvență de program:

```
class complex
{
 double real;
 double imag;
public:
 complex(double x=10.0, double y=10.0){real=x; imag=y;}
 complex(const complex &u)
 {
 real=u.real;
 imag=u.imag;
 }

}
```

Precizați în care situație se realizează o copiere a unui obiect în alt obiect:

- a) `complex z1(3.42, -12.9);`
- b) `complex z2=z1;`
- c) `complex z3(1.0,-1.0);`
- d) `complex z(10.7,0.8);`
- e) `complex z(23.25);`

3. Se consideră următoarea secvență de program:

```
class complex
{
 double real;
 double imag;
public:
```

```

 complex(double x=-11.0, double y=-56.90){real=x; imag=y;}
 complex(const complex &u)
 {
 real=u.real;
 imag=u.imag;
 }

}

```

Precizați în care situație se crează un obiect anonim:

- a) complex z1(3.42, -12.9);
- b) complex z1= complex(0.0,0.9);
- c) complex z2=z1;
- d) complex z3(z1);
- e) complex z(23.25);

4. Se consideră următoarea secvență de program:

```

class complex
{
 double re;
 double im;
public:
 complex(double x=-11.0, double y=-56.90){re=x; im=y;}
 complex(const complex &u)
 {
 real=u.re;
 imag=u.im;
 }

}

```

Precizați în situație se utilizează constructorul de copiere:

- a) complex z1(3.4, -12.9);
- b) complex z3(0.0,-10.9);
- c) complex z2(0.0,1.0);
- d) complex z3(z1);
- e) complex z(2.25);

5. Se consideră următoarea secvență de program:

```

class complex
{
 double real;
 double imag;
public:
 complex(double x=-11.0, double y=-56.90){real=x; imag=y;}
 complex(const complex &u)
 {
 real=u.real;
 imag=u.imag;
 }

}

```

Precizați situația în care nu era necesară folosirea unui constructor cu parametri care iau valori în mod implicit:

- a) complex z2(3.42,-12.9);
- b) complex z3(z2);
- c) complex z=z2;
- d) complex z4(z);
- e) complex z5=z4;

6. Se dă secvența de program:

```
class A
{
 int a[3];
public:
 A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
 int &operator[](int i){return a[i];}
};
void main(void)
{
 A ob(1,2,3);
 cout << ob[1];
 ob[1]=25;
 cout<<ob[1];
}
```

Ce se poate afirma despre operator[]()?

- a) produce supraîncărcarea unei funcții;
- b) produce supraîncărcarea unui operator unar;
- c) supraîncarcă operatorul [];
- d) este o funcție membru oarecare a clasei A care nu produce supraîncărcarea unui operator;
- e) reprezintă un operator ternar;

7. Se dă secvența de program:

```
class A
{
 int a[3];
public:
 A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
 int &operator[](int i){return a[i];}
};
void main(void)
{
 A ob(1,2,3);
 cout << ob[1];
 ob[1]=25;
 cout<<ob[1];
}
```

Ce se poate afirma despre operator[]()?

- a) supraîncarcă operatorul ();
- b) supraîncarcă operatorul [];
- c) supraîncarcă un alt operator binar;
- d) nu supraîncarcă operatori;



e) reprezintă un operator ternar;

8. Se dă secvența de program:

```
class A
{
 int a[3];
 public:
 A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
 int &operator[](int i){return a[i];}
};
void main(void)
{
 A ob(1,2,3);
 cout << ob[1];
 ob[1]=25;
 cout<<ob[1];
}
```

Ce se poate afirma despre operator[]()?

- a) este o metoda oarecare a clasei care nu produce supraîncărcarea unor operatori;
- b) inițializează un obiect al clasei;
- c) este o funcție constructor;
- d) supraîncarcă operatorul [];
- e) supraîncarcă o funcție;

9. Se dă secvența de program:

```
class A
{
 int a[3];
 public:
 A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
 int &operator[](int i){return a[i];}
};
void main(void)
{
 A ob(1,2,3);
 cout << ob[1];
 ob[1]=25;
 cout<<ob[1];
}
```

Ce se poate afirma despre operator[]()?

- a) supraîncarcă operatorul [];
- b) supraîncarcă operatorul ();
- c) este o funcție membru oarecare a clasei care nu produce supraîncărcarea unui operator;
- d) supraîncarcă un operator unar;
- e) este un destructor;

10. Se dă secvența de program:

```
class A
{
 int a[3];
```

```

public:
 A(int i, int j, int k){a[0]=i; a[1]=j; a[2]=k;}
 int &operator[](int i){return a[i];}
};
void main(void)
{
 A ob(1,2,3);
 cout << ob[1];
 ob[1]=25;
 cout<<ob[1];
}

```

Ce se poate afirma despre operator[]()?

- a) supraîncarcă operatorul ();
  - b) supraîncarcă un operator unar;
  - c) supraîncarcă operatorul [];
  - d) este un constructor;
  - e) este un destructor;
11. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau funcții friend. Diferența între aceste două posibilități constă în:
- a) lista de parametri;
  - b) obiect returnat;
  - c) precedența operatorilor;
  - d) n-aritatea operatorului;
  - e) alte situații.
12. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau funcții friend. Diferența între aceste două posibilități constă în:
- a) lista de parametri;
  - b) precedența operatorilor;
  - c) operația efectuată;
  - d) obiect returnat;
  - e) alte situații;
13. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau funcții friend. Diferența între aceste două posibilități constă în:
- a) lista de parametri;
  - b) operația efectuată;
  - c) n-aritatea operatorului;
  - d) obiect returnat;
  - e) alte situații;
14. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau funcții friend. Diferența între aceste două posibilități constă în:
- a) n-aritatea operatorului;
  - b) precedența operatorilor;
  - c) asociativitatea operatorilor;
  - d) operația efectuată;
  - e) numărul parametrilor din lista funcției;

15. Supraîncărcarea unor operatori se poate realiza prin funcții operator sau funcții friend. Diferența între aceste două posibilități constă în:

- a) n-aritatea operatorului;
- b) lista de parametri;
- c) obiect returnat;
- d) operația efectuată;
- e) asociativitatea operatorilor;

16. În secvența de program:

```
.....
int k=100;
void fct()
{
 int k;

 k++;

}
void gct()
{
 int k=2;

 ::k++; // (?)

}
```

În instrucțiunea marcată cu (?), k este o variabilă:

- a) externă;
- b) statică;
- c) registru;
- d) globală;
- e) automatică;

17. În secvența de program:

```
.....
int k=90;
void fct()
{
 int k=8;

 k++; (?)

}
void gct()
{
 int k=9;

 ::k++;

}
```

În instrucțiunea marcată cu (?), variabila k este:

- a) locală;

- b) statică;
- c) registru;
- d) globală;
- e) altă situație;

18. În secvența de program:

```

.....
int k;
void fct()
{
 static int k;

 k++;

}
void gct()
{
 int k; // (?)

 ::k++;

}

```

În declararea marcată cu (?), variabila k este o variabilă:

- a) locală;
- b) pointer;
- c) registru;
- d) globală;
- e) altă situație;

19. În secvența de program:

```

.....
int k=32;
void fct()
{
 static int k=9;

 k++;

}
void gct()
{
 int k=8;

 k=k+5; // (?)
 ::k++;

}

```

În instrucțiunea marcată cu (?), variabila k ia valoarea:

- a) 33;
- b) 37;
- c) 13;

- d) 14;
- e) altă valoare;

20. În secvența de program:

```

.....
int k=90;
void fct()
{
 int k=5;

 ::k++;
 k=k+7; // (?)

}
void gct()
{
 int k=7;

 k--;

}

```

În instrucțiunea marcată cu (?), k este o variabilă:

- a) locală;
- b) pointer;
- c) registru;
- d) globală;
- e) statică;

21. Se consideră secvența de program:

```

class B1 {
public:
 B1(){cout<<"(1)\n";}
 ~B1(){cout<<"(2)\n";}
};
class B2 {
public:
 B2(){cout<<"(3)\n";}
 ~B2(){cout<<"(4)\n";}
};
class B3 {
public:
 B3(){cout<<"(5)\n";}
 ~B3(){cout<<"(6)\n";}
};
void main(){
D1 ob1;
D2 ob2;
}

class D1:public B1,public B2 {
public:
 D1(){cout<<"(7)\n";}
 ~D1(){cout<<"(8)\n";}
};
class D2:public D1,public B3 {
public:
 D2(){cout<<"(9)\n";}
 ~D2(){cout<<"(10)\n";}
};

```

Care mesaj se va scrie?

- a) (1),(3),(7),(3),(5),(7),(9),(10),(6),(8),(4),(2),(2),(3),(2),(2);
- b) (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(9),(7),(2),(3),(2),(2);

- c) (1),(3),(7),(1),(3),(7),(5),(9),(10),(6),(8),(4),(2),(8),(4),(2);  
 d) (1),(3),(5),(7),(9),(2),(4),(6),(6),(8),(8),(10),(2),(2),(4),(2);  
 e) (1),(3),(7),(1),(3),(7),(9),(5),(10),(6),(4),(8),(2),(8),(4),(2);

22. Se consideră secvența de program:

```
class B1 {
public:
 B1(){cout<<"(1)\n";}
 ~B1(){cout<<"(2)\n";}
};
class B2 {
public:
 B2(){cout<<"(3)\n";}
 ~B2(){cout<<"(4)\n";}
};
class B3 {
public:
 B3(){cout<<"(5)\n";}
 ~B3(){cout<<"(6)\n";}
};
void main(){
 D1 ob1;
 D2 ob2;
}
```

```
class D1:public B1,public B2 {
public:
 D1(){cout<<"(7)\n";}
 ~D1(){cout<<"(8)\n";}
};
class D2:public D1,public B3 {
public:
 D2(){cout<<"(9)\n";}
 ~D2(){cout<<"(10)\n";}
};
```

Care mesaj se va scrie?

- a) (1),(3),(7),(1),(3),(7),(9),(5),(10),(6),(4),(8),(8),(2),(4),(2);  
 b) (1),(3),(7),(1),(3),(7),(5),(9),(10),(6),(8),(4),(2),(8),(4),(2);  
 c) (1),(3),(7),(3),(5),(7),(9),(10),(6),(8),(4),(4),(2),(2),(3),(1);  
 d) (1),(8),(4),(4),(3),(3),(7),(3),(5),(7),(9),(10),(2),(2),(2),(4);  
 e) (1),(3),(1),(7),(7),(3),(9),(5),(6),(10),(8),(6),(2),(4),(4),(2);

23. Se consideră secvența de program:

```
class B1 {
public:
 B1(){cout<<"(1)\n";}
 ~B1(){cout<<"(2)\n";}
};
class B2 {
public:
 B2(){cout<<"(3)\n";}
 ~B2(){cout<<"(4)\n";}
};
class B3 {
public:
 B3(){cout<<"(5)\n";}
 ~B3(){cout<<"(6)\n";}
};
void main(){
 D1 ob1;
 D2 ob2;
}
```

```
class D1:public B1,public B2 {
public:
 D1(){cout<<"(7)\n";}
 ~D1(){cout<<"(8)\n";}
};
class D2:public D1,public B3 {
public:
 D2(){cout<<"(9)\n";}
 ~D2(){cout<<"(10)\n";}
};
```

Care mesaj se va scrie?

- a) (1),(3),(1),(7),(7),(3),(9),(5),(6),(10),(8),(6),(2),(4),(4),(2);
- b) (1),(4),(8),(3),(4),(7),(3),(5),(3),(9),(7),(2),(10),(4),(2),(2);
- c) (1),(3),(7),(3),(5),(7),(9),(10),(6),(8),(4),(4),(2),(2),(3),(1);
- d) (1),(7),(3),(3),(1),(9),(7),(5),(6),(10),(8),(4),(2),(8),(4),(2);
- e) (1),(3),(7),(1),(3),(7),(5),(9),(10),(6),(8),(4),(2),(8),(4),(2);

24. Se consideră secvența de program:

```

class B1 {
 public:
 B1(){cout<<"(1)\n";}
 ~B1(){cout<<"(2)\n";}
};
class B2 {
 public:
 B2(){cout<<"(3)\n";}
 ~B2(){cout<<"(4)\n";}
};
class B3 {
 public:
 B3(){cout<<"(5)\n";}
 ~B3(){cout<<"(6)\n";}
};
void main(){
 D1 ob1;
 D2 ob2;
}
class D1:public B1,public B2 {
 public:
 D1(){cout<<"(7)\n";}
 ~D1(){cout<<"(8)\n";}
};
class D2:public D1,public B3 {
 public:
 D2(){cout<<"(9)\n";}
 ~D2(){cout<<"(10)\n";}
};

```

Care mesaj se va scrie?

- a) (1),(2),(3),(4),(5),(6),(7),(8),(9),(10),(9),(7),(4),(2),(3),(2);
- b) (1),(3),(2),(5),(4),(7),(6),(8),(9),(10),(9),(4),(7),(3),(2),(2);
- c) (1),(3),(7),(1),(3),(7),(5),(9),(10),(6),(8),(4),(2),(8),(4),(2);
- d) (1),(3),(7),(5),(3),(9),(7),(6),(10),(4),(8),(4),(2),(3),(2),(2);
- e) (1),(2),(3),(4),(5),(4),(6),(5),(7),(8),(5),(9),(10),(7),(8),(2);

25. Se consideră secvența de program:

```

class B1 {
 public:
 B1(){cout<<"(1)\n";}
 ~B1(){cout<<"(2)\n";}
};
class B2 {
 public:
 B2(){cout<<"(3)\n";}
 ~B2(){cout<<"(4)\n";}
};
class B3 {
 public:
 B3(){cout<<"(5)\n";}
 ~B3(){cout<<"(6)\n";}
};
void main(){
 D1 ob1;
 D2 ob2;
}
class D1:public B1,public B2 {
 public:
 D1(){cout<<"(7)\n";}
 ~D1(){cout<<"(8)\n";}
};
class D2:public D1,public B3 {
 public:
 D2(){cout<<"(9)\n";}
 ~D2(){cout<<"(10)\n";}
};

```

Care mesaj se va scrie?

- a) (1),(3),(7),(1),(3),(7),(5),(9),(10),(6),(8),(4),(2),(8),(4),(2);
- b) (1),(7),(3),(3),(7),(1),(10),(9),(5),(4),(8),(6),(4),(8),(2),(2);
- c) (1),(3),(3),(7),(9),(10),(1),(7),(6),(8),(4),(5),(2),(8),(4),(2);
- d) (1),(3),(5),(7),(3),(1),(4),(6),(10),(9),(2),(4),(8),(2),(8),(2);
- e) (1),(3),(7),(5),(7),(6),(8),(7),(9),(10),(8),(9),(4),(6),(5d),(2);



| Test | Raspuns |
|------|---------|
| 1    | d       |
| 2    | b       |
| 3    | b       |
| 4    | d       |
| 5    | a       |
| 6    | c       |
| 7    | b       |
| 8    | d       |
| 9    | a       |
| 10   | c       |
| 11   | a       |
| 12   | a       |
| 13   | a       |
| 14   | e       |
| 15   | b       |
| 16   | d       |
| 17   | a       |
| 18   | a       |
| 19   | c       |
| 20   | a       |
| 21   | c       |
| 22   | b       |
| 23   | e       |
| 24   | c       |
| 25   | a       |

| Test | Raspuns |
|------|---------|
|------|---------|

## 5 Arhitectura sistemelor de calcul

1. Converteți în zecimal numerele următoare (baza este indicată în indice):  $DA.C_{16}$ , 27,  $4_8$ , 11011,  $101_2$ .
  - a) 218.75 23.5 27.625
  - b) 2187.5 17.5 27.625
  - c) 218.75 22.5 26.575
  - d) 21.875 22.5 55.125
  - e) 4367.5 47.0 55.125
2. Dați valoarea zecimală cu semn a numărului  $B7_{16}$  codificat în complement față de 2 (cod complementar) pe 8 biți.
  - a) +7,3
  - b) -73
  - c) -71
  - d) -7,3
  - e) +73
3. Dați valoarea zecimală cu semn a numărului  $B7_{16}$  codificat în complement față de 1 (cod invers) pe 8 biți.
  - a) +73
  - b) +72
  - c) -72
  - d) -73
  - e) -74
4. Dați pe 8 biți (exprimată în hexazecimal) reprezentarea în cod complementar (complement față de 2) a valorii întregi -32.
  - a) B0
  - b) D0
  - c) A0
  - d) E0
  - e) C0
5. Dați pe 8 biți (exprimată în hexazecimal) reprezentarea în complement față de 1 (cod invers) a valorii întregi -32.
  - a) BF
  - b) CF
  - c) AF
  - d) EF
  - e) DF
6. Dați sub forma  $\pm a \times 2^b$  (a și b reprezentate în sistemul zecimal) valoarea care corespunde reprezentării interne în virgulă mobilă simplă precizie, (1 bit = semnul mantisei, 8 biți=exponentul decalat, 23 biți=mantisa normalizată), a numărului 27632000000 exprimat în octal.
  - a)  $-0.8125 \times 2^{-4}$
  - b)  $-8.8125 \times 2^7$
  - c)  $-0.8185 \times 2^{-7}$
  - d)  $-8.8125 \times 2^4$
  - e)  $0.8185 \times 2^{-4}$

