

Varianta 1 online

1. Variabila n reprezintă un număr natural cu exact două cifre. Precizați câte dintre expresiile următoare au valoarea **1/true** dacă și numai dacă, cifrele lui n au aceeași paritate.

Limbajul C++/ Limbajul C

1. $(n/10-n\%10)\%2==0$
2. $n/10\%2==n\%2$
3. $n/10==n\%10$
4. $(n/10+n\%10*10)\%2==n\%2$
5. $n/2==n\%2$

Limbajul Pascal

1. $(n \text{ div } 10 - n \text{ mod } 10) \text{ mod } 2 = 0$
2. $n \text{ div } 10 \text{ mod } 2 = n \text{ mod } 2$
3. $n \text{ div } 10 = n \text{ mod } 10$
4. $(n \text{ div } 10 + n \text{ mod } 10 * 10) \text{ mod } 2 = n \text{ mod } 2$
5. $n \text{ div } 2 = n \text{ mod } 2$

- a) 0 b) 1 c) 2 d) 3 e) 4 f) 5

2. Pentru implementarea ecuației unei drepte de forma $ax+by+c=0$ (unde $a, b, c \in \mathbb{R}$), se definește structura:

Limbajul C++/ Limbajul C

```
typedef struct
    {float a, b, c;}dreapta;
```

Limbajul Pascal

```
type dreapta=record
    a, b, c : real;
end;
```

Dacă $d1$ și $d2$ sunt două variabile de tipul dreapta, precizați care dintre următoarele expresii verifică dacă $d1$ și $d2$ sunt paralele.

Limbajul C++/ Limbajul C

- a) $d1 \parallel d2$
- b) $d1.a==d2.a \ \&\& \ d1.b==d2.b$
- c) $a.d1/a.d2==b.d1/b.d2$
- d) $d1.a/d2.a==d1.b/d2.b$
- e) $d1.a==0 \ \&\& \ d2.a==0$
- f) $d1.a*d2.b-d1.b*d2.a==0$

Limbajul Pascal

- a) $d1 = d2$
- b) $(d1.a=d2.a) \text{ and } (d1.b = d2.b)$
- c) $a.d1/a.d2 = b.d1/b.d2$
- d) $d1.a/d2.a=d1.b/d2.b$
- e) $(d1.a=0) \text{ and } (d2.a=0)$
- f) $d1.a*d2.b-d1.b*d2.a=0$

3. Funcția f primește ca parametri două valori reale și returnează cea mai mare dintre cele două valori. Antetul funcției este:

(Limbajul C++/C) `float f(float x, float y);`

(Limbajul Pascal) `function f(x, y: real):real;`

Precizați care dintre următoarele expresii reprezintă suma celor mai mici două valori dintre numerele reale a, b și c .

- | | |
|----------------------------|-----------------------------|
| a) $a+b+c-f(a,b)$ | b) $a+b+c-f(a,b)-f(b,c)$ |
| c) $a+2*b+c-f(a,b)-f(b,c)$ | d) $a+b+c-f(a,b,c)$ |
| e) $a+b+c-f(a,f(c,b))$ | f) $a+b+c-f(f(a,b),f(b,a))$ |

4. Precizați care sunt valorile afișate în urma execuției următorului program.

Limbajul C++

```
#include<iostream>
using namespace std;
int a,b;
void f(int a, int &b)
{
    if(a>0)
    {
        a++; b--; f(b,a);
    }
}
```

Limbajul C

```
#include<stdio.h>
int a,b;
void f(int a, int *b)
{
    if(a>0)
    {
        a++; (*b)--;
        f(*b,&a);
    }
}
```

Limbajul Pascal

```
var a,b:integer;
procedure f(a:integer;
var b:integer);
begin
    if a>0 then
    begin
        a:=a+1; b:=b-1;
        f(b,a)
    end;
```

<pre> cout<<a<<" "<<b<<" "; } int main() { a=0; b=1; f(b,a); cout<<a<<" "<<b; } </pre>	<pre> printf("%d %d ",a,*b); } void main () { a=0; b=1; f(b,&a); printf("%d %d",a,b); } </pre>	<pre> write(a,' ',b,' ') end; begin a:=0; b:=1; f(b,a); write(a,' ',b,' ') end. </pre>
a) -1 2 2 -1 -1 1	b) 0 1 0 1	c) Ciclare infinită
d) -1 2 2 -1 0 1	e) 0 2 0 -1 0 1	f) -1 0 1 -1 1 1

5. În secvența următoare variabilele **n** și **m** au ca valori numere naturale.

Limbajul C++/ Limbajul C

```

n=42015; m=0;
while (n>0)
{
m=m*100+n/10%10*10+n%10;
n/=100;
}

```

Limbajul Pascal

```

n:=42015; m:=0;
while n>0 do
begin
m:=m*100+n div 10 mod 10*10+n mod 10;
n:=n div 100;
end;

```

După rularea secvenței, valoarea variabilei **m** este:

a) 15024 b) 15204 c) 24051 d) 51024 e) 152004 f) 152400

6. Se consideră următorul program:

Limbajul C++

```

#include<iostream>
using namespace std;
int main( )
{
int n, cn, x=0,p=1;
cin>>n;
cn=n;
while (n)
{
if (n%10>x)x=n%10;
n/=10;
}
x++;
while (cn)
{
n=n+cn%10*p;
p*=x;
cn/=10;
}
cout<<n;
}

```

Limbajul C

```

#include<stdio.h>
void main ( )
{
int n, cn, x=0,p=1;
scanf("%d", &n);
cn=n;
while (n)
{
if (n%10>x)x=n%10;
n/=10;
}
x++;
while (cn)
{
n=n+cn%10*p;
p*=x;
cn/=10;
}
printf("%d", n);
}

```

Limbajul Pascal

```

var n,cn,x,p : longint;
begin
readln(n);
cn:=n;
x:=0;
p:=1;
while n>0 do
begin
if n mod 10>x then
x:=n mod 10;
n:=n div 10;
end;
x:=x+1;
while cn>0 do
begin
n:=n+cn mod 10*p;
p:=p*x;
cn:=cn div 10;
end;
write(n)
end.

```

Precizați care este cel mai mic număr natural format din 5 cifre distincte care poate fi citit ca dată de intrare astfel încât valoarea afișată să fie aceeași.

a) 10000 b) 10192 c) 10234 d) 10239 e) 10923 f) 12345

7. Tabloul bidimensional **b** (cu liniile și coloanele numerotate de la 1 la n) se obține din tabloul bidimensional **a** prin rotire cu 90° spre dreapta.

De exemplu, dacă **a** este: $\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{pmatrix}$ se obține tabloul bidimensional **b**: $\begin{pmatrix} 7 & 4 & 1 \\ 8 & 5 & 2 \\ 9 & 6 & 3 \end{pmatrix}$

Pentru obținerea unei transformări corecte, secvența:

Limbajul C++/ Limbajul C

```
for(i=1; i<=n; i++)
    for(j=1; j<=n; j++) .....
```

trebuie completată cu atribuirea:

Limbajul C++/ Limbajul C

- a) `b[i][j]=a[j][i]`
- b) `b[i][j]=a[j][n-i+1]`
- c) `b[i][j]=a[n-j+1][n-i+1]`
- d) `b[i][j]=a[n-i+1][n-j+1]`
- e) `b[i][j]=a[n-j+1][i]`
- f) `b[i][j]=a[n-i+1][j]`

Limbajul Pascal

```
for i:=1 to n do
    for j:=1 to n do .....
```

Limbajul Pascal

- a) `b[i][j]:=a[j][i]`
- b) `b[i,j]:=a[j,n-i+1]`
- c) `b[i,j]:=a[n-j+1,n-i+1]`
- d) `b[i,j]:=a[n-i+1,n-j+1]`
- e) `b[i,j]:=a[n-j+1,i]`
- f) `b[i][j]:=a[n-i+1][j]`

8. Variabila **x** este de tip întreg și reprezintă o cifră nenulă. Precizați care dintre expresiile următoare este echivalentă cu expresia:

(Limbajul C++/C)

`x == 7 || x == 5`

(Limbajul Pascal)

`(x=7) or (x=5)`

Limbajul C++/ Limbajul C

- a) `35%x==0`
- b) `x!=7&&x!=5`
- c) `x>4&&!(x%2==0||x%3==0)`
- d) `x%2!=0&&x%3!=0`
- e) `!(x!=7||x!=5)`
- f) `x>4&&!(x%2==0&&x%3==0)`

Limbajul Pascal

- a) `35 mod x = 0`
- b) `(x<>7) and (x<>5)`
- c) `(x>4) and not((x mod 2=0) or (x mod 3=0))`
- d) `(x mod 2<>0) and (x mod 3<>0)`
- e) `not((x<>7) or (x<>5))`
- f) `(x>4) and not((x mod 2=0) and (x mod 3=0))`

9. Tabloul unidimensional **a** conține **n** numere naturale, ordonate crescător. Se cere afișarea mesajului **DA** dacă în **a** există două elemente a căror diferență este egală cu **s** (număr natural) sau a mesajului **NU**, în caz contrar. Precizați condiția ce trebuie utilizată în locul punctelor de suspensie astfel încât secvența următoare să rezolve corect problema dată.

Limbajul C++/C

```
i = 1; j = 2;
while ( ..... )
{
    if (a[j]-a[i]<s) j++;
    else i++;
}
if (j <= n) cout<<"DA"; | printf("DA");
else cout<<"NU"; | printf("DA");
```

- a) `j<n`
- b) `j<=n&&a[j]-a[i]!=s`
- c) `j<=n&&a[j]-a[i]==s`
- d) `a[j]-a[i]!=s`
- e) `i<=j`
- f) `i<=n&&a[j]-a[i]==s`

Limbajul Pascal

```
i:= 1; j:= 2;
while ..... do
begin
    if a[j]-a[i]<s then inc(j)
    else inc(i);
end;
if j <=n then write('DA')
else write('NU');
```

- a) `j<n`
- b) `(j<=n) and (a[j]-a[i]<>s)`
- c) `(j<=n) and (a[j]-a[i]=s)`
- d) `a[j]-a[i]<>s`
- e) `i<=j`
- f) `(i<=n) and (a[j]-a[i]=s)`

10. Precizați care este rolul următorului subprogram.

Limbajul C++/ Limbajul C

```
void f(char s[],char t[],int k)
{
    char aux[255];
    strcpy(aux,s+k);
    s[k]=0;
    strcat(s,t);
    strcat(s,aux);
}
```

Limbajul Pascal

```
procedure f(var s:string;t:string;
k:byte);
var aux:string;
begin
    aux:=copy(s,k,255);
    delete(s,k,255);
    s:=concat(s,t);
    s:=concat(s,aux);
end;
```

- a) Șterge ultimele k caractere ale lui s și concatenează rezultatul cu șirul t
b) Concatenează șirul s cu rezultatul concatenării șirurilor s și t
c) Inserează șirul s în șirul t , începând cu poziția k
d) Concatenează șirurile s și t , obținând un șir de lungime k
e) Înlocuiește primele k caractere din s cu primele k caractere din t
f) Inserează șirul t în șirul s , începând cu poziția k
11. Se consideră un graf orientat cu 6 noduri, numerotate $1, 2, \dots, 6$. Arcele grafului sunt de forma $(x, 2 \cdot x)$ pentru orice $x \in \{1, 2, 3\}$ și de forma $(x, x-1)$ pentru orice $x \in \{2, 3, 4, 5, 6\}$. Care este numărul minim de arce ce trebuie adăugate astfel încât graful să fie tare conex?
a) 0 b) 1 c) 2 d) 3 e) 4 f) 5
12. Precizați care dintre tablourile următoare poate reprezenta vectorul gradelor unui graf neorientat conex.
a) $(3, 2, 1, 5, 1, 1)$ b) $(5, 1, 6, 4, 5, 3)$ c) $(1, 1, 1, 1, 2, 2)$
d) $(1, 1, 1, 1, 1, 6)$ e) $(2, 1, 3, 1, 0, 1)$ f) $(1, 3, 5, 2, 1, 2)$
13. Dacă un graf neorientat conex are n vârfuri și $3n+2$ muchii, precizați care este valoarea minimă pentru n .
a) 16 b) 8 c) 4 d) 2 e) 1 f) 0
14. Pentru un număr natural nenul n , se construiește un arbore cu rădăcină astfel: rădăcina este numerotată n și orice nod care este numerotat cu o valoare $x > 1$ are ca fii nodurile numerotate cu divizorii săi, mai puțin numărul însuși. Toate frunzele arborelui sunt numerotate cu 1. Precizați câte dintre numerele naturale din intervalul $[10, 20]$ pot fi alese ca rădăcină, astfel încât arborele asociat să aibă un număr maxim de frunze.
a) 1 b) 2 c) 3 d) 4 e) 5 f) 6
15. Un pulover norvegian este frumos dacă pentru a-l tricota se folosesc cel puțin 2 și cel mult 4 culori de lână. Precizați câte modalități de combinare a culorilor există pentru a tricota un pulover norvegian frumos, având la dispoziție 5 ghemuri de lână de culori diferite.
a) 5 b) 12 c) 24 d) 25 e) 48 f) 125

Varianta 2 online

1. Precizați care dintre următoarele expresii are valoarea **1/true** dacă și numai dacă numărul natural nenul memorat în variabila **x** nu este divizibil cu 6.

Limbajul C++/ Limbajul C

- a) $x/6=0$
- b) $x==6$
- c) $x\%6 = 0$
- d) $x/6>0$
- e) $x\%2+x\%3>0$
- f) $x>6$

Limbajul Pascal

- a) $x \text{ div } 6 = 0$
- b) $x=6$
- c) $x \text{ mod } 6 = 0$
- d) $x/6>0$
- e) $x \text{ mod } 2+x \text{ mod } 3>0$
- f) $x>6$

2. Precizați care dintre următoarele instrucțiuni este corectă dacă variabilele **x**, **y** și **z** au declarările de mai jos:

Limbajul C++/ Limbajul C

- ```
float x;
int y,z;
a) x = x*y*z;
b) x = z*y*x;
c) x = x*y*z;
d) x = x*z*x;
e) x = x%z;
f) y = z%x;
```

Limbajul Pascal

- ```
x : real;  
y,z:integer;  
a) x:= x*y mod z;  
b) x:= z mod y*x;  
c) x:= x mod y*z;  
d) x:= x*z mod x;  
e) x:=x mod z;  
f) y:=z mod x;
```

3. Precizați ce valoare se va afișa pe ecran în urma executării secvenței de program următoare, știind că **s** este o variabilă care memorează un șir de caractere, iar **i** este o variabilă de tip întreg.