7. Dați sub forma  $\pm a \times 2^b$  (a și b reprezentate în sistemul zecimal) valoarea care corespunde reprezentării interne în virgulă mobilă simplă precizie, (1 bit = semnul mantisei, 8 biți=exponentul decalat, 23 biți=mantisa normalizată), a numărului 27660000000 exprimat în octal.
- $-0.1011 \times 2^{-4}$
  - $-0.5 \times 2^{-3}$
  - $0.101 \times 2^3$
  - $0.A1 \times 2^{-5}$
  - $0.B1 \times 2^{-6}$
8. Dați sub forma  $\pm a \times 2^b$  (a și b reprezentate în sistemul zecimal) valoarea care corespunde reprezentării interne în virgulă mobilă simplă precizie, (1 bit = semnul mantisei, 8 biți=exponentul decalat, 23 biți=mantisa normalizată), a numărului 37564000000 exprimat în octal.
- $-8.8125 \times 2^{200}$
  - $0.8185 \times 2^{-4}$
  - $-0.625 \times 2^{123}$
  - $0.101 \times 2^{-4}$
  - $0.111 \times 2^{123}$
9. Dați reprezentarea internă, exprimată în hexazecimal, în virgulă mobilă simplă precizie, (1 bit = semnul mantisei, 8 biți = exponentul decalat, 23 biți = mantisa normalizată), pentru valoarea zecimală 278.
- 44B59000
  - 34C58000
  - 45DC3400
  - 44C58000
  - 54CD9000
10. Dați reprezentarea internă, exprimată în hexazecimal, în virgulă mobilă simplă precizie, (1 bit = semnul mantisei, 8 biți = exponentul decalat, 23 biți = mantisa normalizată), pentru valoarea zecimală -6,25.
- C4B59000
  - 44C58000
  - C4C58000
  - CDF0A00
  - C1E40000
11. Fie o transmisie care utilizează codul lui Hamming cu o paritate impară. Găsiți mesajul transmis (reprezentare octală) știind că reprezentarea octală, a datelor de transmis (pe 16 biți) este 116570, în octal.
- 4653510
  - 4654510
  - 4655510
  - 4664510
  - 4653500
12. Fie o transmisie care utilizează codul lui Hamming cu o paritate impară. Regăsiți (în octal) mesajul inițial (corectând eventualele erori) dacă mesajul primit (pe 21 biți) este 6130014, în octal.
- 140441
  - 141441
  - 142442
  - 142542

- e) 142452
13. Fie o transmisie care utilizează metoda codurilor polinomiale (CRC) prin intermediul polinomului generator  $G(x) = x^5 + x + 1$ . Dacă se dorește transmiterea datelor (9 biți), având reprezentarea octală 456, care va fi mesajul trimis (în octal) ?
- 22563
  - 23734
  - 22734
  - 22735
  - 22834
14. Fie o transmisie care utilizează metoda codurilor polinomiale (CRC) prin intermediul polinomului generator  $G(x) = x^2 + x + 1$ . Mesajul recepționat este  $T(x) = x^{10} + x^7 + x^6 + x^4 + x^3 + x$ . S-au produs erori de transmisie ? Dacă nu, dați sub forma polinomului  $M(x)$  datele inițiale transmise.
- $M(x) = x^7 + x^5 + x^3 + 1$
  - $M(x) = x^7 + x^5 + 1$ ;
  - erori de transmisie;
  - $M(x) = x^8 + x^5 + x^4 + x^2 + x$ ;
  - $M(x) = x^8 + x^5 + 1$
15. Fie o transmisie care utilizează codul lui Hamming cu o paritate pară. Regăsiți (în octal) mesajul inițial (corectând eventualele erori) dacă mesajul primit (pe 7 biți) este 134, exprimat în octal.
- 141
  - 142
  - 21
  - 77
  - 11
16. Reduceți expresia formei normale disjunctive (cu mintermeni) a funcției logice  $f(p, q, r) = NAND(p, q, r)$ .
- $\bar{p} + \bar{q} + \bar{r}$
  - $pqr + \bar{q}$
  - $p + \bar{p}\bar{q}$
  - $\bar{p}\bar{q}\bar{r}$
  - $\bar{p}(\bar{q} + \bar{r})$
17. Reduceți expresia formei normale disjunctive (cu mintermeni) a funcției logice  $f(p, q, r) = \bar{p}\bar{q}r + p\bar{q}\bar{r} + p\bar{q}r + pq\bar{r} + pqr$ .
- $pqr + \bar{q}$
  - $p + \bar{q}r$
  - $\bar{p}\bar{q}r + p\bar{q} + pq$
  - $pqr + \bar{r}$
  - $pq\bar{r} + \bar{q}$
18. Reduceți expresia formei normale disjunctive (cu mintermenii) a funcției logice  $f(p, q) = pq + p\bar{q} + \bar{p}q$
- $pqr + \bar{q}$
  - $\bar{p} + \bar{q} + \bar{r}$
  - $p + q$
  - $p + \bar{p}\bar{q}$
  - $pq$

19. Reduceți expresia formei normale disjunctive (cu mintermeni) a funcției logice  $f(p, q, r) = \bar{p}\bar{q}\bar{r} + p\bar{q}\bar{r} + p\bar{q}r + pqr$
- $\bar{p}qr + p\bar{q} + pq$
  - $p + \bar{p}\bar{q}$
  - $pq + \bar{p}\bar{q}$
  - $pr + \bar{q}\bar{r}$
  - $pqr + \bar{p}\bar{q}$
20. Reduceți expresia formei normale disjunctive (cu mintermeni) a funcției logice de 4 variabile  $(p, q, r, s)$  care ia valoarea 1 dacă  $pqrs$  văzut ca număr binar are o valoare mai mică decât 10.
- $\bar{q}\bar{r} + \bar{p}$
  - $\bar{p}qr + p\bar{q} + pq$
  - $p + \bar{p}\bar{q}$
  - $pr + \bar{p}\bar{q}$
  - $pq + pr$
21. Presupunem construirea unei memorii de 64Mo ( $2^{26}$  octeți) din blocuri de câte 4Mo. Câți biți ar fi necesari pentru a specifica o adresă de memorie?
- 26
  - 22
  - 20
  - 2
  - 24
22. Se consideră o memorie centrală de 2Mo unde fiecare octet este adresat separat. Calculați adresa în hexazecimal al celui de-al șaselea element al unui tablou cu adresa primului element  $1000_{16}$ , și care are rezervat pentru reprezentarea fiecărui element 16 biți.
- 100B
  - 100A
  - 77B
  - 1005
  - 1006
23. Se consideră o memorie centrală de 2Mo unde fiecare octet este adresat separat. Calculați capacitatea acestei memorii, ca număr de cuvinte de 16 biți respectiv de cuvinte de 32 biți.
- $2^{11}$  cuvinte de 16 biți,  $2^9$  cuvinte de 32 biți;
  - $2^{10}$  cuvinte de 16 biți,  $2^{11}$  cuvinte de 32 biți;
  - $2^{20}$  cuvinte de 16 biți,  $2^{19}$  cuvinte de 32 biți;
  - $2^9$  cuvinte de 16 biți,  $2^{10}$  cuvinte de 32 biți;
  - $2^{11}$  cuvinte de 16 biți,  $2^9$  cuvinte de 32 biți;
24. Dacă registrul de adresă RA dispune de 32 de biți, calculați numărul de cuvinte adresabile dacă un cuvânt are 32 biți și cea mai mare adresă posibilă pentru aceste cuvinte de 32 biți.
- $2^{16} \quad 2^{32} - 1$ ;
  - $2^{32} \quad 2^{16} - 1$ ;
  - $2^{31} \quad 2^{31}$ ;
  - $2^{32} \quad 2^{32} - 1$ ;
  - $2^{32} \quad 2^{32}$ ;

25. Se consideră o memorie întretesută constituită din două unități de câte 4 Mo. Știind că adresa de memorie se constituie pe 23 de biți astfel: 1 bit identifică superunitatea, 20 biți specifică adresa în cadrul unității, iar ultimii 2 biți octetul din cadrul cuvântului), să se identifice (în zecimal) linia într-o memorie cache de 1Ko precum și biții de pondere mare ai unei adrese de memorie  $A=10000$ .

- a) 600 42
- b) 600 2
- c) 452 42
- d) 600 24
- e) 452 2

26. Care este efectul următoarei secvențe de instrucțiuni care se execută pe o mașină cu programare pe o adresă ?

- 1) LOAD F
- 2) DIV G
- 3) STA T1
- 4) LOAD D
- 5) MPY E
- 6) ADD C
- 7) SUB T1
- 8) MPY B
- 9) STA A

- a)  $A = B \times (C + D \times E - F/G)$
- b)  $R = A - B/(C + D) + E \times F - G$
- c)  $A = B \times (C + D \times E) - F/G$
- d)  $R = B/(C + D) + E \times F - G$
- e)  $A = B \times (C \times D + E) - F/G$

27. Care este efectul următoarei secvențe de instrucțiuni care se execută pe o mașină cu programare pe zero adrese (LIFO)?

- 1) LOAD B
- 2) LOAD C
- 3) LOAD D
- 4) LOAD E
- 5) MPY
- 6) ADD
- 7) LOAD F
- 8) LOAD G
- 9) DIV
- 10) SUB
- 11) MPY
- 12) STA A

- a)  $R = B \times (C + D) \times E - F/G$
- b)  $A = B \times (C + D \times E - F/G)$
- c)  $A = B \times (F/G - C + D \times E)$
- d)  $R = B \times (C + D \times E) - F/G$
- e)  $A = B \times F/G - (C + D \times E)$

28. Se consideră următorul conținut al registrelor și al memoriei:

(XR1) = 1  
(XR2) = 2  
(B1) = 1000  
(B2) = 2000  
(0) = 1000  
(1000) = 0  
(1001) = 1  
(2000) = 2  
(2001) = 3  
(3000) = 4  
(3001) = 5

Care este valoarea lui F după execuția programului următor:

1) LOAD 3000  
2) ADD 2000, XR1  
3) SUB 2001, B1  
4) MPY 1001, B2  
5) DIV 2000  
6) ADD 1000, IM  
7) SUB 0, I, XR1  
8) STA F

- a) F=1
- b) F=1000
- c) F=1004
- d) F=1001
- e) F=4

29. Presupunem că una dintre unitățile din cadrul UAL, ale unui sistem de calcul este organizată în pipeline. Operațiile efectuate pe această unitate sunt descompuse în 5 acțiuni elementare. Dacă fiecare acțiune este executată în 8ns (nanosecunde), calculați timpul necesar pentru repetarea aceleiași operații asupra unui vector de 20 de componente și timpul necesar pentru prelucrarea vectorului.

- a) 50ns 192ns
- b) 40ns 200ns
- c) 50ns 200ns
- d) 40ns 192ns
- e) 8ns 200ns

30. Presupunem că una dintre unitățile din cadrul UAL ale unui sistem de calcul este organizată în pipeline. Operațiile efectuate pe această unitate sunt descompuse în 5 acțiuni elementare. Dacă acțiunile sunt executate respectiv în 15, 12, 6, 8 și 10ns (nanosecunde), calculați timpul necesar pentru repetarea aceleiași operații asupra unui vector de 32 de componente și timpul necesar pentru prelucrarea vectorului.

- a) 40ns 500 ns
- b) 40ns 200ns
- c) 51ns 500 ns
- d) 40ns 516ns
- e) 51ns 516 ns

31. Presupunem că o adresă în memoria virtuală paginată, necesită 20 de biți organizați astfel:

| pagina   | offset |
|----------|--------|
| 19....12 | 11...0 |

Care este capacitatea acestei memorii virtuale, exprimată în număr de cuvinte respectiv în număr de pagini ?

- a)  $2^{20}$  cuvinte,  $2^8$  pagini;
- b)  $2^{20}$  cuvinte,  $2^{10}$  pagini;
- c)  $2^{10}$  cuvinte,  $2^{10}$  pagini;
- d)  $2^{12}$  cuvinte,  $2^{12}$  pagini;
- e)  $2^{12}$  cuvinte,  $2^8$  pagini;

32. Presupunem că o adresă în memoria virtuală paginată, necesită 20 de biți organizați astfel:

|         |        |
|---------|--------|
| pagina  | offset |
| 19...12 | 11...0 |

Care este adresa octală a celui de-al 970-lea cuvânt de la pagina 213 ?

- a) 3251717;
- b) 3251711;
- c) 3261713;
- d) 3261711;
- e) 3261715;

33. Presupunem că o adresă în memoria virtuală paginată, necesită 20 de biți organizați astfel:

|         |        |
|---------|--------|
| pagina  | offset |
| 19...12 | 11...0 |

Care este numărul cuvântului și numărul de pagină corespunzătoare adresei hexazecimale ABCDE ? Dați rezultatul în zecimal.

- a) Număr cuvânt=4000, Număr pagină=171;
- b) Număr cuvânt=3295, Număr pagină=170;
- c) Număr cuvânt=3294, Număr pagină=171;
- d) Număr cuvânt=4000, Număr pagină=170;
- e) Număr cuvânt=3295, Număr pagină=171;

34. Presupunem că o adresă în memoria virtuală segmentată și paginată, necesită 24 de biți organizați astfel:

|         |        |        |
|---------|--------|--------|
| segment | pagina | offset |
| 23...18 | 17...8 | 7...0  |

Care este capacitatea acestei memorii virtuale, exprimată în număr de segmente, în număr de pagini, respectiv în număr de cuvinte?

- a)  $2^{10}$  segmente,  $2^8$  pagini,  $2^{20}$  cuvinte;
- b)  $2^{10}$  segmente,  $2^{10}$  pagini,  $2^{10}$  cuvinte;
- c)  $2^6$  segmente,  $2^{16}$  pagini,  $2^{10}$  cuvinte;
- d)  $2^6$  segmente,  $2^{16}$  pagini,  $2^{24}$  cuvinte;
- e)  $2^6$  segmente,  $2^{10}$  pagini,  $2^{24}$  cuvinte;

35. Presupunem că o adresă în memoria virtuală segmentată și paginată, necesită 24 de biți organizați astfel:

|         |        |        |
|---------|--------|--------|
| segment | pagina | offset |
| 23...18 | 17...8 | 7...0  |

Care este adresa octală a celui de-al 145-lea cuvânt de la pagina 111 a segmentului al 32-lea ?

- a) 32517180;
- b) 30706720;
- c) 32517160;
- d) 30706740;
- e) 37067620;



| Test | Raspuns |
|------|---------|
| 1    | a       |
| 2    | b       |
| 3    | c       |
| 4    | d       |
| 5    | e       |
| 6    | a       |
| 7    | b       |
| 8    | c       |
| 9    | d       |
| 10   | e       |
| 11   | a       |
| 12   | b       |
| 13   | c       |
| 14   | d       |
| 15   | e       |
| 16   | a       |
| 17   | b       |
| 18   | c       |
| 19   | d       |
| 20   | a       |
| 21   | a       |
| 22   | b       |
| 23   | c       |
| 24   | d       |
| 25   | e       |
| 26   | a       |
| 27   | b       |
| 28   | c       |
| 29   | d       |
| 30   | e       |
| 31   | a       |
| 32   | b       |
| 33   | c       |
| 34   | d       |
| 35   | e       |

| Test | Raspuns |
|------|---------|
|------|---------|

## 6 Sisteme de operare și rețele de calculatoare

1. Atributul PURGE atașat unui fișier Novell nu permite:
  - a) copierea fișierului dintr-un director în altul;
  - b) modificarea datelor din fișier;
  - c) ștergerea fișierului;
  - d) deschiderea simultană de către mai mulți utilizatori;
  - e) recuperarea fișierului după ce acesta a fost șters;
2. În ce caz un fișier Novell copiat pe un floppy pastrează atributele specifice S.O. Novell.
  - a) dacă discul este formatat după încărcarea S.O. Novell;
  - b) dacă discul se formatează în prealabil cu ajutorul comenzii FORMAT A: /U /S /Q;
  - c) dacă numele fișierului respectă convenția de nume și extensie DOS;
  - d) dacă discul se va citi pe un calculator ce face parte tot dintr-o rețea Novell;
  - e) în nici unul din cazurile de mai sus;
3. Lipsa atributului SHAREABLE atașat unui fișier Novell determină:
  - a) imposibilitatea modificării datelor din fișier;
  - b) accesul mai lent la datele conținute în fișier;
  - c) imposibilitatea recuperării fișierului după ce acesta a fost șters;
  - d) setarea automată a atributului READWRITE;
  - e) eșecul tentativei de deschidere simultană de către mai mulți utilizatori;
4. Atributul INDEXED atașat unui fișier Novell:
  - a) crează un fișier index pentru datele fișierului;
  - b) face accesul mai lent la datele conținute în fișier;
  - c) împiedică plasarea acestuia pe mai multe volume;
  - d) determină marcarea sectoarelor defecte din zona unde este plasat fișierul;
  - e) nu are nici unul din efectele de mai sus;
5. Comanda FLAG test.txt N determină setarea următoarelor atribute ale fișierului Novell test.txt :
  - a) SYSTEM;
  - b) ReadOnly+ARCHIVE;
  - c) ARCHIVE;
  - d) ReadWrite+ARCHIVE;
  - e) ReadWrite;
6. Sub DOS, atributul SYSTEM poate fi atașat:
  - a) numai fișierelor ce se crează automat la formatare;
  - b) numai fișierelor Io.sys și MsDos.sys;
  - c) numai fișierelor de lungime nenulă;
  - d) numai fișierelor ascunse;
  - e) oricărui fișier;
7. Sub DOS, atributul SYSTEM atașat unui fișier:
  - a) nu permite redenumirea fișierului;
  - b) determină atașarea automată a atributului HIDDEN;
  - c) nu permite modificarea atributelor fișierului;