Limbajul C++/ Limbajul C

```
strcpy(s,"admitere");  
for(i=0;i<strlen(s);i++)  
    if(strchr("politehnica",s[i]))  
        strcpy(s+i,s+i+1);  
cout<<s; | printf("%s",s);
```

- a) **dmt** b) **dm** c) **dmtr** d) **dmr** e) **mt** f) **mrt**

Limbajul Pascal

```
s:='admitere';  
for i:=1 to length(s) do  
    if pos(s[i],'politehnica')>0 then  
        delete(s,i,1);  
write(s);
```

4. Precizați care dintre următoarele afirmații este adevărată pentru **orice** graf neorientat **G** format din **100** de noduri și **100** de muchii.

- a) Graful G nu este conex
- b) Graful G este conex
- c) Graful G este complet
- d) Graful G conține cel puțin un ciclu
- e) Graful G nu are noduri izolate
- f) Graful G conține un lanț elementar de lungime 100

5. Pentru reprezentarea unui graf orientat **G** se utilizează matricea de adiacență. Precizați care este suma elementelor din această matrice dacă graful are **20** de noduri și **30** de arce.

- a) **60** b) **50** c) **40** d) **30** e) **20** f) **10**

6. Precizați care este lungimea maximă a unui lanț simplu (lanț format din muchii distincte)

într-un arbore cu 10 noduri în care fiecare nod are gradul un număr impar.

- a) 9 b) 8 c) 7 d) 6 e) 5 f) 4

7. Tabloul unidimensional \mathbf{v} conține n numere întregi numerotate de la 1 la n . Precizați care dintre următoarele secvențe determină înlocuirea primului element din tabloul unidimensional \mathbf{v} cu cea mai mică valoare care apare în acesta.

Limbajul C++/Limbajul C

- a) `for (i=1; i<n; i++)
 if (v[i]>v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- b) `for (i=n-1; i>=1; i--)
 if (v[i]>v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- c) `for (i=n-1; i>=1; i--)
 if (v[i]<v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- d) `for (i=1; i<=n-1; i++)
 if (v[i]<v[i+1])
 {
 a=v[i];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- e) `for (i=n-1; i>=1; i--)
 if (v[i]<v[i+1])
 {
 a=v[i+1];
 v[i]=v[i+1];
 v[i+1]=a;
 }`
- f) `for (i=1; i<=n-1; i++)
 if (v[i]<v[i+1])
 {
 a=v[i+1];
 v[i]=v[i+1];
 v[i+1]=a;
 }`

Limbajul Pascal

- a) `for i:=1 to n-1 do
 if v[i]>v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- b) `for i:=n-1 downto 1 do
 if v[i]>v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- c) `for i:=n-1 downto 1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- d) `for i:=1 to n-1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- e) `for i:=n-1 downto 1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i+1];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`
- f) `for i:=1 to n-1 do
 if v[i]<v[i+1] then
 begin
 a:=v[i+1];
 v[i]:=v[i+1];
 v[i+1]:=a;
 end;`

8. Precizați pentru care dintre următoarele tablouri unidimensionale se poate aplica algoritmul căutării binare cu scopul de a găsi în mod eficient, dacă există, numere care au cifra

unităților egală cu o valoare x , dată.

- a) (1, 21, 13, 23, 33, 17, 27) b) (1, 13, 17, 21, 23, 27, 33)
c) (1, 13, 33, 17, 21, 23, 27) d) (33, 27, 23, 21, 17, 13, 1)
e) (1, 13, 33, 21, 23, 27, 17) f) (33, 27, 23, 21, 13, 1, 17)

9. În secvența de mai jos, variabila a memorează un tablou bidimensional cu 4 linii și 4 coloane, numerotate de la 1 la 4, cu elementele întregi. Variabila s este întregă, iar i este de tip întreg. Precizați care dintre instrucțiunile de mai jos poate înlocui punctele de suspensie, astfel încât secvența să determine memorarea în variabila s , a valorii sumei elementelor aflate pe prima și ultima coloană ale matricei.

Limbajul C++/Limbajul C

```
s=0;
for (i=1;i<=4;i++)....
a) s=s+a[4][i]+a[i][4];
b) s=s+a[4-i][4]+a[i][1];
c) s=s+a[i][1]+a[i][4];
d) s=s+a[i][i]+a[1][i];
e) s=s+a[1][i]+a[4][i];
f) s=s+a[i][i]+a[5-i][i];
```

Limbajul Pascal

```
s:=0;
for i:=1 to 4 do .....
a) s:=s+a[4,i]+a[i,4];
b) s:=s+a[4-i,4]+a[i,1];
c) s:=s+a[i,1]+a[i,4];
d) s:=s+a[i,i]+a[1,i];
e) s:=s+a[1,i]+a[4,i];
f) s:=s+a[i,i]+a[5-i,i];
```

10. Utilizând metoda backtracking se generează toate anagramele cuvântului *avion*. Precizați câte anagrama încep și se termină cu câte o consoană.

- a) 6 b) 12 c) 20 d) 36 e) 38 f) 40

11. Subprogramul f are definiția următoare. Dacă variabilele a și b sunt de tip întreg și memorează valorile 3 respectiv 5, precizați care vor fi valorile pe care le memorează variabilele a și b după apelul:

$f(a, b)$; (Limbajul Pascal/C++)

$f(a, \&b)$; (Limbajul C).

Limbajul C++

```
void f(int x,int &y)
{int aux;
  aux=x; x=y;
  y=aux;
}
```

Limbajul C

```
void f(int x,int *y)
{int aux;
  aux=x; x=*y;
  *y=aux;
}
```

Limbajul Pascal

```
procedure f(x:integer;var y:integer);
  var aux:integer;
begin
  aux:=x; x:=y; y:=aux;
end;
```

- a) 3 și 3 b) 4 și 3 c) 5 și 5 d) 3 și 5 e) 3 și 4 f) 5 și 3

12. Considerăm declararea următoare, folosită pentru a memora numărătorul și numitorul unei fracții. Precizați care dintre instrucțiunile de mai jos este corectă.

Limbajul C++/Limbajul C

Limbajul Pascal

```
typedef struct
  { int a, b; }fractie;
fractie m,n;
```

- a) m=n;
- b) if (m>n) m++;
- c) if (m==n) m--;
- d) if (m<=n) m=n;
- e) if (m!=n) m--;
- f) if (m>n) m=n;

```
type fractie=record
  a,b:integer;
end;
var m,n:fractie;
```

- a) m:=n;
- b) if (m>n) then m:=m+1;
- c) if (m=n) then m:=m-1;
- d) if (m<=n) then m:=n;
- e) if (m<>n) then m:=m-1;
- f) if (m>n) then m:=n;

13. Precizați care este numărul de grafuri orientate distincte formate din **3** noduri și **4** arce. Două grafuri sunt distincte dacă au matricea de adiacență diferită.

- a) **32**
- b) **30**
- c) **20**
- d) **16**
- e) **15**
- f) **9**

14. Precizați care este instrucțiunea prin care variabilei **y** i se atribuie numărul obținut prin inversarea ordinii cifrelor numărului natural format din exact **2** cifre, memorat în variabila întregă **x**.

Limbajul C++/Limbajul C

- a) **y=x/10*10+x%100;**
- b) **y=x%10+x/10;**
- c) **y=x/10*10+x%10;**
- d) **y=x%100/10;**
- e) **y=x*10%100+x/10;**
- f) **y=x%10/10;**

Limbajul Pascal

- a) **y:=x div 10*10+x mod 100;**
- b) **y:=x mod 10 +x div 10;**
- c) **y:=x div 10*10+x mod 10;**
- d) **y:=x mod 100 div 10;**
- e) **y:=x*10 mod 100+x div 10;**
- f) **y:=x mod 10 div 10;**

15. Fie **G** un graf neorientat complet cu **100** de noduri. Precizați care dintre următoarele afirmații este adevărată:

- a) În graful **G** există un lanț elementar de lungime **100**
- b) Graful **G** este un graf hamiltonian
- c) Graful **G** este un graf eulerian
- d) Graful **G** nu este conex
- e) Graful **G** are **900** de muchii
- f) Graful **G** are două componente conexe

Varianta 3 online

1. Indicați câte dintre expresiile următoarele au valoarea 1 (**Limbajul C/C++**), respectiv **true** (**Limbajul Pascal**) dacă și numai dacă valorile variabilelor **a** și **b** sunt numere întregi pare consecutive.

Limbajul C/C++

1. `(a%2) && (b%2) && (a-b==2)`
2. `(a%2) && (a-b==2 | | b-a==2)`
3. `!(a%2) && abs(a-b)==2`
4. `!(a%2) && !(b%2) && abs(a-b)==2`
5. `!!(a%2) && (a-b==2)`
6. `(a%2==0) && !(abs(a-b)==2)`

a) 0 b) 1 c) 2

Limbajul Pascal

1. `(a mod 2<>0) and (b mod 2<>0) and (a-b=2)`
2. `(a mod 2<>0) and ((a-b=2) or (b-a=2))`
3. `not(a mod 2<>0) and (abs(a-b)=2)`
4. `not(a mod 2<>0) and not(b mod 2<>0) and (abs(a-b)=2)`
5. `not(not(a mod 2<>0)) and (a-b=2)`
6. `(a mod 2=0) and not(abs(a-b)=2)`

d) 3 e) 4 f) 5

2. Precizați ce se va afișa în urma rulării secvenței următoare, în care se consideră că variabilele **a** și **b** memorează numere reale.

Limbajul C++

```
a=5.2;
b=-3.25;
a-=b;
b*=2;
cout<<ceil(a+b)<<" "<<
floor(a-b);
```

Limbajul C

```
a=5.2;
b=-3.25;
a-=b;
b*=2;
printf("%g %g",ceil(a+b),floor(a-b));
```

a) -5 8 b) -4 8 c) 1 14 d) 1 15 e) 2 14 f) 2 15

Limbajul Pascal

```
a:=5.2;
b:=-3.25;
a:=a-b;
b:=b*2;
write(round(a+b),' ',trunc(a-b));
```

3. Precizați ce se afișează la sfârșitul executării secvenței următoare.

Limbajul C++

```
void p(int a, int &b)
{ a++;
  b=b*a;
  b-=10;}
int g(int a, int b)
```

Limbajul Pascal

```
var a,b:integer;
procedure p(a:integer; var b:integer);
begin
  inc(a);
```

```

{ a*=10;
  b+=a;
  a=b;
  return a;}
int main()
{ int a=2,b=7;
  p(a,b);
  cout<<g(b,a);}

```

Limbajul C

```

void p(int a, int *b)
{ a++;
  *b=*b*a;
  *b-=10;}
int g(int a, int b)
{ a*=10;
  b+=a;
  a=b;
  return a;}
int main()
{ int a=2,b=7;
  p(a,&b);
  printf("%d",g(b,a)); }

```

a) 27 b) 41 c) 72

```

  b:=b*a;
  b:=b-10;
end;

function
g(a,b:integer):integer;
begin
  a:=a*10;
  b:=b+a;
  a:=b;
  g:=a;
end;

begin
  a:=2;
  b:=7;
  p(a,b);
  write(g(b,a));
end.

```

d) 73 e) 112 f) 113

4. Se consideră subprogramul **f** definit mai jos. Precizați ce se afișează în urma apelului **f(8)**.

Limbajul C++

```

void f(int i)
{ if (i>1)
  if (i%2)
    {f(i-1);
     cout<<i-1<<" ";}
  else {i--;
        f(i);}
}

```

Limbajul C

```

void f(int i)
{ if (i>1)
  if (i%2)
    {f(i-1);
     printf("%d ",i-1);}
  else {i--;
        f(i);}
}

```

a) 2 3 5 b) 2 4 6 c) 3 5 7

Limbajul Pascal

```

procedure f(i:integer);
begin
  if i>1 then
    if i mod 2<>0 then
      begin
        f(i-1);
        write(i-1,' ');
      end
    else
      begin
        dec(i);
        f(i);
      end;
end;

```

d) 5 3 2 e) 6 4 2 f) 7 5 3

5. Precizați ce se va afișa în urma rulării secvenței date, în care se consideră că variabilele **x** și **y** sunt de tip întreg.

Limbajul C++

```
x=5;
y=2;
cout<<"+x/y+++1";
cout<<endl<<x<<" "<<y;
```

Limbajul Pascal

```
x:=5;
y:=2;
inc(x);
writeln(x div y+1);
inc(y);
writeln(x, ' ', y);
```

Limbajul C

```
x=5;
y=2;
printf("%d", ++x/y+++1);
printf("\n%d %d", x, y);
```

- a) 4 b) 4 c) 4 d) 3 e) 3 f) 3
 6 3 6 2 5 2 6 3 6 2 5 2

6. Știind că variabila **s** este de tip șir de caractere, precizați ce se va afișa după executarea următoarei secvențe de instrucțiuni.

Limbajul C/C++

```
strcpy(s, "ExamenUPB");
for (i=0; i<strlen(s)/2; i++)
    s[i]=s[strlen(s)-i-2];
strcpy(s, s+2);
strcpy(s+strlen(s)-2,
s+strlen(s)-1);
printf("%s", s); | cout<<s;
```

Limbajul Pascal

```
s:='ExamenUPB';
for i:=1 to length(s) div 2
do
s[i]:=s[length(s)-i];
delete(s, 1, 2);
delete(s, length(s)-1, 1);
write(s);
```

- a) UnenUB b) UnenUP c) neenUB d) neenUP e) nennUB f) nennUP

7. Indicați care este numărul de comparații executate pentru ordonarea descrescătoare a unui tablou unidimensional cu 50 elemente, prin metoda interschimbării.