- d) nu permite copierea fișierului;
  - e) nu permite modificarea datelor din fișier;
8. Sub DOS, atributul ARCHIVE se atașază automat unui fișier atunci când:
- a) utilizatorul lansează comanda TYPE nume\_fișier.extensie;
  - b) comprimă datele din fișier cu ajutorul unui program specializat;
  - c) se copiază fișierul cu ajutorul comenzii COPY cu parametrul /M;
  - d) fișierul este deschis pentru citirea sau scrierea datelor;
  - e) fișierul este modificat;
9. Sub DOS, atributul ReadOnly împiedică:
- a) copierea fișierului;
  - b) deschiderea fișierului;
  - c) modificarea atributelor fișierului;
  - d) redenumirea fișierului;
  - e) ștergerea fișierului;
10. Sub DOS, octetul de atribute al unui fișier:
- a) poate lipsi, atunci când fișierul are lungimea 0;
  - b) are valori cuprinse între 1 și 128;
  - c) este plasat automat la sfârșitul fișierului;
  - d) este memorat pe 10 biți;
  - e) nu poate avea valoarea 24;
11. Sub DOS, octetul de atribute al unui fișier:
- a) poate lipsi, atunci când fișierul are lungimea 0;
  - b) are valori cuprinse între 1 și 128;
  - c) este plasat automat la sfârșitul fișierului;
  - d) este memorat pe 10 biți;
  - e) se memorează în zona directorului în care este plasat fișierul;
12. Sub DOS, în octetul de atribute al unui fișier:
- a) valoarea 1 a primului bit determină anularea valorii bitului 7;
  - b) biții 1 și 2 nu pot fi simultan 1;
  - c) biții 1 și 2 nu pot fi simultan 1;
  - d) biții 8 și 9 sunt 1;
  - e) ultimii doi biți sunt nefolosiți;
13. Sub DOS, data creării unui fișier:
- a) se memorează pe 6 octeți sub forma ZZ/LL/AA;
  - b) se memorează pe 6 octeți sub forma LL/ZZ/AA;
  - c) se memorează pe 8 octeți sub forma ZZ/LL/AAAA;
  - d) se memorează pe 8 octeți sub forma LL/ZZ/AAAA;
  - e) se memorează pe 16 biți;
14. Sub DOS, ora creării unui fișier:
- a) se memorează pe 32 de biți forma OO:MM:SS;
  - b) se memorează pe 32 de biți forma Ora\*3600+Min\*60+Sec;

- c) se memorează pe 24 de biți forma  $\text{Ora} * 3600 + \text{Min} * 60 + \text{Sec}$ ;
  - d) se memorează la sfârșitul fișierului;
  - e) se memorează pe 16 biți;
15. Sub DOS, spațiul în octeți ocupat efectiv de datele unui fișier:
- a) este egal cu cel raportat de comanda `DIR nume_fișier.extensie`;
  - b) depinde de numărul de atributele atașate fișierului;
  - c) este mai mic decât cel raportat de comanda `DIR nume_fișier.extensie`;
  - d) se memorează la sfârșitul fișierului;
  - e) este un număr par;
16. Sub DOS, datele unui fișier:
- a) sunt plasate totdeauna imediat după zona în care sememorează numele fișierului;
  - b) sunt șterse simultan cu ștergerea fișierului;
  - c) sunt șterse imediat după ștergerea numelui fișierului;
  - d) sunt deplasate la execuția comenzii `MOVE`;
  - e) se găsesc în afara zonei sistem a discului;
17. În zona `BOOT` a unui disc `DOS` se memorează:
- a) numele etichetei de volum a discului;
  - b) tabela de alocare a partițiilor;
  - c) fișierele sistemului de operare;
  - d) `COMMAND.COM`;
  - e) versiunea sistemului de operare;
18. Sub DOS, tabela de alocare a partițiilor:
- a) conține, pe 16 biți, numărul de partiții;
  - b) conține programul ce lansează în execuție interpretorul de comenzi;
  - c) ocupă 1024 octeți;
  - d) este creată automat la formatarea discului;
  - e) se termină cu `55AA`;
19. Sub DOS, tabela de alocare a partițiilor:
- a) conține, pe 16 biți, numărul de partiții;
  - b) conține zona `FAT`;
  - c) se află imediat după tabela `FAT`;
  - d) este modificată la ștergerea unui fișier;
  - e) este plasată în primul sector fizic al discului;
20. Sub DOS, zona `FAT`:
- a) conține, pe 16 biți, numărul de partiții;
  - b) conține directorul rădăcină;
  - c) se află la sfârșitul partiției;
  - d) memorează partiția activă;
  - e) începe în al doilea sector al unui disc floppy;
21. Sub DOS, zona `FAT`:
- a) conține, memorat în complement față de 2, numărul de partiții;

- b) conține dimensiunea directorului rădăcină;
  - c) se află la începutul partiției active;
  - d) memorează partiția activă;
  - e) este modificată la ștergerea unui fișier;
22. Sub DOS, zona FAT:
- a) ocupă un sector;
  - b) se află numai în spațiul partiției active;
  - c) conține un program ce încarcă sistemul de operare;
  - d) se modifică de fiecare dată când se redenumeste un fișier;
  - e) este creată automat la formatarea discului;
23. Directorul rădăcină al unui disc DOS:
- a) ocupă primul sector al discului;
  - b) conține un program ce încarcă sistemul de operare;
  - c) se modifică de fiecare dată când se redenumeste un fișier;
  - d) ocupă un spațiu variabil, în funcție de numărul de fișiere și directoare memorate în această zonă;
  - e) este creat automat la formatarea discului;
24. Directorul rădăcină al unui disc DOS:
- a) conține întotdeauna fișierul Autoexec.bat;
  - b) conține întotdeauna fișierele Io.sys și MsDos.sys;
  - c) poate ocupa cel mult dublul spațiului alocat zonei BOOT;
  - d) memorează sectoarele defecte ale discului;
  - e) nu poate fi șters;
25. Directorul rădăcină al unui disc DOS:
- a) conține întotdeauna fișierele Io.sys și MsDos.sys;
  - b) este plasat după zona BOOT;
  - c) se modifică la ștergerea oricărui fișier de pe disc;
  - d) poate fi redenumit;
  - e) nu are director părinte;
26. Comanda DOS Dir \*A.??? determină:
- a) afișarea tuturor fișierelor și directoarelor vizibile din directorul curent al căror nume se termină cu litera A, indiferent de extensie;
  - b) afișarea tuturor fișierelor și directoarelor vizibile din directorul curent al căror nume se termină cu litera A, și care au extensia formată din 3 caractere identice;
  - c) afișarea tuturor fișierelor și directoarelor vizibile din directorul curent al căror nume se termină cu litera A și extensia formată din 3 caractere;
  - d) afișarea tuturor fișierelor și directoarelor vizibile din directorul curent al căror nume se termină cu litera A și nu au extensie;
  - e) afișarea tuturor numelor fișierelor și directoarelor vizibile din directorul curent;
27. Comanda DOS TYPE command.com :
- a) permite lansarea în execuție a interpretorului de comenzi command.com;
  - b) dacă este urmată de parametrul /M permite editarea fișierului command.com;
  - c) permite vizualizarea numelui comenzilor interne conținute de interpretorul de comenzi;

- d) determină apariția unui mesaj de eroare;
  - e) este echivalentă cu Copy command.com CON;
28. Cu ajutorul comenzii DOS MODE nu se poate realiza:
- a) modificarea numărului de caractere afișate pe ecran în mod text;
  - b) deplasarea spre stânga sau spre dreapta a tuturor caracterelor aflate pe ecran;
  - c) setarea modului de operare pentru portul serial;
  - d) încărcarea unei copii a interpretorului de comenzi command.com;
  - e) redirectarea ieșirii paralele spre cea serială;
29. Comanda DOS PATH permite:
- a) modificarea directorului curent de pe discul implicit;
  - b) introducerea unei liste de directoare în care să fie căutate fișierele de date ce nu se găsesc în directorul curent;
  - c) ștergerea fișierelor executabile ce nu se află în directorul curent;
  - d) încărcarea unei copii a interpretorului de comenzi command.com;
  - e) introducerea unei liste de directoare în care să fie căutate fișierele executabile ce nu se găsesc în directorul curent;
30. Comanda DOS COPY A:\*. \* + ,,A: realizează:
- a) actualizarea datei și orei pentru toate fișierele din directorul curent de pe discul A;
  - b) copiază toate fișierele din directorul rădăcină al discului A: în directorul curent;
  - c) copiază toate fișierele din directorul curent al discului A: în directorul curent de pe discul implicit;
  - d) apariția unui mesaj de eroare datorită prezenței caracterelor nepermise + și ,;
  - e) concatenarea tuturor fișierelor din directorul curent de pe unitatea A: într-unul singur, ce va înlocui primul fișier întâlnit;
31. Comanda DOS ERASE:
- a) permite ștergerea fișierelor și directoarelor;
  - b) permite ștergerea directoarelor dacă este urmată de parametrul /S;
  - c) șterge ecranul;
  - d) anulează valorile variabilelor de sistem;
  - e) este echivalentă cu comanda DEL;
32. Comanda DOS COPY \*.txt +\*.doc del.bat realizează:
- a) copiază fișierele cu extensiile txt și doc în directorul curent;
  - b) șterge fișierele cu extensiile txt și doc din directorul curent;
  - c) apariția unui mesaj de eroare datorită prezenței caracterului nepermis +;
  - d) copiază fișierele și lansează del.bat;
  - e) concatenarea tuturor fișierelor cu extensia txt și apoi a celor cu extensia doc în fișierul del.bat;
33. Comanda DOS COPY \*.txt all.txt realizează:
- a) generează mesaj de eroare dacă all.txt nu există;
  - b) modifică data și ora creării fișierelor cu extensia txt din directorul curent;
  - c) copiază toate fișierele cu extensia txt din directorul curent în directorul all.txt, dacă acesta a fost inclus într-o comandă PATH;
  - d) șterge conținutul fișierului all.txt dacă acesta există în directorul curent și îl înlocuiește cu cel al primului fișier txt întâlnit;

- e) copiază toate fișierele cu extensia txt din directorul curent în directorul all.txt, dacă acesta există în directorul curent;
34. Următoarea comandă DOS generează întotdeauna mesaj de eroare:
- a) COPY CON CON;
  - b) COPY all.txt+ \*.txt;
  - c) COPY CON NUL;
  - d) COPY CON AUX;
  - e) COPY LPT1 CON;
35. Comanda DOS DIR >> abc.txt determină:
- a) apariția unui mesaj de eroare dacă fișierul abc.txt dacă acesta există;
  - b) apariția unui mesaj de eroare dacă fișierul abc.txt dacă acesta nu există;
  - c) ștergerea datelor fișierului abc.txt;
  - d) afișarea conținutului fișierului abc.txt, dacă acesta există;
  - e) mărirea dimensiunii fișierului abc.txt, dacă acesta există;
36. Într-un director, datele despre numele, extensia și celelalte caracteristici ale fișierelor conținute se păstrează întotdeauna:
- a) ordonate alfabetic, după nume;
  - b) ordonate după dimensiune;
  - c) ordonate după data și ora când au fost create/modificate;
  - d) ordonate după tipul fișierului (cele executabile, primele);
  - e) în ordinea în care au fost create;
37. Directorul rădăcină se deosebește întotdeauna de un alt director prin:
- a) numărul de octeți alocat;
  - b) numărul de fișiere conținute;
  - c) faptul intrarea în directorul rădăcină se poate realiza cu comanda CD..;
  - d) numărul de directoare conținute;
  - e) faptul că directorul rădăcină nu are atribute;
38. Adâncimea maximă a unei structuri de directoare este determinată de:
- a) dimensiunea în octeți a discului;
  - b) dimensiunea partiției active;
  - c) numărul de intrări în tabela FAT;
  - d) dimensiunea zonei sistem a discului;
  - e) lungimea liniei de comandă;
39. Partiția activă este acea zonă a discului:
- a) ce conține directorul rădăcină;
  - b) ce conține fișierele sistemului de operare;
  - c) creată cu ajutorul comenzii FORMAT;
  - d) în care se memorează tabela de alocare a partițiilor;
  - e) de pe care se încarcă la pornirea calculatorului sistemul de operare;
40. Execuția comenzii FORMAT A: /Q duce la:
- a) distrugerea iremediabilă a datelor stocate pe discul A.;

- b) plasarea în directorul rădăcină a fișierelor sistemului de operare;
  - c) apariția unui mesaj de eroare dacă discul A: conține date;
  - d) rescrierea tabelii de alocare a partițiilor;
  - e) rescrierea zonei sistem a discului A.;
41. Presupunând că în directorul curent se află fișierele a.exe, a.bat, a.com și se tastează din linia de comandă a <enter> atunci întotdeauna:
- a) se execută succesiv a.bat, a.com, a.exe;
  - b) se execută primul fișier executabil cu numele a, întâlnit în lista de intrări a directorului;
  - c) se execută a.exe;
  - d) se execută a.bat;
  - e) se execută a.com;
42. Comenzile interne DOS sunt:
- a) comenzile a căror execuție nu poate fi întreruptă prin CTRL-C;
  - b) prezente ca fișiere cu extensia com;
  - c) comenzile aflate în directorul DOS;
  - d) comenzile conținute în partea rezidentă a command.com-ului;
  - e) comenzile conținute în partea tranzientă a command.com-ului;
43. Următoarea afirmație este falsă:
- a) un program își poate modifica propriile instrucțiuni în timpul execuției;
  - b) se pot lansa consecutiv mai multe copii ale interpretorului de comenzi command.com;
  - c) numele oricărui fișier poate fi schimbat în command.com;
  - d) în două directoare, al căror nume este prezent ca parametru pentru comanda PATH pot exista două fișiere executabile care au același nume și aceeași extensie;
  - e) în zona memoriei video nu pot fi instalate instrucțiunile unui program;
44. Următoarea afirmație este adevărată:
- a) se poate schimba extensia unui fișier din com în exe și invers, fără ca funcționarea lui să fie afectată;
  - b) după terminarea execuției unui program, fiecare octet din zona de memorie pe care acesta a ocupat-o ia valoarea 0;
  - c) execuția oricărui program poate fi întreruptă cu ajutorul combinației CTRL-C sau CTRL-BREAK;
  - d) command.com-ul nu se poate lansa în execuție pe el însuși;
  - e) în memoria calculatorului pot exista zone de date comune pentru mai multe programe aflate și ele în memorie;
45. Următoarea afirmație este falsă:
- a) fișierele cu extensia com se încarcă întotdeauna la aceeași adresă de memorie;
  - b) în zona rezervată pentru command.com se poate încărca și un alt program;
  - c) interpretorul de comenzi convertește toate caracterele mici de pe linia de comandă în majuscule;
  - d) este permisă folosirea numelui unei comenzi externe DOS ca nume de fișier executabil;
  - e) fișierele executabile cu extensia exe au totdeauna dimensiunea în octeți mai mare decât cele cu extensia com;
46. Supervisorul nu poate:
- a) să atribuie drepturi în directorul rădăcină al volumului SYS;
  - b) să interzică accesul unui utilizator în directorul PUBLIC;



- c) să permită accesul unui utilizator în directorul SYSTEM;
- d) să stabilească drepturile managerilor de grup;
- e) să interzică accesul utilizatorilor în directorul LOGIN;

47. Supervisorul poate:

- a) să stabilească contravaloarea serviciilor puse la dispoziție prin rețea, pentru fiecare utilizator sau grup în parte;
- b) să interzică accesul unui utilizator la resursele stației de lucru pe care lucrează acesta;
- c) să șteargă de la distanță fișierele de pe discurile locale ale stațiilor de lucru;
- d) să cunoască parolele tuturor utilizatorilor;
- e) să modifice masca de drepturi moștenite;

48. Operatorul la consola File-Serverului este utilizatorul care:

- a) este echivalent în securitate cu supervisorul;
- b) poate accesa fișierele din directorul SYSTEM;
- c) poate modifica parolele utilizatorilor rețelei;
- d) poate porni serverul unei rețele Novell;
- e) deține o parolă cu care are acces la tastatura serverului rețelei;

49. Un utilizator își poate retrage drepturile acordate pe un director dacă:

- a) este supervisor;
- b) directorul nu are atributul ReadOnly;
- c) este singurul utilizator cu acces în acel director;
- d) are credit nelimitat;
- e) are dreptul de control al accesului;

50. Următorul serviciu nu poate fi contabilizat și taxat:

- a) citirea fișierelor de pe un disc al rețelei;
- b) scrierea fișierelor pe un disc de rețea;
- c) păstrarea fișierelor pe un disc al rețelei;
- d) timpul cât utilizatorul este cuplat la rețea;
- e) acordarea de drepturi altor utilizator și/sau grupuri;

51. Accesul unui utilizator la resursele mai multor servere dintr-o rețea:

- a) se realizează automat după cuplarea la unul din ele;
- b) poate fi realizat numai cu o nouă comandă LOGIN;
- c) este posibil numai dacă utilizatorul este supervisor;
- d) este posibil numai dacă utilizatorul este manager de grup;
- e) poate fi realizat prin comanda MAP;

52. Accesul unui utilizator la resursele mai multor servere dintr-o rețea:

- a) este posibil numai dacă utilizatorul are același nume de cont pe toate serverele;
- b) este posibil numai dacă utilizatorul are același parolă pentru toate serverele;
- c) este totdeauna permis pentru utilizatorul generic GUEST;
- d) este posibil numai dacă utilizatorul este operator la consola serverului;
- e) nu poate fi realizat cu o nouă comandă LOGIN;