- a) 25 b) 49 c) 50 d) 1225 e) 1226 f) 2450

8. Utilizând metoda backtracking se generează toate codurile formate din cinci caractere distincte ale mulțimii {**a, b, c, d, e, f**}. Primele cinci soluții generate sunt: **abcde, abcdf, abced, abcef, abcfcd**. Indicați care sunt codurile generate imediat în fața soluției **dcbae**, dar și imediat după aceasta.

- a) **dcaef**; b) **dcafe**; c) **dcbafe**; d) **dcbfef**; e) **dcbeff**; f) **dcbfaf**;
 dcafe **dcbafe** **dcbeaf** **dceab** **dcbfef** **dcbafe**

9. La o cantină se prepară zilnic 5 sortimente pentru felul întâi, 10 pentru felul doi și 6 tipuri de desert. Precizați câte posibilități de a alege un meniu există, știind că un meniu este alcătuit din felul întâi, felul doi și facultativ desert.

- a) 50 b) 80 c) 90 d) 250 e) 300 f) 350

10. Fie un graf neorientat cu **10** noduri. Gradele vârfurilor acestuia sunt reținute în șirul: **4, 2, 2, 3, 3, 3, 2, 4, 2, 3**. Precizați care este numărul de muchii ce trebuie adăugate pentru ca graful să devină complet.
 a) **45** b) **41** c) **31** d) **27** e) **26** f) **17**
11. Fie un graf neorientat complet cu **50** noduri. Precizați care este numărul minim de muchii care trebuie eliminate pentru ca graful să fie hamiltonian.
 a) **0** b) **25** c) **50** d) **612** e) **1175** f) **1225**
12. Precizați câți arbori binari cu **3** noduri, numerotate de la **1** la **3**, se pot construi. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii), ordonați: fiu stâng, fiu drept. Dacă un nod are un singur descendent trebuie specificat dacă este fiu stâng sau fiu drept.
 a) **6** b) **8** c) **9** d) **21** e) **24** f) **30**
13. Fie arborele cu rădăcină cu nodurile numerotate de la **1** la **15**, reprezentat prin vectorul de tați: **(10, 8, 4, 10, 1, 4, 5, 10, 8, 0, 3, 5, 3, 12, 3)**. Precizați câți descendenți are nodul **4**.
 a) **6** b) **5** c) **4** d) **3** e) **2** f) **1**
14. Precizați câte grafuri neorientate distincte cu **25** noduri, dintre care cel puțin un nod este izolat, se pot construi.
 a) 2^{276} b) 2^{300} c) $3 \cdot 2^{279}$ d) $5^2 \cdot 2^{276}$ e) $3 \cdot 2^{303}$ f) $5^2 \cdot 2^{300}$
15. Fie secvența de instrucțiuni folosită pentru ridicarea la puterea **p** a unei matrice pătratice **a**, de ordin **n**. Elementele tabloului **a** sunt numere întregi, iar **n** și **p** sunt numere naturale nenule. Variabilele **a**, **b**, **c** sunt tablouri bidimensionale, cu **n** linii și **n** coloane, iar variabilele **i**, **j**, **k** sunt de tip întreg.
- | | |
|--|---|
| <pre> Limbajul C/C++ for (i=1; i<=n; i++) for (j=1; j<=n; j++) b[i][j]=(i==j); for (q=1; q<=p; q++) { for (i=1; i<=n; i++) for (j=1; j<=n; j++) {c[i][j]=0; for (k=1; k<=n; k++) c[i][j]= c[i][j]+b[i][k]*a[k][j]; } } for (i=1; i<=n; i++) for (j=1; j<=n; j++) b[i][j]=c[i][j]; } </pre> | <pre> Limbajul Pascal for i:=1 to n do for j:=1 to n do if i=j then b[i,j]:=1 else b[i,j]:=0; for q:=1 to p do begin for i:=1 to n do for j:=1 to n do begin c[i,j]:=0; for k:=1 to n do c[i,j]:=c[i,j]+b[i,k]*a[k,j]; end; for i:=1 to n do for j:=1 to n do b[i,j]:=c[i,j]; end; end; </pre> |
|--|---|

| **end;**

Precizați de câte ori se execută operația de înmulțire în cadrul secvenței date pentru ridicarea la puterea **p** a matricei pătratice **a** de ordin **n**.

- a) $(n^3)^p$ b) $(p+1) \cdot n^3$ c) $p \cdot n^3$ d) $(p-1) \cdot n^3$ e) $p \cdot n^2$ f) n^3

Varianta 4 online

1. Indicați care expresie dintre următoarele are valoarea **1** (**Limbajul C/C++**), respectiv **true** (**Limbajul Pascal**) dacă și numai dacă valorile variabilelor **a** și **b** sunt numere întregi impare consecutive.

Limbajul C/C++

- `(a%2==1) &&! (b%2) && abs(a-b)==2`
- `!(a%2) &&! (b%2) && abs(a-b)==2`
- `!(!(a%2) &&! (b%2)) && abs(a-b)==2`
- `(a%2==1 && b%2!=1) && (a-b==2 || b-a==2)`
- `!(a%2) &&! (b%2) &&! (abs(a-b)==2)`
- `(a%2) &&(b%2) && abs(a-b)!=2`

a) 1 b) 2 c) 3

Limbajul Pascal

- `(a mod 2=1) and (not(b mod 2<>0)) and (abs(a-b)=2)`
- `not(a mod 2<>0) and not(b mod 2<>0) and (abs(a-b)=2)`
- `not(not(a mod 2<>0) and (not(b mod 2<>0))) and (abs(a-b)=2)`
- `((a mod 2=1) and (b mod 2<>1)) and ((a-b=2) or (b-a=2))`
- `not(a mod 2<>0) and not(b mod 2<>0) and not(abs(a-b)=2)`
- `(a mod 2<>0) and (b mod 2<>0) and (abs(a-b)<>2)`

d) 4 e) 5 f) 6

2. Precizați ce se va afișa în urma rulării secvenței următoare, dacă variabila întregă **x** are valoarea inițială **1234**.

Limbajul C++

```
x=x%100/10*10/10%10 +  
x/10%10;  
cout<<ceil(sqrt(x)+0.5)<<"  
<< floor(sqrt(x)-0.5);
```

Limbajul C

```
x=x%100/10*10/10%10 +  
x/10%10;  
printf("%g %g",  
ceil(sqrt(x)+0.5), floor(sqrt(x)-0.5));
```

a) 2 0 b) 2 1 c) 2 2

Limbajul Pascal

```
x:= x mod 100 div 10*10 div  
10 mod 10 +x div 10 mod 10;  
write(round(sqrt(x)+0.5), '  
, trunc(sqrt(x)-0.5));
```

d) 3 1 e) 3 2 f) 5 3

3. Precizați ce se afișează la sfârșitul executării secvenței următoare.

Limbajul C++

```
void p(int a, int &b)  
{ a+=a+b;  
  b+=a;  
  a=b-a; }
```

Limbajul Pascal

```
var a,b:integer;  
procedure p(a:integer; var  
b:integer);  
begin
```

```
int main()
{ int a=5,b=10;
  p(a,b);
  cout<<a<<" "<<b;
  p(a,b);
  cout<<endl<<a<<" "<<b;
}
```