53. Dreptul MODIFY pe un fișier permite utilizatorului schimbarea:

- a) datelor din fișier;
  - b) datei și orei cînd acesta a fost creat;
  - c) lungimii fișierului;
  - d) numelui și a extensiei fișierului;
  - e) atributelor fișierului;
54. Dreptul FileScan pe un fișier permite utilizatorului:
- a) să modifice fișierul;
  - b) să citească datele din fișier;
  - c) să scaneze fișierul în căutarea unor viruși;
  - d) să modifice atributele fișierului;
  - e) să vizualizeze dimensiunea fișierului;
55. Dreptul de control al accesului pe un director permite utilizatorului:
- a) să interzică accesul oricărui alt utilizator în directorul considerat;
  - b) să modifice numele oricărui fișier sau director din directorul dat;
  - c) să șteargă directorul;
  - d) să citească și să modifice orice fișier din director;
  - e) să atribuie și altui utilizator drepturile pe care el le are în director;
56. Dreptul Supervisor pe un fișier:
- a) poate fi acordat numai supervisorului;
  - b) permite recuperarea fișierului după ce acesta a fost șters;
  - c) permite accesul simultan la datele din fișier;
  - d) împiedică copierea lui în rețea;
  - e) permite parcurgerea structurii de directoare de la nivelul directorului rădăcină până la cel în care se află fișierul;
57. Echivalentul în securitate cu supervisorul nu poate:
- a) să retragă drepturile altor echivalenți în securitate cu supervisorul;
  - b) să șteargă fișierele din directorul SYSTEM;
  - c) să modifice atributele fișierelor din directorul PUBLIC;
  - d) să modifice fișierul System Login Script;
  - e) cunoaște parolele utilizatorilor;
58. Sincronizarea parolelor este realizată automat de către:
- a) utilitarul SECURITY;
  - b) supervisor;
  - c) utilitarul MAP;
  - d) managerul de grup;
  - e) utilitarul LOGIN;
59. CASTOFF All:
- a) șterge ecranul;
  - b) furnizează o listă cu toți utilizatorii rețelei;
  - c) permite trimiterea mesajelor către toți utilizatorii;
  - d) blochează primirea mesajelor de la toate stațiile de lucru;
  - e) blochează primirea tuturor mesajelor;

60. Utilitarul COLORPAL permite:
- a) modificarea culorii pentru toate textele urmează să fie scrise pe ecran;
  - b) ștergerea ecranului;
  - c) selecția paletei de culori pentru modul grafic;
  - d) modificarea numărului de caractere de pe fiecare linie (40 sau 80);
  - e) ștergerea unei palete predefinite;
61. Utilitarul FLAG:
- a) permite atribuirea și retragerea de drepturi pe fișiere;
  - b) este urmat întotdeauna de o listă de parametri;
  - c) permite modificarea atributelor fișierelor și directoarelor;
  - d) se lansează de la consola file-serverului;
  - e) suportă caracterele \* și ?;
62. Masca de drepturi moștenite a unui director:
- a) nu influențează masca de drepturi moștenite a fișierelor din director;
  - b) permite modificarea atributelor fișierelor din director;
  - c) poate fi modificată prin adăugarea sau retragerea de drepturi;
  - d) facilitează accesul mai rapid la datele din director;
  - e) nu este afectată de masca de drepturi moștenite a subdirectoarelor sale;
63. Următoarea comandă este corectă:
- a) MAP abc TO F:\;
  - b) MAP INS TO F:\PUBLIC;
  - c) MAP X:\PUBLIC:=Y:\STUDENT;
  - d) MAP DEL := F:\;
  - e) MAP F::;
64. Utilitarul PURGE permite:
- a) modificarea atributelor fișierelor Novell;
  - b) eliberarea zonei de memorie folosită de ultimul program executat;
  - c) marcarea sub Novell a sectoarelor defecte de pe disc;
  - d) recuperarea datelor din fișierele partajate;
  - e) ștergerea definitivă a fișierelor șterse anterior;
65. REMOVE DOS are ca efect:
- a) ștergerea fișierelor sistemului de operare DOS de pe server;
  - b) ștergerea fișierelor sistemului de operare DOS de pe stația de lucru;
  - c) încărcarea unei noi copii a interpretorului de comenzi command.com;
  - d) eliberarea memoriei ocupate de DOS pe stația de lucru;
  - e) eliberarea memoriei ocupate de DOS pe server;
66. Utilitarul SALVAGE permite:
- a) salvarea datelor de pe hard-disk;
  - b) scrierea într-un fișier a datelor programului ce și-a încheiat execuția;
  - c) scrierea pe disc a unei zone de memorie RAM;
  - d) refacerea atributelor fișierelor ce au fost modificate;

- e) vizualizarea numelor fișierelor șterse;
67. Cu ajutorul utilitarului SYSCON nu se poate realiza:
- a) modificarea drepturilor unui utilizator;
  - b) modificarea parolei unui utilizator;
  - c) dezactivarea unui cont;
  - d) adăugarea sau ștergerea unui utilizator din grup;
  - e) modificarea atributelor fișierelor și directoarelor din rețea;
68. În utilitarele Novell cu meniu se folosește pentru selecție multiplă:
- a) tasta Insert;
  - b) combinația Shift+PgDn sau Shift+PgUp;
  - c) combinația Shift+una din tastele cu săgeți;
  - d) tasta F10;
  - e) tasta F5;
69. WHOAMI /A furnizează și:
- a) o listă cu atributele fișierelor din directorul curent;
  - b) numele și versiunea sistemului de operare instalat pe stația de lucru;
  - c) o listă cu restricțiile pe file-serverul curent;
  - d) o listă cu utilizatorii ce au drept de acces în directorul curent;
  - e) echivalența în securitate pentru utilizatorul conectat în rețea;
70. Unui utilizator nu i se poate interzice:
- a) accesul la file-server;
  - b) modificarea parolei;
  - c) execuția programelor din directorul PUBLIC;
  - d) ștergerea unui fișier dintr-un director pe care are dreptul ERASE;
  - e) lansarea unui program de pe un disc local;

| Test | Raspuns |
|------|---------|
| 1    | e       |
| 2    | e       |
| 3    | e       |
| 4    | e       |
| 5    | e       |
| 6    | e       |
| 7    | e       |
| 8    | e       |
| 9    | e       |
| 10   | e       |
| 11   | e       |
| 12   | e       |
| 13   | e       |
| 14   | e       |
| 15   | e       |
| 16   | e       |
| 17   | e       |
| 18   | e       |
| 19   | e       |
| 20   | e       |
| 21   | e       |
| 22   | e       |
| 23   | e       |
| 24   | e       |
| 25   | e       |
| 26   | e       |
| 27   | e       |
| 28   | e       |
| 29   | e       |
| 30   | e       |
| 31   | e       |
| 32   | e       |
| 33   | e       |
| 34   | e       |
| 35   | e       |
| 36   | e       |
| 37   | e       |
| 38   | e       |
| 39   | e       |
| 40   | e       |

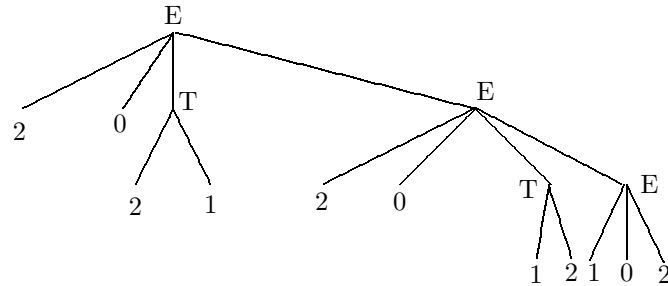
| Test | Raspuns |
|------|---------|
| 41   | e       |
| 42   | e       |
| 43   | e       |
| 44   | e       |
| 45   | e       |
| 46   | e       |
| 47   | e       |
| 48   | e       |
| 49   | e       |
| 50   | e       |
| 51   | e       |
| 52   | e       |
| 53   | e       |
| 54   | e       |
| 55   | e       |
| 56   | e       |
| 57   | e       |
| 58   | e       |
| 59   | e       |
| 60   | e       |
| 61   | e       |
| 62   | e       |
| 63   | e       |
| 64   | e       |
| 65   | e       |
| 66   | e       |
| 67   | e       |
| 68   | e       |
| 69   | e       |
| 70   | e       |

## 7 Teoria compilatoarelor

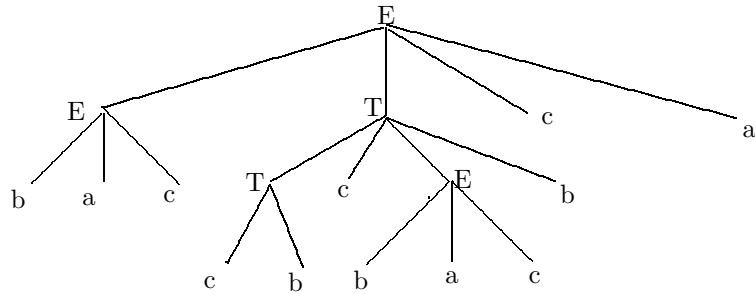
1. Se consideră schema de traducere orientată de sintaxă  $S = (\{E, T\}; \{0, 1, 2\}; R; E)$  unde  $R$ :

$$\begin{aligned} E &\rightarrow 20TE | ETca \\ T &\rightarrow 1E2T | TcEb \\ E &\rightarrow 102 | bac \\ T &\rightarrow 21 | cb \end{aligned}$$

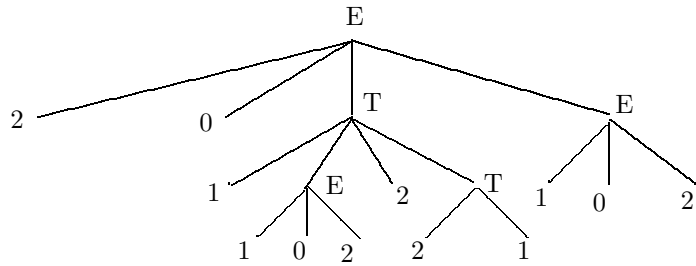
Stabiliți care dintre următorii arbori corespunde pentru traducerea secvenței  $w = 201102221102$ .



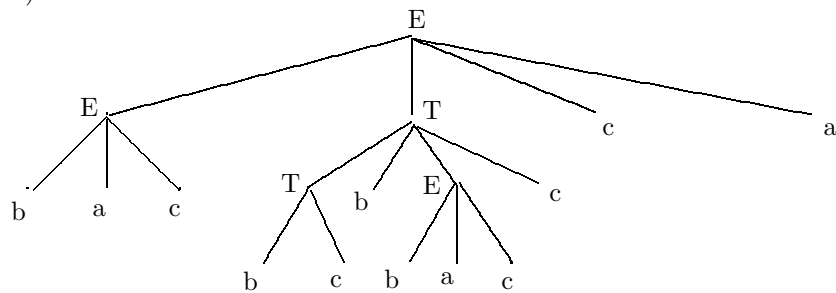
a)



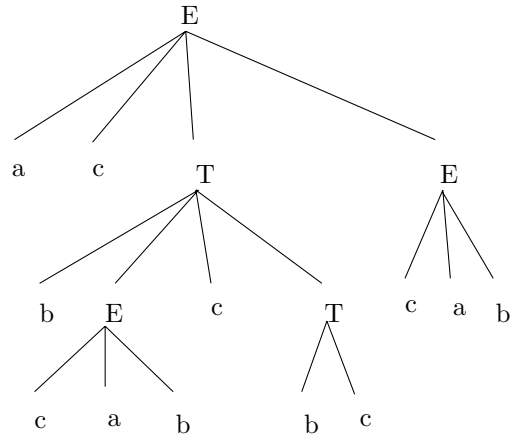
b)



c)



d)



e)

2. Se consideră traducătorul pushdown:

$$T = (\{q\}; \{a, +, (, *, )\}; \{a, +, *, (, )\}; E; \{1, 2, 3, 4\}; \delta; E; \{q\})$$

$$\delta(q, \lambda, E) = \{(q, E + E, 1); (q, E * E, 2); (q, (E), 3); (q, a, 4)$$

$$\delta(q, a, a) = (q, \lambda, \lambda) \forall a \in \Sigma$$

Traducerea secvenței  $w = a + (a * a + a)$  este:

- a) 13142444
  - b) 14341244
  - c) 14312444
  - d) 12144324
  - e) 13412444
3. Fie  $T = (Q, \Sigma, \Gamma, \Delta, \delta, q_0, Z_0, F)$  un traducător pushdown unde  $\delta$  reprezintă funcția de tranziție. Care dintre următoarele definiții este corectă?

a)

$$\delta : Q \times \Sigma \times \Gamma^* \rightarrow P_f(Q \times \Gamma \times \Delta)$$

b)

$$\delta : Q \times \Gamma \times \Sigma^* \rightarrow P_f(Q \times \Delta^* \times \Gamma)$$

c)

$$\delta : Q \times \Sigma \cup \{\lambda\} \times \Gamma \rightarrow P_f(Q \times \Gamma^* \times \Delta^*)$$

d)

$$\Delta : Q \times \Sigma^* \times \Gamma \rightarrow P_f(Q \times \Delta^*)$$

e)

$$\Delta : Q \times \Gamma^* \times \Sigma \rightarrow P_f(Q \times \Delta \times \Sigma)$$

4. Fie traducătorul finit  $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ , unde  $Q = \{q_0, q_1, q_2, q_3\}$ ,  $\Sigma = \{a, b, c\}$ ,  $\Delta = \{0, 1, 2\}$ ,  $F = \{q_2, q_3\}$ . Funcția de tranziție este definită astfel:

$$\begin{array}{llll} \delta(q_0, a) = (q_3, 0) & \delta(q_1, a) = (q_1, 0) & \delta(q_2, a) = (q_3, 0) & \delta(q_3, a) = (q_1, 0) \\ \delta(q_0, b) = (q_1, 1) & \delta(q_1, b) = (q_2, 1) & \delta(q_2, c) = (q_2, 2) & \delta(q_3, b) = (q_3, 1) \\ \delta(q_0, c) = (q_3, 2) & \delta(q_1, c) = (q_0, 2) & & \end{array}$$

Traducerea secvenței  $babcabcc$  este:

- a) 1011200101
- b) 2022100202
- c) 0100211010
- d) 1211022121

e) 1012010122

5. ie traducatorul finit  $T = (Q, \Sigma, \Delta, \delta, q_0, F)$ , unde  $Q = \{q_0, q_1, q_2, q_3, q_4\}$ ,  $\Sigma = \{a, +, -\}$ ,  $\Delta = \{a, +, -\}$ ,  $F = \{q_1\}$ .  
 Functia de tranziție este definită astfel;

$$\begin{array}{lll} \delta(q_0, a) = (q_1, a) & \delta(q_0, +) = (q_0, \epsilon) & \delta(q_0, -) = (q_4, \epsilon) \\ \delta(q_1, +) = (q_2, \epsilon) & \delta(q_1, -) = (q_3, \epsilon) & \\ \delta(q_2, a) = (q_1, +a) & \delta(q_2, +) = (q_2, \epsilon) & \delta(q_2, -) = (q_2, \epsilon) \\ \delta(q_3, a) = (q_1, -a) & \delta(q_3, +) = (q_3, \epsilon) & \delta(q_3, -) = (q_3, \epsilon) \\ \delta(q_4, a) = (q_1, -a) & \delta(q_4, +) = (q_4, \epsilon) & \delta(q_4, -) = (q_0, \epsilon) \end{array}$$

Traducerea secvenței  $w = + - -a - +a + +-$  este:

- a)  $-a - a+$   
 b)  $+a - a-$   
 c)  $a - a-$   
 d) nu se poate traduce  
 e)  $-a + a$

6. Se consideră gramatica  $G = (\{S, A, B\}; \{a, b, c\}; S; P)$  unde  $P$ :

$$\begin{array}{l} S \rightarrow aAaB|bAbB \\ A \rightarrow a|ab \\ B \rightarrow aB|a \end{array}$$

Valorile funcției  $PRIM_3$  sunt:

a)

$$\begin{array}{l} PRIM_3(a) = \{a\} \\ PRIM_3(b) = \{b\} \\ PRIM_3(S) = \{a^3, a^2b, bab\} \\ PRIM_3(A) = \{a, ab\} \\ PRIM_3(B) = \{a, a^2, a^3\} \end{array}$$

b)

$$\begin{array}{l} PRIM_3(a) = \{a\} \\ PRIM_3(b) = \{b\} \\ PRIM_3(S) = \{a, b\} \\ PRIM_3(A) = \{a\} \\ PRIM_3(B) = \{b\} \end{array}$$

c)

$$\begin{array}{l} PRIM_3(a) = \{a\} \\ PRIM_3(b) = \{b\} \\ PRIM_3(S) = \{a^2, ba\} \\ PRIM_3(A) = \{a, ab\} \\ PRIM_3(B) = \{a, a^2\} \end{array}$$

d)

$$\begin{array}{l} PRIM_3(a) = \{a\} \\ PRIM_3(b) = \{b\} \\ PRIM_3(S) = \{ab, ba\} \\ PRIM_3(A) = \{a, ab\} \\ PRIM_3(B) = \{a, a^2, a^3\} \end{array}$$

e)

$$\begin{array}{l} PRIM_3(a) = \{a\} \\ PRIM_3(b) = \{b\} \\ PRIM_3(S) = \{a^3, ab, b^3\} \\ PRIM_3(A) = \{a^2, b^2\} \\ PRIM_3(B) = \{a, a^2, a^3\} \end{array}$$



7. Fie gramatica cu producțiile:

$$\begin{aligned} S &\rightarrow 0S11 \\ S &\rightarrow 011 \end{aligned}$$

Relațiile de precedență Wirth-Weber stabilite între simbolurile 1 și 1 sunt:

- a) relația <
- b) relația =
- c) relația =>
- d) relația =<
- e) relația >

8. Fie gramatica unde  $G = (\{S, A, B\}; \{a, b\}; S; P)$  unde  $P$ :

$$\begin{aligned} S &\rightarrow aA|bB \\ A &\rightarrow bAa|abA|b \\ B &\rightarrow aB|bb \end{aligned}$$

Valorile funcției  $Prim_2$  sunt:

a)

$$\begin{aligned} PRIM_2(a) &= \{a\} \\ PRIM_2(b) &= \{b\} \\ PRIM_2(S) &= \{ab, ba, bb, aa\} \\ PRIM_2(A) &= \{bb, ab, ba, b\} \\ PRIM_3(B) &= \{bb, ab, aa\} \end{aligned}$$

b)

$$\begin{aligned} PRIM_2(a) &= \{a\} \\ PRIM_2(b) &= \{b\} \\ PRIM_2(S) &= \{ab, ba, bb\} \\ PRIM_2(A) &= \{bb, ab, ba, b, a\} \\ PRIM_2(B) &= \{bb, ab\} \end{aligned}$$

c)

$$\begin{aligned} PRIM_2(a) &= \{a\} \\ PRIM_2(b) &= \{b\} \\ PRIM_2(S) &= \{ba, bb, aa\} \\ PRIM_2(A) &= \{bb, ba, b\} \\ PRIM_2(B) &= \{ab, aa\} \end{aligned}$$

d)

$$\begin{aligned} PRIM_2(a) &= \{a\} \\ PRIM_2(b) &= \{b\} \\ PRIM_2(S) &= \{ba, aa\} \\ PRIM_2(A) &= \{aa, bb, ba, b\} \\ PRIM_2(B) &= \{ab, aa, b\} \end{aligned}$$

e)

$$\begin{aligned} PRIM_2(a) &= \{a\} \\ PRIM_2(b) &= \{b\} \\ PRIM_2(S) &= \{ba, bb\} \\ PRIM_2(A) &= \{bb, ba, a\} \\ PRIM_2(B) &= \{ab, aa, ba\} \end{aligned}$$

9. Fie gramatica  $G = (\{E\}; \{a, b, 0, 1\}; E; P)$  de tip  $LR(1)$  unde mulțimea de producții este  $P$ :

$$\begin{aligned} E &\rightarrow E1aE0 \\ E &\rightarrow b \end{aligned}$$

Utilizând algoritmul pentru calculul  $V_k^G$ , să se determine valorile pentru  $V_1(E1a)$ :

a)

$$\{[E \rightarrow E1a.E0, 1]; [E \rightarrow .E1aE0, 0/1]\}$$

b)

$$\{[E \rightarrow E1a.E0, 0/1]; [E \rightarrow .E1aE0, \epsilon/1]; [E \rightarrow .S, 0/1]\}$$

c)

$$\{[E \rightarrow E1a.E0, \epsilon/1]; [E \rightarrow .E1aE0, 0/1]; [E \rightarrow .b, 0/1]\}$$

d)

$$\{[E \rightarrow E1a.E0, \epsilon/1]; [E \rightarrow E.1aE0, 0/1]; [E \rightarrow .S, 0/1]\}$$

e)

$$\{[E \rightarrow E1.aE0, 0/1]; [E \rightarrow .b, 0/1]\}$$

10. Fie gramatica cu productiile:

$$S \rightarrow bAAc$$

$$A \rightarrow Sa$$

$$A \rightarrow a$$

Relațiile de precedență Wirth-Weber stabilite între simbolurile  $S$  și  $A$  sunt:

a) relația  $>$

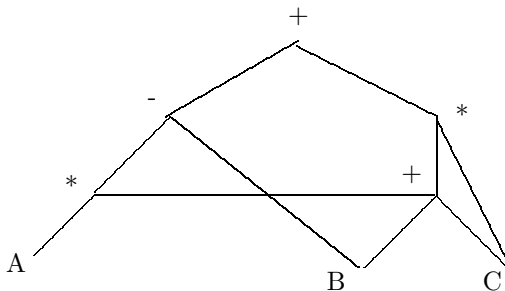
b) relația  $<$

c) relația  $\Rightarrow$

d) nu se stabilește nici o relație

e) relația  $=$

11. Fie expresia  $A * (B + C) - B + (B + C) * C$ . Graful orientat fără cicluri asociat expresiei este



Folosind algoritmul de etichetare a nodurilor determinați numărul de regiștri necesari pentru generarea codului obiect:

a) 1

b) 2

c) 3

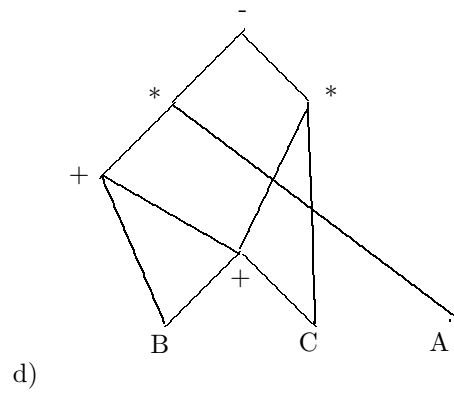
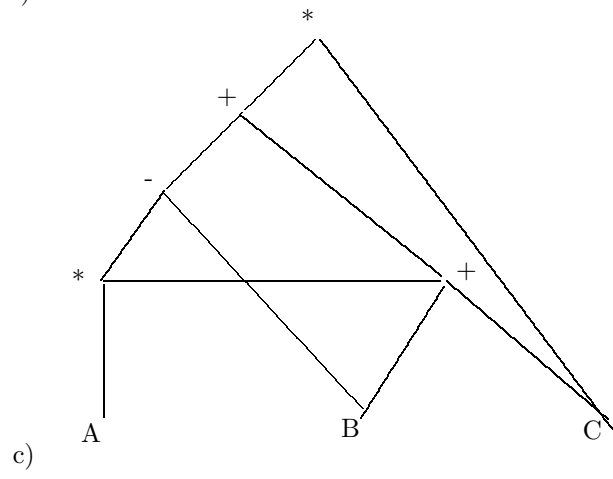
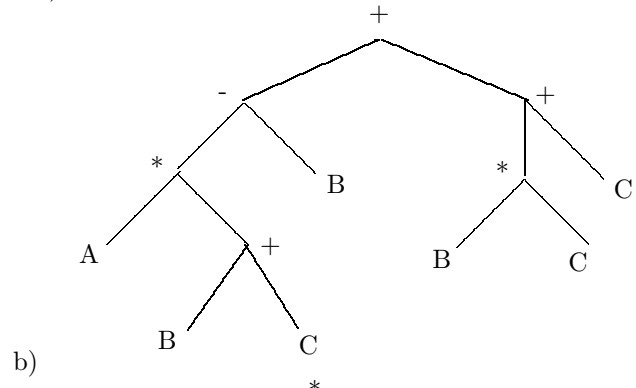
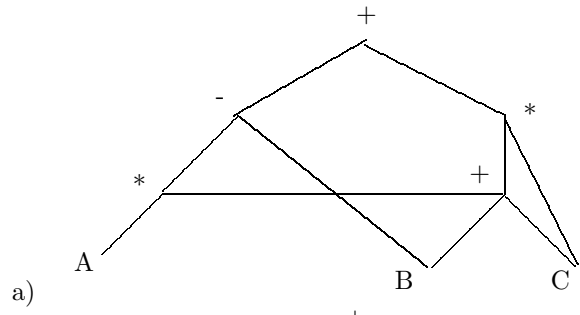
d) 4

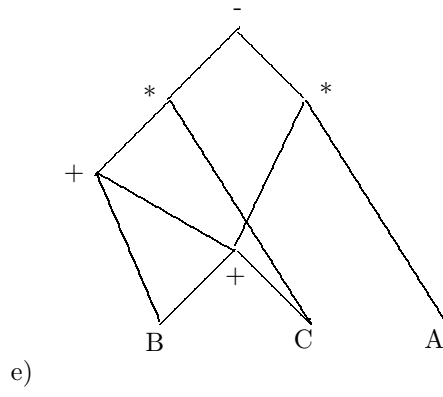
e) 5

12. Pentru generarea codului obiect corespunzător expresiei

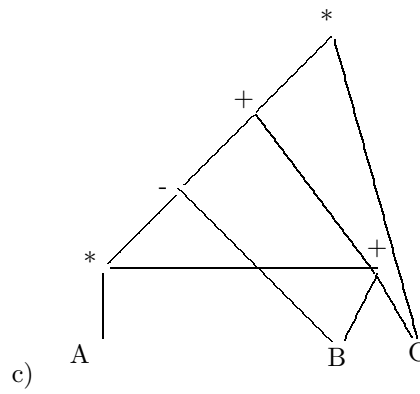
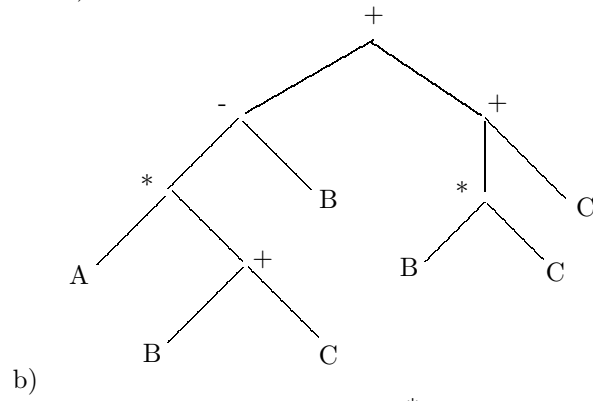
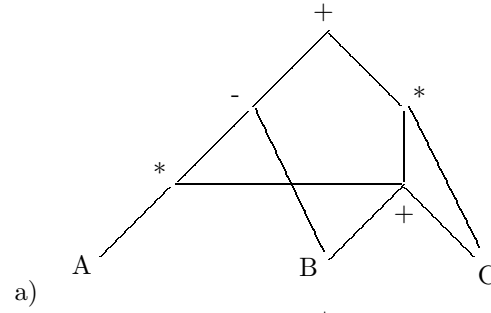
$$A * (B + C) - B + (B + C) * C$$

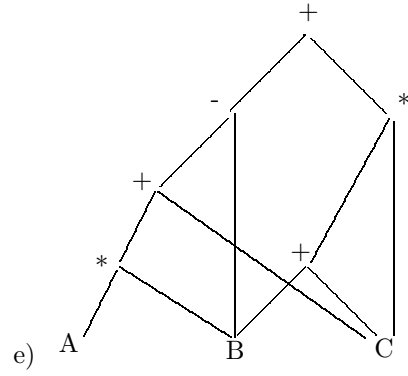
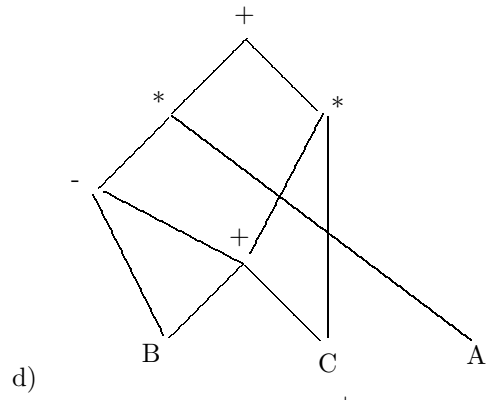
s-a asociat graful orientat fără cicluri corespunzător. Alegeți varianta corectă:





13. Pentru generarea codului obiect  $A * B + C - B + (B + C) * C$  corespunzător expresiei s-a asociat graful orientat fără cicluri corespunzător. Alegeți varianta corectă :

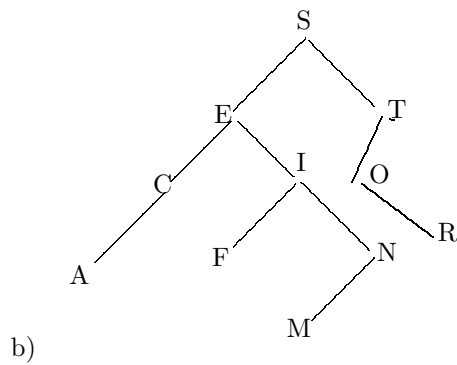
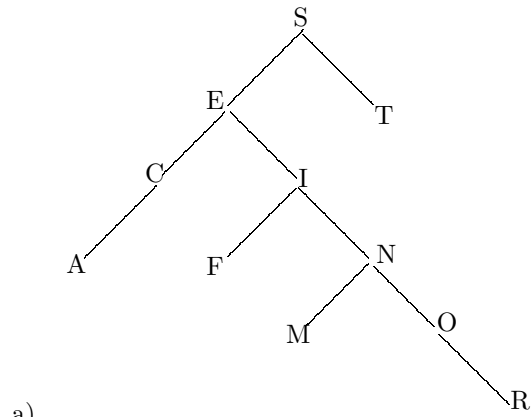


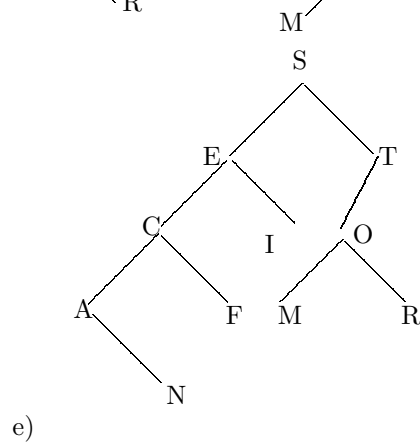
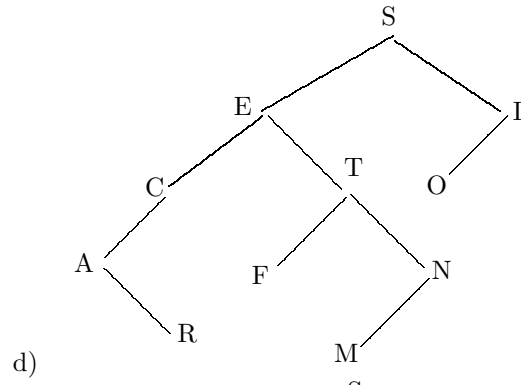
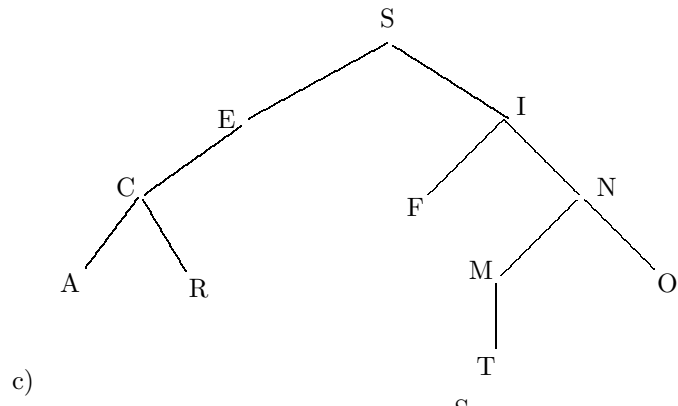


14. Alegeți tabela arborescentă corespunzătoare pentru textul :

SECȚIA INFORMATICĂ

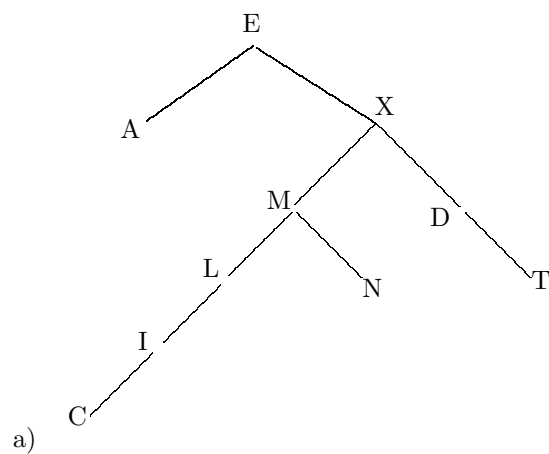
Obs: Nu se face diferență între literele A,Ă și T,Ț.

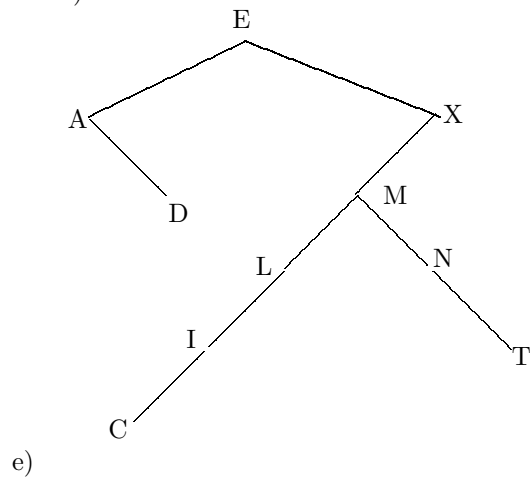
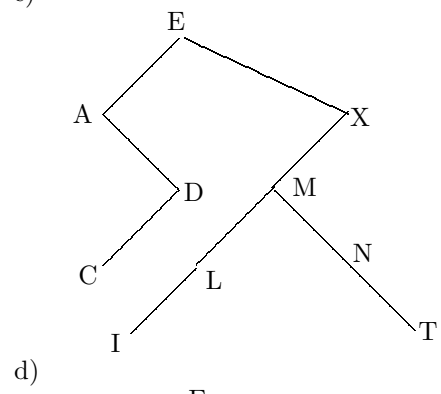
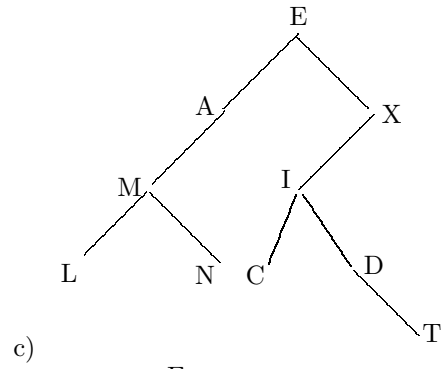
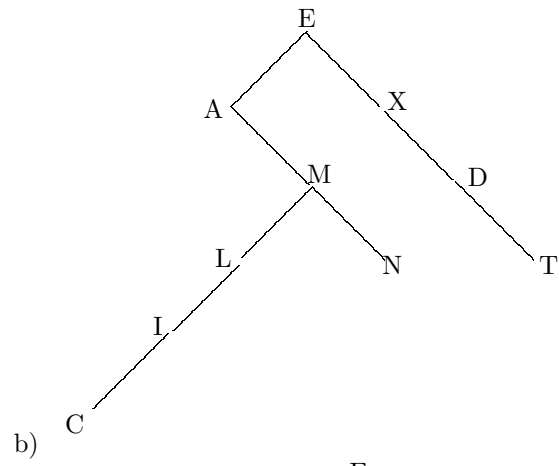




15. Alegeți tabela arborescentă corespunzătoare pentru textul

EXAMEN DE LICENȚĂ





| Test | Raspuns |
|------|---------|
| 1    | b       |
| 2    | c       |
| 3    | c       |
| 4    | e       |
| 5    | d       |
| 6    | a       |
| 7    | c       |
| 8    | a       |
| 9    | c       |
| 10   | d       |
| 11   | b       |
| 12   | a       |
| 13   | e       |
| 14   | a       |
| 15   | d       |

| Test | Raspuns |
|------|---------|
|------|---------|



## 8 Complexitatea algoritmilor

1. Pentru ce valoare a lui  $k$  are loc relația  $\log(n+h) = \log n + \frac{h}{n} + o(\frac{1}{n^k})$  ( $h$  este o constantă).