Limbajul C

```
void p(int a, int *b)
{ a+=a>(*b);
  *b+=a;
  a=*b-a; }
```

```
int main()
{ int a=5,b=10;
  p(a,&b);
  printf("%d %d",a,b);
  p(a,&b);
  printf("\n%d %d",a,b);}
```

a) 5 30 b) 5 30 c) 5 30
5 70 5 80 10 80

```
  a:=a+a+b;
  b:=b+a;
  a:=b-a;
end;

begin
  a:=5;
  b:=10;
  p(a,b);
  writeln(a,' ',b);
  p(a,b);
  writeln(a,' ',b);
end.
```

d) 10 30 e) 10 30 f) 10 30
10 30 30 70 30 80

4. Precizați care este valoarea returnată de funcție la apelul **f(1502)**.

Limbajul C/C++

```
int f(int i)
{ if(i==0) return 10;
  else
    if(i%10==0 || i%10==5)
      return f(i/10)*10+i;
    else
      return i%10 * f(i/10);
}
```

a) 3175 b) 2600 c) 2330

Limbajul Pascal

```
function
f(i:integer):integer;
begin
  if i=0 then f:=10
  else
  if (i mod 10=0) or (i mod
5=0)
  then f:=f(i div 10)*10 + i
  else
  f:=(i mod 10)*f(i div 10);
end;
```

d) 2050 e) 530 f) 350

5. Precizați ce se va afișa în urma rulării secvenței următoare, în care se consideră că variabilele **x** și **y** sunt de tip întreg.

Limbajul C++

```
x=2; y=5;
while(x<y)
{ cout<<++x+y++<<" ";
  x++; }
```

Limbajul C

```
x=2; y=5;
```

Limbajul Pascal

```
x:=2; y:=5;
while x<y do
begin
  inc(x);
  write(x+y,' ');
  inc(y);
  inc(x);
```

```

while (x<y)
{ printf("%d ",++x+y++);
  x++; }

```

```
end;
```

- a) 7 10 13 b) 8 10 c) 8 11 14
d) 9 11 e) 9 12 f) 9 12 15

6. Știind că variabila **s** este de tip șir de caractere, precizați ce se va afișa după executarea următoarei secvențe de instrucțiuni.

Limbajul C/C++

```

strcpy(s, "Examen-UPB");
for(i=strlen(s)/2;i>0;i--)
    s[i]=s[strlen(s)-i];
strcpy(s+strlen(s)/2-1,
s+strlen(s)/2+1);
printf("%s",s); | cout<<s;

```

Limbajul Pascal

```

s:='Examen-UPB';
for i:=length(s) div 2 downto 2
do s[i]:=s[length(s)-i+2];
delete(s,length(s) div 2,2);
write(s);

```

- a) -UPB b) -n-UPB c) n-UPB
d) EBPUUPB e) EBPU-UPB f) EBPU--UPB

7. Indicați care este numărul necesar de comparații pentru ordonarea prin interschimbare a unui tablou unidimensional cu **100** elemente.

- a) 99 b) 2475 c) 4851 d) 4950 e) 5050 f) 10000

8. Indicați câte numere divizibile cu **10**, cu **10** cifre, pot fi construite folosind numai cifrele **0, 1 și 2**.

- a) 6561 b) 13122 c) 13212 d) 15322 e) 19683 f) 59049

9. Fie mulțimile **A={1, 2, 3, 4}**, **B={1, 2, 3}**, **C={1, 2}**, **D={1, 2, 3, 4}**. Precizați care este al **10**-lea element al produsului cartezian **A×B×C×D**, cât și antepenultimul element.

- a) 1212; 4322 b) 1212; 4323 c) 1213; 4322
d) 1221; 4322 e) 1312; 4322 f) 1312; 4323

10. Fie un graf neorientat cu **25** noduri și **40** muchii. Precizați care este numărul maxim de noduri izolate pe care le poate avea graful.

- a) 16 b) 15 c) 14 d) 13 e) 10 f) 5

11. Fie un graf neorientat cu **100** noduri. Precizați care este numărul minim de muchii necesar pentru ca graful să nu aibă noduri izolate.

- a) 48 b) 49 c) 50 d) 98 e) 99 f) 100

12. Precizați care este numărul maxim de frunze al unui arbore binar cu **100** noduri care are înălțimea minimă. Un arbore binar este un arbore în care fiecare nod are cel mult doi descendenți direcți (fii).

- a) 99 b) 69 c) 51 d) 50 e) 37 f) 34

13. Fie arborele cu rădăcină cu nodurile numerotate de la 1 la 15, reprezentat prin vectorul de tați: (10, 8, 4, 10, 1, 4, 5, 10, 8, 0, 3, 5, 3, 12, 3). Precizați câte lanțuri elementare distincte de lungime 3, care pleacă din rădăcină, există.
a) 2 b) 3 c) 4 d) 5 e) 6 f) 7
14. Precizați câte grafuri orientate distincte cu 25 noduri, dintre care cel puțin un nod este izolat, se pot construi.
a) $5^2 \cdot 2^{600}$ b) $3 \cdot 2^{603}$ c) 2^{600} d) $5^2 \cdot 2^{552}$ e) $3 \cdot 2^{555}$ f) 2^{552}
15. Precizați care este complexitatea timp pentru următoarea secvență de program, unde n reprezintă numărul de elemente al unui tablou unidimensional v , numerotat de la 1 la n , cu elemente numere întregi, iar x un număr întreg.

Limbajul C/C++

```
j=0;
for(i=1; i<=n; i++)
  if(v[i]!=x)
    { j++; v[j]=v[i]; }
n=j;
```

Limbajul Pascal

```
j:=0;
for i:=1 to n do
  if v[i]<>x then
    begin
      inc(j);v[j]:=v[i];
    end;
n:=j;
```

- a) $O(n)$
d) $O(n^2)$

- b) $O(\log n)$
e) $O(n^3)$

- c) $O(n \cdot \log n)$
f) $O(2^n)$

Varianta 5 online

1. Fie subprogramul:

Limbajul C++/C

```
int f (int n, int s){
    if (n < s)
        return 0;
    else
        if(n%s == 0)
            return 1+ f(n/s,s+1);
        else
            return f(n/s,s);
}
```

Limbajul Pascal

```
function f(n,s:integer):
integer;
begin
    if (n < s) then f:=0
    else
        if (n mod s =0) then
            f:= 1 + f(n div s,s+1)
        else
            f:= f(n div s,s);
end;
```

Subprogramul **f** se execută pentru următoarele seturi de valori **n=720, s=2; n=120, s=3; n=120, s=1; n=720, s=1**. Pentru câte dintre apeluri subprogramul **f** va returna valoarea 5?

- a) un apel b) 2 apeluri c) 3 apeluri d) nici un apel e) 4 apeluri f) 5 apeluri
2. Fie subprogramul de mai jos unde **n** și **c** sunt variabile de tip întreg.