- a)  $k = 1$
- b)  $k = 2$
- c)  $k = 3$
- d)  $k = 4$
- e)  $k = n$

2. Fie  $f, g : N \rightarrow N$  definite astfel:

$$f(n) = \begin{cases} 2, & n \text{ par} \\ 1, & n \text{ impar} \end{cases} \quad g(n) = \begin{cases} 3, & n \text{ par} \\ 4, & n \text{ impar} \end{cases}$$

Care din următoarele afirmații este adevărată ?

- a)  $f(n) = O(g(n))(n \rightarrow \infty)$
- b)  $g(n) = O(f(n))(n \rightarrow \infty)$
- c)  $g(n) \sim f(n)(n \rightarrow \infty)$
- d)  $f(n) \in \Omega(g(n))(n \rightarrow \infty)$
- e)  $g(n) = o(f(n))(n \rightarrow \infty)$

3. Fie  $f, g : N \rightarrow N$  definite astfel:

$$f(n) = \binom{n}{k} \quad g(n) = n^k, \quad (0 < k < n)$$

Care din următoarele afirmații este adevărată ?

- a)  $f(n) = o(g(n))(n \rightarrow \infty)$
- b)  $g(n) \notin O(f(n))(n \rightarrow \infty)$
- c)  $g(n) \sim f(n)(n \rightarrow \infty)$
- d)  $f(n) \notin O(g(n))(n \rightarrow \infty)$
- e)  $f(n) \in O(g(n))(n \rightarrow \infty)$

4. Fie  $f, g : N \rightarrow N$  definite astfel:

$$f(n) = \sum_{k=1}^n (2k)^2, \quad g(n) = \sum_{k=1}^n (2k-1)^2$$

Care din următoarele afirmații este adevărată:

- a)  $g(n) \notin O(f(n))(n \rightarrow \infty)$
- b)  $f(n) \notin O(g(n))(n \rightarrow \infty)$
- c)  $g(n) = o(f(n))(n \rightarrow \infty)$
- d)  $g(n) \sim f(n)(n \rightarrow \infty)$
- e)  $f(n) = o(g(n))(n \rightarrow \infty)$

5. Fie  $f, g : N \rightarrow N$  definite astfel:

$$f(n) = \sum_{k=1}^n k!, \quad g(n) = (2n)!$$

Care din următoarele afirmații este adevărată ?

- a)  $g(n) \in O(f(n))(n \rightarrow \infty)$
- b)  $f(n) \in \omega(g(n))(n \rightarrow \infty)$

- c)  $f(n) = o(g(n))(n \rightarrow \infty)$   
 d)  $g(n) = o(f(n))(n \rightarrow \infty)$   
 e)  $f(n) \sim g(n)(n \rightarrow \infty)$
6. Funcția  $f(y) = \frac{1}{1-y} + 2y^3$  este funcție generatoare pentru șirul:
- a)  $0, 1, 2, 3, 4, \dots$   
 b)  $1, 2, 3, 4, \dots$   
 c)  $2, 4, 6, 8, \dots$   
 d)  $1, 1, 3, 3, 3, \dots$   
 e)  $1, 1, 1, 3, 1, 1, 1, \dots$
7.  $f(y) = \frac{y+1}{(1-y)^3}$  este funcție generatoare pentru șirul:
- a)  $1, 2, 3, 4, \dots$   
 b)  $0^3, 1^3, 2^3, 3^3, \dots$   
 c)  $1^2, 2^2, 3^2, \dots$   
 d)  $1, 1, 1, 0, 0, 0, \dots$   
 e)  $\binom{n}{0}, \binom{n}{1}, \binom{n}{2}, \dots, \binom{n}{n}, 0, 0, 0, \dots$
8. Care este funcția generatoare pentru șirul  $1, (1+h), (1+h)^2, (1+h)^3, \dots (h \in R)$
- a)  $f(y) = \frac{1}{1-y}$   
 b)  $f(y) = \frac{y^2}{1-hy}$   
 c)  $f(y) = (1-hy)^{-3}$   
 d)  $f(y) = \frac{1}{1-(1+h)y}$   
 e)  $f(y) = \frac{1}{1-hy}$
9. Care este funcția generatoare pentru șirul  $0, 3^0, 3^1, 3^2, 3^3, 3^4, \dots$
- a)  $f(y) = \sum_{r=0}^{\infty} 3^r y^{r+1}$   
 b)  $f(y) = \frac{1}{1-3y}$   
 c)  $f(y) = \frac{y}{1-3y}$   
 d)  $f(y) = \frac{1}{1+y}$   
 e)  $f(y) = \frac{y^2}{1-3y}$
10. Care este funcția generatoare pentru șirul  $2, 1+h, 1+h^2, 1+h^3, \dots (h \in R)$
- a)  $f(y) = \frac{1}{1-y}$   
 b)  $f(y) = \frac{y^2}{1-hy}$   
 c)  $f(y) = (1-hy)^{-1}$   
 d)  $f(y) = \frac{1}{1-(1+h)y}$   
 e)  $f(y) = \frac{1}{1-y} + \frac{1}{1-hy}$
11. Care este marginea asimptotică tare pentru relația de recurență:

$$f(n) = 9f\left(\frac{n}{10}\right) + n$$

- a)  $f(n) = \Theta(n^2)$   
 b)  $f(n) = \Theta(\sqrt{n} \log n)$   
 c)  $f(n) = \Theta(1)$

d)  $f(n) = \Theta(n^{\log 7})$

e)  $f(n) = \Theta(n)$

12. Care este marginea asimptotică tare pentru relația de recurență:

$$f(n) = 16f\left(\frac{n}{4}\right) + n^2$$

a)  $f(n) = \Theta(n^3)$

b)  $f(n) = \Theta(n^2 \log n)$

c)  $f(n) = \Theta(n)$

d)  $f(n) = \Theta(\sqrt{n} \log n)$

e)  $f(n) = \Theta(1)$

13. Care este marginea asimptotică tare pentru relația de recurență:

$$f(n) = 7f\left(\frac{n}{3}\right) + n^2$$

a)  $f(n) = \Theta(1)$

b)  $f(n) = \Theta(n^2 \log n)$

c)  $f(n) = \Theta(n^2)$

d)  $f(n) = \Theta(n)$

e)  $f(n) = \Theta(n^{\log 5})$

14. Care este marginea asimptotică tare pentru relația de recurență:

$$f(n) = 2f\left(\frac{n}{4}\right) + \sqrt{n}$$

a)  $f(n) = \Theta(n^2)$

b)  $f(n) = \Theta(1)$

c)  $f(n) = \Theta(n^2 \log n)$

d)  $f(n) = \Theta(\sqrt{n} \log n)$

e)  $f(n) = \Theta(n\sqrt{n})$

15. Care este marginea asimptotică tare pentru relația de recurență:

$$f(n) = f(\sqrt{n}) + 1$$

a)  $f(n) = \Theta(n^2)$

b)  $f(n) = \Theta(\sqrt{n} \log n)$

c)  $f(n) = \Theta(\log \log n)$

d)  $f(n) = \Theta(\log n)$

e)  $f(n) = \Theta(n)$

| Test | Raspuns |
|------|---------|
| 1    | a       |
| 2    | a       |
| 3    | e       |
| 4    | d       |
| 5    | c       |
| 6    | e       |
| 7    | c       |
| 8    | d       |
| 9    | c       |
| 10   | e       |
| 11   | e       |
| 12   | b       |
| 13   | c       |
| 14   | d       |
| 15   | c       |

| Test | Raspuns |
|------|---------|
|------|---------|

## 9 Calcul numeric

1. Pentru a realiza o factorizare QR a matricei:

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

sunt necesare două matrici de tip Householder. Acestea sunt:

a)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

b)

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

c)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

d)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

e)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

2. Pentru a realiza o factorizare QR a matricei:

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

sunt necesare două matrici de tip Householder. Acestea sunt:

a)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

b)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{6}}{3} \\ 0 & -\frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

c)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

d)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

e)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

3. Pentru a realiza o factorizare QR a matricei:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \\ 1 & 2 & 0 \end{pmatrix}$$

sunt necesare două matrici de tip Householder. Acestea sunt:

a)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix}$$

b)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix}$$

c)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{6}}{3} \\ 0 & -\frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \end{pmatrix}$$

d)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix}$$

e)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \end{pmatrix}$$

4. Pentru a realiza o factorizare QR a matricei:

$$A = \begin{pmatrix} 1 & 0 & -1 \\ 0 & 1 & 1 \\ 1 & 1 & 0 \end{pmatrix}$$

sunt necesare două matrici de tip Householder. Acestea sunt:

a)

$$\begin{pmatrix} \frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{pmatrix}$$

b)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ \frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{pmatrix}$$

c)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{pmatrix}$$

d)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{3} & -\frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{pmatrix}$$

e)

$$\begin{pmatrix} -\frac{\sqrt{2}}{2} & 0 & -\frac{\sqrt{2}}{2} \\ 0 & 1 & 0 \\ -\frac{\sqrt{2}}{2} & 0 & \frac{\sqrt{2}}{2} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{6}}{3} & \frac{\sqrt{3}}{3} \\ 0 & -\frac{\sqrt{3}}{3} & \frac{\sqrt{6}}{3} \end{pmatrix}$$

5. Pentru a realiza o factorizare QR a matricei:

$$A = \begin{pmatrix} 1 & 0 & 2 \\ 1 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$$

sunt necesare două matrici de tip Householder. Acestea sunt:

a)

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ \frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} \\ \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

b)

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} \\ \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

c)

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} \\ \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \end{pmatrix}$$

d)

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} \\ \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & \frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

e)

$$\begin{pmatrix} -\frac{\sqrt{3}}{3} & -\frac{\sqrt{3}}{3} & \frac{\sqrt{3}}{3} \\ -\frac{\sqrt{3}}{3} & \frac{3+\sqrt{3}}{6} & \frac{3-\sqrt{3}}{6} \\ \frac{\sqrt{3}}{3} & \frac{3-\sqrt{3}}{6} & \frac{3+\sqrt{3}}{6} \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} \\ 0 & -\frac{\sqrt{2}}{2} & \frac{\sqrt{2}}{2} \end{pmatrix}$$

6. O factorizare LR a matricei

$$A = \begin{pmatrix} 1 & 2 & 4 \\ 2 & 3 & 1 \\ -1 & -1 & 2 \end{pmatrix}$$

este realizată de matricele:

a)

$$\begin{pmatrix} 1 & 2 & 4 \\ 0 & -1 & -7 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 2 & -4 \\ 0 & -1 & -7 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 2 & 4 \\ 0 & -1 & 7 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 2 & 4 \\ 0 & -1 & -7 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 1 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 2 & 4 \\ 0 & -1 & -7 \\ 0 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & -1 & 1 \end{pmatrix}$$

7. O factorizare LR a matricei

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

este realizată de matricele:

a)

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

8. O factorizare LR a matricei

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ -1 & 1 & 0 \end{pmatrix}$$

este realizată de matricele:

a)

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & 3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$$



b)

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & 2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & -2 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 1 & 1 \\ 0 & 1 & -2 \\ 0 & 0 & -3 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 2 & 1 \end{pmatrix}$$

9. O factorizare LR a matricei

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

este realizată de matricele:

a)

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & 5 \\ 0 & 0 & 18 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & 1 & -5 \\ 0 & 0 & 18 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -5 \\ 0 & 0 & 18 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 5 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -5 \\ 0 & 0 & 18 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 5 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 2 & 3 \\ 0 & -1 & -5 \\ 0 & 0 & 18 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & -5 & 1 \end{pmatrix}$$

10. O factorizare LR a matricei

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 3 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

este realizată de matricele:

a)

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 2 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & -2 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & -1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & -1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & -2 & 1 \end{pmatrix}$$

11. Pentru a transforma matricea:

$$A = \begin{pmatrix} 1 & -1 & 2 \\ 2 & 1 & 1 \\ 1 & 2 & -3 \end{pmatrix}$$

într-o matrice superior triunghiulară se pot folosi matricele inferior triunghiulare:

a)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

12. Pentru a transforma matricea:

$$A = \begin{pmatrix} 1 & 1 & 0 \\ 2 & 3 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

într-o matrice superior triunghiulară se pot folosi matricele inferior triunghiulare:

a)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

13. Pentru a transforma matricea:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 2 & 3 & 1 \\ 3 & 1 & 2 \end{pmatrix}$$

într-o matrice superior triunghiulară se pot folosi matricele inferior triunghiulare:

a)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -5 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 0 \\ -2 & 1 & 0 \\ 3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 0 \\ 2 & 1 & 0 \\ -3 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 5 & 1 \end{pmatrix}$$

14. Pentru a transforma matricea:

$$A = \begin{pmatrix} 1 & 0 & 1 \\ -1 & 1 & 1 \\ 0 & 1 & -1 \end{pmatrix}$$

într-o matrice superior triunghiulară se pot folosi matricele inferior triunghiulare:

a)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -1 & -1 \end{pmatrix}$$

15. Pentru a transforma matricea:

$$A = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 2 & 3 \\ -1 & 1 & 0 \end{pmatrix}$$

într-o matrice superior triunghiulară se pot folosi matricele inferior triunghiulare:

a)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

b)

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

c)

$$\begin{pmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

d)

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 2 & 1 \end{pmatrix}$$

e)

$$\begin{pmatrix} 1 & 0 & 0 \\ -1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & -2 & 1 \end{pmatrix}$$

16. Dacă o matrice  $A$  are polinomul caracteristic:

$$P_A(\lambda) = \lambda^4 - 10\lambda^3 + 26\lambda^2 - 10\lambda + 1$$

atunci valoarea determinantului matricei  $A$  este:

a) 1

b) -1

c) 10

d) -10

e) 26

17. Dacă o matrice  $A$  are polinomul caracteristic:

$$P_A(\lambda) = \lambda^4 - 3\lambda^2 + \lambda$$

atunci valoarea determinantului matricei  $A$  este:

- a) 1
- b) 0
- c) 3
- d) -3
- e) 2

18. Dacă o matrice  $A$  are polinomul caracteristic:

$$P_A(\lambda) = \lambda^3 - 3\lambda + 2$$

atunci valoarea determinantului matricei  $A$  este:

- a) 2
- b) 3
- c) -2
- d) -3
- e) 0

19. Dacă o matrice  $A$  are polinomul caracteristic:

$$P_A(\lambda) = \lambda^5 - 5\lambda^3 + 2\lambda - 1$$

atunci valoarea determinantului matricei  $A$  este:

- a) -1
- b) 2
- c) -5
- d) 1
- e) 0

20. Dacă o matrice  $A$  are polinomul caracteristic:

$$P_A(\lambda) = \lambda^3 - 2\lambda^2 + \lambda - 1$$

atunci valoarea determinantului matricei  $A$  este:

- a) 0
- b) -1
- c) 2
- d) -2
- e) 1

| Test | Raspuns |
|------|---------|
| 1    | a       |
| 2    | b       |
| 3    | c       |
| 4    | d       |
| 5    | e       |
| 6    | a       |
| 7    | b       |
| 8    | c       |
| 9    | d       |
| 10   | e       |
| 11   | a       |
| 12   | b       |
| 13   | c       |
| 14   | d       |
| 15   | e       |
| 16   | a       |
| 17   | b       |
| 18   | c       |
| 19   | d       |
| 20   | e       |

| Test | Raspuns |
|------|---------|
|------|---------|

## 10 Baze de date

1. Fie relația *persoana*, de schemă  $R=\{\text{COD, NUME, PRENUME, DATA-NAȘTERII, LOCALITATE, AVERE}\}$  notată pe scurt cu  $R=\{\text{C,N,P,D,L,A}\}$ . Specificați care din următoarele mulțimi de atribute pot fi chei ?
  - a)  $\{\text{C}\}, \{\text{N,P,L}\}, \{\text{N,P,D}\}$
  - b)  $\{\text{C,N,P}\}$
  - c)  $\{\text{N,P,L,D}\}$
  - d)  $\{\text{A,N,P}\}$
  - e)  $\{\text{A,D,L}\}$
2. Fie relația *persoana*, de schemă  $R=\{\text{COD, NUME, PRENUME, DATA-NAȘTERII, ADRESA, MESERIE}\}$  notată pe scurt cu  $R=\{\text{C,N,P,D,A,M}\}$ . Care expresie determină informaticienii din orașul Craiova ?
  - a)  $\sigma_{M='INF'}(\Pi_{A,M}(\textit{persoana}))$
  - b)  $\sigma_{A='CRAIOVA'}(\sigma_{M='INF'}(\textit{persoana}))$
  - c)  $\sigma_{A='CRAIOVA'}(\textit{persoana})$
  - d)  $\pi_A(\sigma_{M='INF'}(\textit{persoana}))$
  - e)  $\sigma_{M='INF'}(\pi_A(\textit{persoana}))$
3. Fie relația *salarizat*, de schemă  $R=\{\text{COD, NUME, PRENUME, DATA-NAȘTERII, ADRESA, MESERIE}\}$  notată pe scurt cu  $R=\{\text{C,N,P,D,A,M}\}$  și mulțimile de atribute  $X=\{\text{N,P,A}\}$  și  $Y=\{\text{N,P}\}$ . Care este relația între  $\text{card}(\pi_Y(\textit{salarizat}))$  și  $\text{card}(\pi_X(\textit{salarizat}))$  ? *card* reprezintă cardinalul unei mulțimi.
  - a)  $\leq$
  - b)  $\equiv$
  - c)  $\geq$
  - d)  $<$
  - e)  $=$
4. Fie relația *carte*, de schemă  $R=\{\text{COTA, NUME-AUTOR, PRENUME-AUTOR, TITLU, AN-APARIȚIE, PREȚ}\}$  și mulțimea de atribute  $X=\{\text{NUME-AUTOR, PRENUME-AUTOR, TITLU, AN-APARIȚIE}\}$ . Care din următoarele expresii este adevărată ?
  - a)  $\pi_X(\sigma_{\text{AN-APARIȚIE}=2000}(\textit{carte})) \subset \sigma_{\text{AN-APARIȚIE}=2000}(\pi_X(\textit{carte}))$
  - b)  $\pi_X(\sigma_{\text{AN-APARIȚIE}=2000}(\textit{carte})) \supseteq \sigma_{\text{AN-APARIȚIE}=2000}(\pi_X(\textit{carte}))$
  - c)  $\pi_X(\sigma_{\text{AN-APARIȚIE}=2000}(\textit{carte})) \not\subset \sigma_{\text{AN-APARIȚIE}=2000}(\pi_X(\textit{carte}))$
  - d)  $\pi_X(\sigma_{\text{AN-APARIȚIE}=2000}(\textit{carte})) = \sigma_{\text{AN-APARIȚIE}=2000}(\pi_X(\textit{carte}))$
  - e)  $\pi_X(\sigma_{\text{AN-APARIȚIE}=2000}(\textit{carte})) \neq \sigma_{\text{AN-APARIȚIE}=2000}(\pi_X(\textit{carte}))$
5. Comanda **CREATE TABLE** specifică:
  - a) Numai numele tabelului
  - b) Schema tabelului
  - c) Restricțiile pe care le satisfac coloanele
  - d) Tipurile coloanelor
  - e) Totul de la celelalte puncte
6. Fie relația *grafic* cu schema  $R=\{\text{NUME-PILOT, PRENUME-PILOT, CURSA, DATA, ORA-DECOLĂRII}\}$  notată pe scurt cu  $R=\{\text{NP,PP,CS,DT,OD}\}$ , data de tabelul

| NP    | PP  | CS  | DT    | OD    |
|-------|-----|-----|-------|-------|
| Ursu  | Ion | 83  | 9-06  | 10:15 |
| Ursu  | Ion | 116 | 10-06 | 11:15 |
| Jder  | Gh  | 83  | 13-06 | 10:15 |
| Jder  | Gh  | 116 | 9-06  | 11:15 |
| Lupu  | Cr  | 281 | 8-06  | 5:15  |
| Lupu  | Cr  | 83  | 11-06 | 10:15 |
| Lupu  | Cr  | 301 | 12-06 | 18:35 |
| Vulpe | Dr  | 281 | 18-06 | 5:15  |

care satisface restricția data de dependența funcțională  $CS \rightarrow OD$ . Care este rezultatul următoarei operații de actualizare

$CH(\text{grafic}: NP=Jder, PP=Gh, CS=83, DT=13-06, OD=10:15 |$

$NP=Jder, PP=Gh, CS=83, DT=13-06, OD=16:15)$  ce trebuie să țină cont de restricția dată.

- Nu modifică nici un tuplu deoarece contrazice  $CS \rightarrow OD$
  - Modifică ora din tuplu 3
  - Modifică tot tuplu 3
  - Modifică toate orele de decolare a cursei 83
  - Este nefuncțională
7. Fie relația *grafic*, cu schema  $R=\{NUME-PILOT, PRENUME-PILOT, CURSA, DATA, ORA-DECOLARII\}$  notată pe scurt cu  $R=\{NP, PP, CS, DT, OD\}$ , data de tabelul

| NP    | PP  | CS  | DT    | OD    |
|-------|-----|-----|-------|-------|
| Ursu  | Ion | 83  | 9-06  | 10:15 |
| Ursu  | Ion | 116 | 10-06 | 11:15 |
| Jder  | Gh  | 83  | 13-06 | 10:15 |
| Jder  | Gh  | 116 | 9-06  | 11:15 |
| Lupu  | Cr  | 281 | 8-06  | 5:15  |
| Lupu  | Cr  | 83  | 11-06 | 10:15 |
| Lupu  | Cr  | 301 | 12-06 | 18:35 |
| Vulpe | Dr  | 281 | 18-06 | 5:15  |

care satisface restricția data de dependența funcțională  $\{CS, DT\} \rightarrow \{NP, PP\}$ . Care este rezultatul următoarei operații:

$ADD(\text{grafic}: CS=116, DT=9-06, NP=Leu, PP=Cr, OD=11:15)$

- Adaugă un nou tuplu pentru că se ia în calcul dependența funcțională
  - Nu se poate adăuga deoarece într-o cursă, la o anumită dată, trebuie să existe numai un pilot
  - Inlocuiește tuplul 4
  - Se șterge tuplul 4
  - Adaugă un nou tuplu pentru că nu se ia calculul dependența funcțională
8. Fie relația *grafic*, cu schema  $R=\{NUME-PILOT, PRENUME-PILOT, CURSA, DATA, ORA-DECOLARII\}$  notată pe scurt cu  $R=\{NP, PP, CS, DT, OD\}$ , data de tabelul

| NP    | PP  | CS  | DT    | OD    |
|-------|-----|-----|-------|-------|
| Ursu  | Ion | 83  | 9-06  | 10:15 |
| Ursu  | Ion | 116 | 10-06 | 11:15 |
| Jder  | Gh  | 83  | 13-06 | 10:15 |
| Jder  | Gh  | 116 | 9-06  | 11:15 |
| Lupu  | Cr  | 281 | 8-06  | 5:15  |
| Lupu  | Cr  | 83  | 11-06 | 10:15 |
| Lupu  | Cr  | 301 | 12-06 | 18:35 |
| Vulpe | Dr  | 281 | 18-06 | 5:15  |



care satisface restricția dată de dependența funcțională  $\{NP, PP, DT, OD\} \rightarrow \{CS\}$ . Care este rezultatul următoarei operații de actualizare

CH(*grafic*:NP=Vulpe,PP=Dr,CS=281,DT=18-06,OD=5:15|

NP=Vulpe, PP=Dr,CS=83,DT=18-06,OD=5:15) ce trebuie să țină cont de restricția dată ?

- Schimbă numai numărul cursei în ultimul tuplu
  - Șterge tuplul 4
  - Este inactivă și determină un trigger de semnalizare
  - Se schimbă fără să țină cont de restricție
  - Trebuie modificată data
9. Se consideră relația persoană de schemă  $R = \{COD, NUME, PRENUME, LOCALITATE, AVERE\}$  notată pe scurt cu  $R = \{C, N, P, L, A\}$ , unde mulțimile  $\{C\}$  și  $\{N, P, L\}$  sunt chei. Specificați care restricție nu este dependență funcțională.
- $\{C\} \rightarrow \{N, P\}$
  - $\{C\} \rightarrow \{A\}$
  - $\{N, P, L\} \rightarrow \{A\}$
  - $\{A\} \rightarrow \{N, P, L\}$
  - $\{C\} \rightarrow \{N, P, L\}$
10. Se consideră relația persoană de schemă  $R = \{COD, NUME, PRENUME, LOCALITATE, AVERE\}$  notată pe scurt cu  $R = \{C, N, P, L, A\}$ , unde mulțimile  $\{C\}$  și  $\{N, P, L\}$  sunt chei. Specificați care dependență funcțională este derivată.
- $\{C\} \rightarrow \{N, P\}$
  - $\{C\} \rightarrow \{A\}$
  - $\{N, P, L\} \rightarrow \{1100\}$
  - $\{N, P, L\} \rightarrow \{A\}$
  - $\{C\} \rightarrow \{N, P, A\}$
11. Fie relația *grafic* de schemă  $R = \{NOME-PILOT, PRENUME-PILOT, CURSA, DATA, ORA-DEC\}$  notată pe scurt cu  $R = \{NP, PP, CS, DT, OD\}$  dată de tabelul

| NP    | PP  | CS  | DT    | OD    |
|-------|-----|-----|-------|-------|
| Ursu  | Ion | 83  | 9-06  | 10:15 |
| Ursu  | Ion | 116 | 10-06 | 11:15 |
| Jder  | Gh  | 83  | 13-06 | 10:15 |
| Jder  | Gh  | 116 | 9-06  | 11:15 |
| Lupu  | Cr  | 281 | 8-06  | 5:15  |
| Lupu  | Cr  | 83  | 11-06 | 10:15 |
| Lupu  | Cr  | 301 | 12-06 | 18:35 |
| Vulpe | Dr  | 281 | 18-06 | 5:15  |

Care mulțime de restricții (dependențe funcționale) este redundantă ?

- $F = \{CS \rightarrow OD, (NP, PP, DT) \rightarrow CS, (NP, PP, DT) \rightarrow OD\}$
  - $F = \{CS \rightarrow OD, (CS, DT) \rightarrow (NP, PP)\}$
  - $F = \{(CS, DT) \rightarrow (NP, PP, OD)\}$
  - $F = \{(NP, PP, DT) \rightarrow CS, CS \rightarrow OD\}$
  - $F = \{CS \rightarrow OD, (DT, OD) \rightarrow (NP, PP)\}$
12. Fie relația *grafic* de schemă  $R = \{NOME-PILOT, PRENUME-PILOT, CURSA, DATA, ORA-DEC\}$  notată pe scurt cu  $R = \{NP, PP, CS, DT, OD\}$  dată de tabelul

| NP    | PP  | CS  | DT    | OD    |
|-------|-----|-----|-------|-------|
| Ursu  | Ion | 83  | 9-06  | 10:15 |
| Ursu  | Ion | 116 | 10-06 | 11:15 |
| Jder  | Gh  | 83  | 13-06 | 10:15 |
| Jder  | Gh  | 116 | 9-06  | 11:15 |
| Lupu  | Cr  | 281 | 8-06  | 5:15  |
| Lupu  | Cr  | 83  | 11-06 | 10:15 |
| Lupu  | Cr  | 301 | 12-06 | 18:35 |
| Vulpe | Dr  | 281 | 18-06 | 5:15  |

Care mulțime de restricții (dependențe funcționale) este neredundantă?

- a)  $F = \{CS \rightarrow OD, (CS, DT) \rightarrow OD\}$   
b)  $F = \{(NP, PP, DT) \rightarrow CS, CS \rightarrow OD, (NP, PP, DT) \rightarrow OD\}$   
c)  $F = \{(CS, DT) \rightarrow (NP, PP), (CS, DT) \rightarrow (OD), (CS, DT) \rightarrow (OD, NP, PP)\}$   
d)  $F = \{CS \rightarrow OD, (NP, PP, DT) \rightarrow OD, (CS, DT) \rightarrow (NP, PP)\}$   
e)  $F = \{(CS, DT) \rightarrow (OD, NP, PP), (CS, DT) \rightarrow (NP, PP), CS \rightarrow NP\}$
13. Fie relațiile *persoana*, de schemă  $R1 = \{NUME, PRENUME, ADRESA, COD\}$  și *stare\_sanatate*, de schemă  $R2 = \{NUME, PRENUME, ADRESA, COD, NR\_CARNET\_SANATATE, MEDIC\_FAM\}$ , notate pe scurt cu  $R1 = \{N, P, A, C\}$  respectiv  $R2 = \{N, P, A, C, NS, MF\}$ , unde  $\{C\}$  și  $\{NS\}$  sunt chei. Care schemă, normalizată, de bază de date determină o redundanță minimă ?
- a)  $R = \{(N, P, A, C), (N, P, A, C, NS, MF)\}$   
b)  $R = \{(N, P, A), (N, P, A, C, NS, MF)\}$   
c)  $R = \{(N, P, A, C), (C, NS), (NS, MF)\}$   
d)  $R = \{(N, P, A, C), (C, NS, MF)\}$   
e)  $R = \{(N, P, A, C), (A, C, NS, MF)\}$
14. Fie relațiile *persoana*, de schemă  $R1 = \{NUME, PRENUME, ADRESA, COD\}$  și *stare\_sanatate*, de schemă  $R2 = \{NUME, PRENUME, ADRESA, COD, NR\_CARNET\_SANATATE, MEDIC\_FAM\}$ , notate pe scurt cu  $R1 = \{N, P, A, C\}$  respectiv  $R2 = \{N, P, A, C, NS, MF\}$ , unde  $\{C\}$  și  $\{NS\}$  sunt chei. Care este rezultatul următoarei operații: *persoana*  $\triangleright$  *stare\_sanatate* ?
- a) Conservă toate restricțiile  
b) Nu elimină complet redundanța  
c) Specifică carnetele de sănătate care nu au medic de familie  
d) Elimină redundanța  
e) Specifică persoanele care nu au carnet de sănătate
15. Fie schemele de relații  $R, R_1 \subset R, R_2 \subset R$  și relațiile  $r(R), r_1 = \pi_{R_1}(r), r_2 = \pi_{R_2}(r)$  determinate prin proiecție. Notăm cu  $t, t_1, t_2, t'$  timpul de actualizare al lui  $r$  respectiv  $r_1, r_2, r_1 \triangleright \triangleleft r_2$ . Care din următoarele condiții este satisfăcută ?
- a)  $t_1 + t_2 < t/3$   
b)  $t_1 + t_2 < \sqrt{t'}/2$   
c)  $t_1 + t_2 > 2t$   
d)  $t_1 + t_2 = t'$   
e)  $t_1 + t_2 < t'$
16. Fie relația *student* cu schema  $R = \{Nume, Incadrare, Situatia_scolara, An, Sex\}$  unde atributul *Incadrare* are valorile 'ZI' sau 'FF', *Situatia\_scolara* are valorile 'B' (bursier), 'N' (fără bursă), *An* are valorile 1, 2, ... iar *Sex* are valorile 'F' sau 'M'. Fraza

```
SELECT Nume
FROM student
WHERE (Situatia_scolara='B' OR An=5) AND Sex='F'
```

afișează:

- a) Numele studentelor bursiere și a celor din anul 5
- b) Numele studentelor bursiere din anul 5
- c) Numele tuturor studentelor bursiere
- d) Numele studentelor din anul 5
- e) Numele unei singure studente bursiere din anul 5

17. Fie relația *student* cu schema  $R=\{\text{Nume, Incadrare, Situatia\_scolară, An, Sex}\}$  unde atributul *Incadrare* are valorile 'ZI' sau 'FF', *Situatia\_scolară* are valorile 'B' (bursier), 'N' (fără bursă), *An* are valorile 1, 2, ... iar *Sex* are valorile 'F' sau 'M'. Fraza

```
SELECT Nume
FROM student
WHERE Situatia_scolara='B' OR An=1 AND Sex='F'
```

afișează:

- a) Numele studentelor bursiere din anul 1
- b) Numele studenților bursieri și a studentelor din anul 1
- c) Numele tuturor studenților bursieri
- d) Numele tuturor studentelor din anul 1
- e) Numele studentelor bursiere și a celor din anul 1

18. Fie relația *student* cu schema  $R=\{\text{Nume, Incadrare, Situatia\_scolară, An, Sex}\}$  unde atributul *Incadrare* are valorile 'ZI' sau 'FF', *Situatia\_scolară* are valorile 'B' (bursier), 'N' (fără bursă), *An* are valorile 1, 2, ... iar *Sex* are valorile 'F' sau 'M' și relația *note* cu schema  $R2=\{\text{Nume\_profesor, Nume\_student, Nota}\}$ . Fraza

```
SELECT DISTINCT Nume_profesor
FROM note
WHERE Nume_student IN
(SELECT Nume FROM student WHERE Situatia_scolara='B')
```

afișează:

- a) Numele profesorilor care au un singur student bursier
- b) Numele tuturor studenților bursieri și al profesorilor lor
- c) Numele profesorilor care au cel puțin un student bursier
- d) Numele profesorilor la care toți studenții sunt bursieri
- e) Numele unui singur student bursier și al profesorului său

19. Fie relația *note* cu schema  $R=\{\text{Nume\_profesor, Nume\_student, Nota}\}$ . Fraza

```
SELECT Nume_student
FROM note
WHERE Nota > ANY (SELECT Nota FROM note)
```

afișează:

- a) Numele studenților care au o notă mai mică decât cea mai mare notă din relație
- b) Numele studenților care au cea mai mică notă din relație
- c) Numele studenților care au o notă mai mare decât cea mai mică notă din relație
- d) Numele studenților care au cea mai mare notă din relație
- e) Numele studenților care au toate notele mai mari decât cea mai mică notă din relație

20. Fie relația *note* cu schema  $R=\{\text{Nume\_profesor, Nume\_student, Nota}\}$ . Fraza

```
SELECT DISTINCT Nume_student
FROM note xnote
WHERE Nume_student IN
 (SELECT nume_student FROM note
 WHERE Nume_profesor<>xnote.Nume_profesor)
```

afișează:

- a) Numele studenților care au note de la cel puțin un profesor
- b) Numele studenților care au note de la doi profesori
- c) Numele studenților și al profesorilor care le-au acordat note
- d) Numele studenților care au note de la doi profesori diferiți
- e) Numele studenților care au note de la cel puțin doi profesori

| Test | Raspuns |
|------|---------|
| 1    | a       |
| 2    | b       |
| 3    | a       |
| 4    | d       |
| 5    | e       |
| 6    | a       |
| 7    | b       |
| 8    | c       |
| 9    | d       |
| 10   | e       |
| 11   | a       |
| 12   | d       |
| 13   | c       |
| 14   | d       |
| 15   | e       |
| 16   | a       |
| 17   | b       |
| 18   | c       |
| 19   | e       |
| 20   | e       |

| Test | Raspuns |
|------|---------|
|------|---------|

## 11 Elemente de inteligență artificială

1. Considerați programul  $\Pi$ :

$$p(x) \vee q(g(x)) : \neg r(x) \\ r(a).$$

Care din următoarele afirmații sunt adevărate:

- $\Pi$  este un program Horn
- $\Pi$  nu este valid
- $f(g(a))$  este un termen ground peste baza programului
- $\Pi$  este un program normal
- variabila  $x$  din prima clauză este liberă

2. Considerați programul  $\Pi$ :

$$p(f(x)) \vee q(x) : \neg a(x) \\ a(b). \\ a(c).$$

Care din următoarele afirmații sunt adevărate:

- universul Herbrand este  $\{a, b, c, f(a), f(b), f(c), f(f(a)), f(f(b)), f(f(c)), \dots\}$
- dacă  $\Sigma = (D, k), D = \{2, 3\}, k(a)(2) = k(a)(3) = false, k(f)(x) = x$ , atunci prima formulă din  $\Pi$  este din  $sat_{\Sigma}$
- $f(f(a))$  este o formulă ground peste baza programului
- $\Pi$  este un program normal
- variabila  $x$  din prima clauză este liberă

3. Considerați programul  $\Pi$ :

$$p(f(x)) \vee q(x) : \neg a(x) \\ a(b). \\ a(c).$$

Care din următoarele afirmații sunt adevărate:

- universul Herbrand este  $\{a, b, c, f(a), f(b), f(c), f(f(a)), f(f(b)), f(f(c)), \dots\}$
- dacă  $\Sigma = (D, k), D = \{2, 3\}, k(a)(2) = k(a)(3) = false, k(f)(x) = x - 1$ , atunci prima formulă din  $\Pi$  este din  $sat_{\Sigma}$
- $f(f(a))$  este o formulă ground peste baza programului
- $\Pi$  este un program normal
- $\Pi \not\subseteq vd$

4. Considerați programul  $\Pi$ :

$$p(x) \vee q(x) : \neg a(x) \\ a(b). \\ a(c).$$

Care din următoarele afirmații sunt adevărate:

- baza Herbrand este  $\{p(a), p(b), p(c), q(b), q(c)\}$
- dacă  $\Sigma = (D, k), D = \{b, c\}, k(a)(b) = true, k(a)(c) = false, k(p)(x) = k(q)(x) = false$ , pentru  $x \in \{b, c\}$ , atunci prima formulă din  $\Pi$  este din  $sat_{\Sigma}$
- $p(f(b))$  este un termen ground peste baza programului
- $\Pi$  este un program normal
- $a(b) \wedge a(c) \in sat$

5. Considerați programul  $\Pi$ :

$$\begin{aligned} p(x) \vee q(f(x), g(y)) &: \neg a(x, y) \\ a(b, b). \\ a(c, c). \end{aligned}$$

Care din următoarele afirmații sunt adevărate:

- baza Herbrand este  $\{p(b), p(c), q(f(b), g(b)), q(f(c), g(c))\}$
- dacă  $\Sigma = (D, k)$ ,  $D = \{b, c\}$ ,  $k(a)(x, y) = true$ ,  $k(f)(x) = k(g)(x) = x$ ,  $k(p)(x) = true$ ,  $k(q)(x) = false$ , pentru orice  $x \in D$ , atunci prima formulă din  $\Pi$  este din  $sat_\Sigma$
- $p(a(b, b))$  este o formulă ground peste baza programului
- $\Pi$  este un program Herbrand general
- $a(b, b) \wedge a(c, c) \notin sat$

6. Semantica de punct fix pentru programul:

$$\begin{aligned} p(a, a) &\leftarrow \\ p(c, b) &\leftarrow \\ p(x, z) &\leftarrow p(x, y), p(y, z) \\ p(x, y) &\leftarrow p(y, x) \end{aligned}$$

este:

- $\{p(c, b), p(b, c), p(a, a), p(c, c), p(b, b)\}$
- $\{p(a, a), p(c, b), p(b, a), p(a, c), p(c, a)\}$
- $\{p(c, b), p(b, a), p(b, c), p(a, a), p(a, c)\}$
- $\{p(a, b), p(c, b), p(a, a), p(a, c), p(b, b)\}$
- $\{p(b, a), p(a, a), p(a, c), p(c, a), p(b, b)\}$

7. Semantica de punct fix pentru programul:

$$\begin{aligned} p(a, b) &\leftarrow \\ p(b, c) &\leftarrow \\ p(x, z) &\leftarrow p(x, a), p(a, z) \\ p(x, y) &\leftarrow p(y, x) \end{aligned}$$

este:

- $\{p(a, b), p(b, c), p(a, a), p(c, c), p(b, b)\}$
- $\{p(a, b), p(b, a), p(b, c), p(a, c), p(c, a)\}$
- $\{p(a, b), p(c, b), p(b, a), p(b, c), p(c, c)\}$
- $\{p(a, b), p(c, b), p(b, a), p(b, c), p(a, a)\}$
- $\{p(a, b), p(b, c), p(b, a), p(c, b), p(b, b)\}$

8. Semantica de punct fix pentru programul:

$$\begin{aligned} p(a, b) &\leftarrow \\ p(x, z) &\leftarrow p(x, b), p(b, z) \\ p(x, y) &\leftarrow p(y, x) \end{aligned}$$

este:

- $\{p(a, b), p(b, b), p(a, a)\}$
- $\{p(a, b), p(b, a), p(a, a)\}$
- $\{p(a, b), p(b, b), p(b, a)\}$
- $\{p(b, b), p(b, a), p(a, a)\}$
- $\{p(a, b), p(b, a), p(b, b)\}$

9. Semantica de punct fix pentru programul:

$$\begin{aligned} p(c, b) &\leftarrow \\ p(a, c) &\leftarrow \\ p(x, z) &\leftarrow p(x, y), p(y, z) \\ p(x, a) &\leftarrow p(a, x), p(x, b) \end{aligned}$$

este:

- a)  $\{p(c, b), p(b, a), p(b, c), p(a, a), p(a, c), p(b, b)\}$
- b)  $\{p(a, b), p(b, a), p(b, c), p(a, c), p(c, a), p(c, c)\}$
- c)  $\{p(a, b), p(c, b), p(b, a), p(b, c), p(a, c), p(c, a)\}$
- d)  $\{p(a, c), p(c, b), p(c, a), p(a, b), p(c, c), p(a, a)\}$
- e)  $\{p(a, b), p(b, a), p(a, a), p(a, c), p(c, a), p(c, c)\}$

10. Semantica de punct fix pentru programul:

$$\begin{aligned} p(c, c) &\leftarrow \\ p(a, c) &\leftarrow \\ p(x, b) &\leftarrow p(x, y), p(y, y) \end{aligned}$$

este:

- a)  $\{p(a, b), p(c, b), p(b, a), p(b, c)\}$
- b)  $\{p(a, b), p(a, c), p(c, c), p(b, b)\}$
- c)  $\{p(a, b), p(c, b), p(a, c), p(c, c)\}$
- d)  $\{p(c, b), p(a, c), p(c, c), p(b, b)\}$
- e)  $\{p(c, c), p(b, b), p(b, c), p(a, b)\}$

11. Semantica de *model stabil* pentru programul:

$$\begin{aligned} a &\leftarrow \neg b \\ b &\leftarrow \neg a \\ c &\leftarrow a, b \\ a &\leftarrow \neg c \end{aligned}$$

- a)  $(\{a\}, \{b, c\})$
- b)  $(\{a, b\}, \{c\})$
- c)  $(\{a, c\}, \{b\})$
- d)  $(\{b\}, \{a, c\})$
- e)  $(\{c\}, \{a, b\})$

12. Considerăm programul:

$$\begin{aligned} p(x) &: \neg q(x, y), \neg p(y) \\ q(a, b) &. \end{aligned}$$

Care din următoarele afirmații sunt adevărate ?

- a) Semantica de model stabil este  $\{q(a, b), p(a)\}$
- b) Există două modele stabile pentru acest program
- c) Nu există nici un model stabil pentru acest program
- d) Semantica de model stabil este  $\{q(a, b), p(b)\}$
- e) Nu se poate defini semantica de model stabil deoarece pentru aceste tipuri de programe se definesc alte semantici

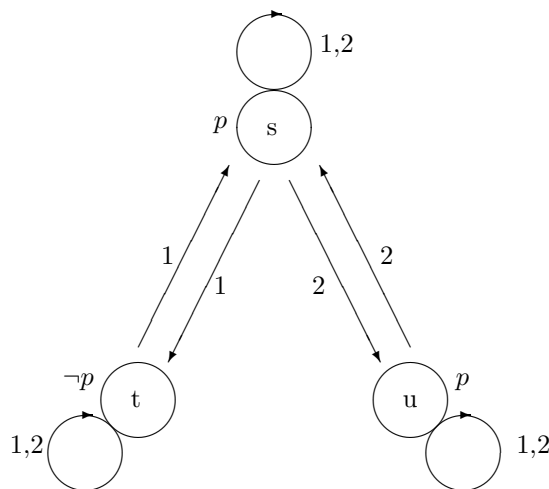
13. Considerați programul  $\Pi$ :

$$\begin{aligned} p(x) \vee q(g(x)) &: \neg r(x) \\ r(a) &. \end{aligned}$$

și  $\Sigma = (\{0, 1\}, k)$  unde  $k(a) = 1, k(g)(x) = 0$  dacă  $x = 1$  și  $k(g)(x) = 1$  dacă  $x = 0, k(p)(x) = k(q)(x) = k(r)(x) = false$  dacă  $x = 0$  și  $k(p)(x) = k(q)(x) = k(r)(x) = true$  dacă  $x = 1$ . Considerăm formulele  $\varphi = \forall x(p(x) \vee q(g(x)) \vee \neg r(x))$  și  $\psi = \forall xp(x) \vee \forall xq(g(x)) \vee \forall x\neg r(x)$ . Care din următoarele afirmații sunt adevărate ?

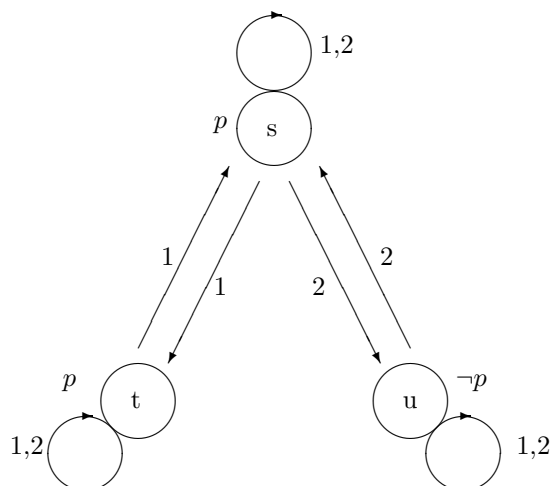


- a)  $truth\_val_{\Sigma}(\varphi, F) = false$  pentru orice  $F \in \{0, 1\}^{VS}$   
 b) există  $F \in \{0, 1\}^{VS}$  astfel încât  $truth\_val_{\Sigma}(\varphi, F) = false$   
 c)  $truth\_val_{\Sigma}(\psi, F) = true$  pentru orice  $F \in \{0, 1\}^{VS}$   
 d) există  $F \in \{0, 1\}^{VS}$  astfel încât  $truth\_val_{\Sigma}(\psi, F) = true$   
 e)  $truth\_val_{\Sigma}(\varphi, F) = \neg truth\_val_{\Sigma}(\psi, F)$  pentru orice  $F \in \{0, 1\}^{VS}$
14. Fie  $\Pi$  o mulțime de formule și  $\varphi$  o formulă (nu neapărat închise). Care din următoarele afirmații sunt adevărate ?
- a)  $\Pi \models \varphi$  dacă există un model pentru  $\Pi$  care face  $\varphi$  adevărată  
 b) dacă  $\Pi \models \varphi$  atunci  $\Pi \cup \{\neg\varphi\}$  este inconsistentă  
 c) dacă  $\Pi \cup \{\neg\varphi\}$  este inconsistentă atunci  $\Pi \models \varphi$   
 d) o structură  $\Sigma$  este un model pentru  $\Pi$  dacă există o formulă din  $\Pi$  care este validă în raport cu  $\Sigma$   
 e)  $\varphi \in \Pi$  este validă dacă și numai dacă  $\varphi$  este satisfiabilă
15. Fie  $p \in PS^{(1)}$  și  $c \in CS$ . Fie formulele  $\varphi_1 = ((\forall x)p(x)) \rightarrow p(c)$  și  $\varphi_2 = (\exists x)(p(x) \rightarrow (\forall y)p(y))$ . Care din afirmațiile de mai jos sunt adevărate ?
- a)  $\{\varphi_1, \varphi_2\} \subseteq vd$   
 b)  $\varphi_1 \in vd, \varphi_2 \notin vd$   
 c)  $\varphi_2 \in vd, \varphi_1 \notin vd$   
 d)  $\varphi_1 \notin vd, \varphi_2 \notin vd$   
 e)  $\varphi_1$  și  $\varphi_2$  sunt în formă clauzală
16. Notăm cu  $p$  propoziția: *Este soare la Craiova*. Considerați structura Kripke cu următorul graf atașat:



Care din următoarele afirmații este adevărată ?

- a) În starea  $s$  agentul 2 nu cunoaște că agentul 1 nu cunoaște că este soare la Craiova  
 b) În starea  $s$  agentul 1 cunoaște că este soare la Craiova  
 c) În starea  $t$  agentul 1 știe că este soare la Craiova  
 d) În starea  $u$  agentul 2 nu știe că este soare la Craiova  
 e) În starea  $s$  agentul 2 cunoaște că agentul 1 nu cunoaște că este soare la Craiova
17. Notăm cu  $p$  propoziția: *Este soare la Craiova*. Considerați structura Kripke cu următorul graf atașat:



Care din următoarele afirmații este adevărată ?

- În starea  $s$  agentul 2 cunoaște că agentul 1 nu cunoaște că este soare la Craiova
- În starea  $s$  agentul 1 cunoaște că nu este soare la Craiova
- În starea  $t$  agentul 1 nu știe că este soare la Craiova
- În starea  $u$  agentul 2 nu știe că agentul 1 știe ca este soare la Craiova
- În starea  $u$  agentul 2 cunoaște că agentul 1 știe că este soare la Craiova

18. Fie programul  $\Pi$  de mai jos:

$$\begin{aligned} p(f(x)) &\leftarrow q(x) \\ p(x) \vee q(x) &\leftarrow \\ p(a) &\leftarrow \\ q(b) &\leftarrow \end{aligned}$$

Care afirmație din cele de mai jos este adevărată ?

- Universul Herbrand este  $\{a, b, p(a), q(b), p(f(b))\}$
- Baza Herbrand este  $\{a, b, p(a), p(b), q(a), q(b), p(f(a)), p(f(b)), \dots\}$
- $\{p(a), q(b)\} \cup \{p(f^n(a)), p(f^n(b))\}_{n \geq 1}$  este model Herbrand pentru  $\Pi$
- $\Sigma = (D, k)$ , unde  $D = \{0, 1\}$ ,  $k(a) = k(b) = 1$ ,  $k(f)(x) = 1 - x$ ,  $k(p)(0) = true$ ,  $k(p)(1) = false$ ,  $k(q) = \neg k(p)$  este un model pentru  $\Pi$
- $\Pi$  este un program Herbrand general

19. Programul:

$$\begin{aligned} \text{ap}([], L, L). \\ \text{ap}([X|L1], L2, [X|L3]) :- \text{ap}(L1, L2, L3). \end{aligned}$$

în SWI-Prolog dă următorul răspuns:

- pentru  $\text{ap}([1, 2, 3], [3], [1, 3])$  răspunde **No**
- pentru  $\text{ap}([1, 2], [3, 4], [3, 4, 1, 2])$  răspunde **Yes**
- pentru  $\text{ap}([1, 2], X, [1, 2, 3, 4])$  răspunde **No**
- pentru  $\text{ap}([1, 2, 3], [2], [1, 3])$  răspunde **Yes**
- pentru  $\text{ap}([1, 2, 1], [3], [1, 2, 3])$  răspunde **Yes**

20. Ce regulă trebuie adăugată la următorul program astfel încât acesta să poată realiza căutarea unui număr într-un arbore binar de sortare:

```
cautare(f(A,_,_),A):-write_ln('Numar gasit').
cautare(f(A,St,Dr),Numar):-A>Numar,cautare(St,Numar).
```

- a) `cautare(f(A,St,Dr),Numar):-A<Numar,cautare(Dr,Numar).`
- b) `cautare([],_).`
- c) `cautare(f(A,St,Dr),Numar):-A=Numar,cautare(Dr,Numar).`
- d) `cautare(f(A,St,Dr),Numar):-A<Numar,cautare(f(A,St,Dr),Numar).`
- e) `cautare(f(A,St,Dr),Numar):-A<Numar,cautare(St,Numar).`

| Test | Raspuns |
|------|---------|
| 1    | b       |
| 2    | b       |
| 3    | e       |
| 4    | e       |
| 5    | b       |
| 6    | a       |
| 7    | e       |
| 8    | b       |
| 9    | d       |
| 10   | c       |
| 11   | a       |
| 12   | a       |
| 13   | e       |
| 14   | b       |
| 15   | a       |
| 16   | a       |
| 17   | d       |
| 18   | c       |
| 19   | a       |
| 20   | a       |

| Test | Raspuns |
|------|---------|
|------|---------|