Limbajul C++

```
int f(int &n, int c)
{
    int a=n%10;
    if(n==0)
        return 0;
    else
        if(a==c)
            {n=n/10;
            return 1+f(n,c);}
        else
            {n=n/10%10;
            return f(n,c);
        }
}
```

Limbajul Pascal

```
function f(var n:integer;
c:integer):integer;
var a:integer;
begin
    a:=n mod 10;
    if(n=0) then
        f:=0
    else
        if(a=c) then
            begin
                n:=n div 10;
                f:=1+f(n,c);
            end
        else
            begin
                n:=n div 10 mod 10;
                f:=f(n,c);
            end;
end;
```

Limbajul C

```
int f (int *n, int c)
{
    int a=*n %10;
    if(*n==0) return 0;
    else
        if(a==c)
            {*n=*n/10;
            return 1+f(n,c);}
        else
```

```

    { *n = (*n) / 10 % 10;
      return f(n, c);
    }
}

```

Variabilele ale căror valori sunt reținute în stiva subprogramului sunt:

a) **n, c, a** b) **c, a** c) **n, c** d) **a** e) **c** f) **n, a**

3. Fie **G** un graf neorientat cu **n > 0** vârfuri și **m > 0** muchii, reprezentat prin liste de adiacență. Complexitatea unui algoritm care afișează matricea de adiacență asociată grafului este:

a) $O(m \cdot \log n)$ b) $O(m \cdot n)$ c) $O(n^2)$ d) $O(m^2)$ e) $O(m^2 + 1)$ f) $O(m^2 - 1)$

4. Variabilele **x** și **y** sunt de tip întreg. Știind că **x > -1** și **y < 3**, dintre expresiile de mai jos cea care are valoarea **1** (**Limbajul C/C++**) respectiv **True** (**Limbajul Pascal**) este:

Limbajul C/C++

a) $x * y + y - 3 * x - 3 > 0$ b) $!(x * y + y - 3 * x - 3 >= 0)$ c) $(x - 1) * (y - 3) < 0$
d) $(x + 1) * (y - 3) > 0$ e) $(x - 1) * (y + 3) > 0$ f) $(x + 1) * (y + 3) > 0$

Limbajul Pascal

a) $x * y + y - 3 * x - 3 > 0$ b) $\text{NOT}(x * y + y - 3 * x - 3 >= 0)$ c) $(x - 1) * (y - 3) < 0$
d) $(x + 1) * (y - 3) > 0$ e) $(x - 1) * (y + 3) > 0$ f) $(x + 1) * (y + 3) > 0$

5. Se consideră numărul natural **n = 231045**. Dacă se determină toate submulțimile formate din cifrele lui **n** care au suma valorilor componentelor egală cu **10**, câte submulțimi conțin cifra **0**?

a) **3** b) **5** c) **2** d) **1** e) **4** f) **6**

6. Se consideră șirul **{a, b, c, u, i, e}**. Se generează folosind metoda backtracking, în ordine lexicografică, toate cuvintele de trei litere distincte, care conțin două vocale. Dacă primele trei soluții sunt **abe, abi, abu** care este a **9-a** soluție?

a) **aic** b) **aib** c) **aec** d) **aub** e) **ace** f) **aei**

7. În câte moduri se poate scrie numărul **12** ca sumă de numere prime?

a) **5** b) **3** c) **7** d) **6** e) **5** f) **4**

8. Se consideră mulțimea de cuvinte **{info, mate, fizica, chimie, biologie}**. Se generează folosind metoda backtracking, lexicografic, în ordinea inversă citirii cuvântului, submulțimi de câte trei cuvinte distincte. Dacă primele trei soluții sunt: **{fizica, biologie, chimie}**; **{fizica, biologie, mate}**; **{fizica, biologie, info}**; înaintea soluției **{chimie, mate, info}** este soluția:

a) **{biologie, mate, info}** b) **{biologie, chimie, mate}**

- c) {chimie,biologie,info}
e) {fizica,mate,biologie}

- d) {chimie,mate,biologie}
f) {chimie,fizica,biologie}

9. Se consideră un arbore cu rădăcină în care fiecare nod intern (nod care nu este pe ultimul nivel) are doi descendenți direcți. Dacă arborele are k niveluri (rădăcina se află pe nivelul 0) câte noduri sunt pe nivelul k ?
a) 2^{k+1} b) $2^{k-1}+1$ c) 2^k d) 2^{k-1} e) $2^{k-2}+1$ f) $2^{k+1}+1$
10. Se consideră șirul primelor $n \times m$ numere naturale unde $n \geq 1$ și $m \geq 1$. Dacă se afișează câte m numere pe o linie, numărul 123 se află pe linia 4 și coloana 3, atunci pe ce linie și coloană se află numărul 167?
a) linia 5, coloana 7 b) linia 4, coloana 7 c) linia 6, coloana 4
d) linia 6, coloana 2 e) linia 5, coloana 2 f) linia 5, coloana 3
11. În secvența de cod următoare se consideră că variabilele a, i, n sunt de tip întreg.

Limbajul C++

```
cin>>n;  
a=1;  
i=2;  
while (i<n && a>0)  
{  
    if(n%i==0)  
        a=0;  
    else  
        i++;  
    cout<<i;  
}
```

Limbajul C

```
scanf("%d", &n);  
a=1;  
i=2;  
while (i<n && a>0)  
{  
    if(n%i==0)  
        a=0;  
    else  
        i++;  
    printf("%d", i);  
}
```

Limbajul Pascal

```
read(n);  
a:=1;  
i:=2;  
while (i<n) and (a>0) do  
begin  
    if(n mod i=0) then  
        a:=0  
    else  
        inc(i);  
    write(i);  
end;
```

Definim, în acest context, **operație** drept o instrucțiune de *atribuire* sau o expresie de *incrementare*. Care este numărul **maxim** de **operații** ce se pot executa în secvența de mai sus, în funcție de valoarea citită pentru variabila n ?

- a) $2n+2$ b) $2n$ c) $n-1$ d) $2n+3$ e) n f) $n+1$

12. Fie secvența de program unde variabila **i** este de tip întreg:

Limbajul C++

```
i=4;
while (i<=25)
{
    cout<<i/10+i%10<<" ";
    i+=2;
}
```

Limbajul C

```
i=4;
while (i<=25)
{
    printf("%d ",i/10+i%10);
    i+=2;
}
```

Limbajul Pascal

```
i:=4;
while i<=25 do
begin
    write(i div 10+i mod 10,' ');
    i:=i+2;
end;
```

Ultimele trei numere afișate sunt:

- a) 2 4 7 b) 2 4 9 c) 6 2 4 d) 9 2 4 e) 2 4 6 f) 9 2 1
13. Fie secvența de cod de mai jos:

Limbajul C++/C

```
float s,p;
float s1(int n)
{
    if(n==0)
        return 2;
    else
        if(n==1)
            return s;
        else
            return s*s1(n-1)-
                p*s1(n-2);
}
```

Limbajul Pascal

```
var s,p:real;
function s1(n : integer) :
real;
begin
    if (n = 0) then s1:=2
    else
        if (n = 1) then s1:=s
        else s1:=s*s1(n-1)-p*s1(n-2);
end;
```

Dacă la apelul subprogramului **s1** se returnează valoarea **82** atunci valorile inițiale ale variabilelor **n**, **s** și **p**, în această ordine, au fost:

- a) 4 3 4 b) 4 4 3 c) 4 2 3 d) 3 3 2 e) 3 1 2 f) 3 4 2
14. Fie secvența de cod unde toate variabilele sunt întregi:

Limbajul C++

```
s=0;
cin>>n>>k;
for(i=1;i<=n;i++)
    for(j=1;j<=n;j++)
    {
        if (i>j)
            t=i-j;
```

```

else
    t=j-i;
if(i==j || t<=k || j==n-i+1 || (i+j)>=n-k+1 && i+j<=n+k+1))
    a[i][j]=1;
else a[i][j]=2;
if(a[i][j]==2)
    s++;
}

```

Limbajul C

```

s=0;
scanf("%d%d", &n, &k);
for(i=1; i<=n; i++)
    for(j=1; j<=n; j++)
    {
        if (i>j)
            t=i-j;
        else
            t=j-i;
        if(i==j || t<=k || j==n-i+1 || (i+j)>=n-k+1 && i+j<=n+k+1))
            a[i][j]=1;
        else
            a[i][j]=2;
        if(a[i][j]==2)
            s++;
    }

```

Limbajul Pascal

```

s:=0;
read(n,k);
for i:=1 to n do
    for j:=1 to n do
        begin
            if(i>j) then
                t:=i-j
            else
                t:=j-i;
            if((i=j) or (t<=k) or (j=n-i+1) or ((i+j)>=n-k+1) and (i+j<=
n+k+1)) then
                a[i,j]:=1
            else
                a[i,j]:=2;
            if(a[i,j] = 2) then
                inc(s);
        end;

```

După execuție, variabila **s** va avea valoarea **8** dacă **n** și **k** au inițial valorile:

a) **n=6; k=2** b) **n=6; k=1** c) **n=5; k=2** d) **n=7; k=1** e) **n=4; k=2** f) **n=6; k=3**

15. Se consideră un tablou bidimensional **a**, cu **3** linii și **3** coloane numerotate de la **1** la **3**, în care $a[i][j]=j+3*(i-1)$, ($1 \leq i, j \leq 3$). Fie secvența de cod de mai jos:

Limbaajul C++

```
k=0;
for (i=1;i<=3;i++)
{for (j=1;j<=3-k;j++)
    cout<<a[α][β]<<" ";
    k++;
}
```

Limbaajul C

```
k=0;
for (i=1;i<=3;i++)
{for (j=1;j<=3-k;j++)
    printf("%d ",a[α][β]);
    k++;
}
```

Limbaajul Pascal

```
k:=0;
for i:=1 to 3 do
begin
    for j:=1 to 3-k do
        write(a[α,β], ' ');
    inc(k);
end;
```

După execuția secvenței se afișează șirul 3 5 7 2 4 1, dacă α și β au valorile:

- a) $\alpha=3-j-k$; $\beta=j$ b) $\alpha=j$; $\beta=3-j-k$ c) $\alpha=j$; $\beta=4-j-k$
d) $\alpha=4-j-k$; $\beta=j$ e) $\alpha=4+j-k$; $\beta=j-1$ f) $\alpha=4+j+k$; $\beta=j+1$

Varianta 6 online

1. Trei variabile de tip întreg au valorile $a=13$, $b=5$, $c=3$. Dintre expresiile următoare, cea care are valoarea **1** (C++/C) respectiv **true** (Pascal) este:

Limbajul C++/C

- a) $a/c*2 < 5+c*4\%5$
- b) $c\%b == a\%c$
- c) $b+a/10 != b\%c*a/c$
- d) $(b > c) \ \&\& \ ! (b*c\%7 == 2*a-b*b)$
- e) $c\%b*10 < a*2$
- f) $c/b*b/c == 1$

Limbajul Pascal

- a) $a \text{ div } c * 2 < (5+c*4 \text{ mod } 5)$
- b) $c \text{ mod } b = a \text{ mod } c$
- c) $b+a \text{ div } 10 <> b \text{ mod } c * a \text{ div } c$
- d) $(b > c) \text{ and not } ((b*c \text{ mod } 7) = (2*a-b*b))$
- e) $c \text{ mod } b*10 < a*2$
- f) $c \text{ div } b * b \text{ div } c = 1$

2. Într-un graf neorientat cu **13** noduri, fiecare nod are gradul **d**. Valoarea lui **d** nu poate fi:

- a) 2
- b) 4
- c) 6
- d) 8
- e) 10
- f) 11

3. Variabila **i** este de tip întreg. Numărul total al atribuirilor care se execută în urma rulării secvenței următoare este:

Limbajul C++/C

```
i=1;
while(i*i<2020)
    i=i*2;
```

- a) 5
- b) 6
- c) 7
- d) 9
- e) 11
- f) 12

Limbajul Pascal

```
i:=1;
while i*i<2020 do
    i:=i*2;
```

4. Considerând că variabila **s** poate reține un șir cu cel mult **100** de caractere și variabila **i** este de tip întreg, în urma executării următoarei secvențe de instrucțiuni, lungimea efectivă a șirului **s** este:

Limbajul C++/C

```
strcpy(s, "2020+2020=4040");
for(i=0; i<strlen(s); i++)
    if(strchr("0123456789", s[i]))
        strcpy(s+i, s+i+1);
```

- a) 0
- b) 2
- c) 5
- d) 6
- e) 8
- f) 11

Limbajul Pascal

```
s:='2020+2020=4040';
for i:=1 to length(s) do
    if pos(s[i], '0123456789') <> 0
        then delete(s, i, 1);
```

5. Pentru a verifica dacă elementele unui tablou unidimensional cu **n** elemente numere întregi sunt distincte două câte două, numărul de comparații executate este:

- a) $2n$
- b) $n(n-1)/2$
- c) $n(n-1)$

d) $(n-1)^2$

e) n^2

f) $n!$

6. În urma executării următoarei secvențe de program, variabila x , de tip întreg, va avea valoarea:

Limbajul C++/C

```
x=15;  
x=x*3/4*4/3;  
do {if(x%2==0) x=x/2;  
    else x=x-5;  
   }while(x>0);
```

Limbajul Pascal

```
x:=15;  
x:=x*3 div 4*4 div 3;  
repeat  
if x mod 2=0 then x:=x div 2  
else x:=x-5;  
until x<=0;
```

a) -6

b) -5

c) -4

d) 0

e) 2

f) 5

7. Utilizând un algoritm backtracking se generează în ordine crescătoare toate numerele naturale cu patru cifre care au suma cifrelor egală cu 4. Primele trei soluții sunt: 1003, 1012, 1021. În șirul generat, numărul 2020 ocupă poziția:

a) 10

b) 11

c) 12

d) 13

e) 14

f) 15

8. Dacă s, i, j, n sunt variabile de tip întreg și a este un tablou bidimensional cu n linii și n coloane numerotate de la 1 la n , următorul algoritm calculează:

Limbajul C++/C

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

Limbajul Pascal

```
s:=0;  
for i:=1 to n do  
    for j:=1 to i-1 do  
        s:=s+a[i,j];
```

- a) suma elementelor de sub diagonala principală exclusiv elementele diagonalei principale
- b) suma elementelor de sub diagonala secundară exclusiv elementele diagonalei secundare
- c) numărul elementelor de deasupra diagonalei principale inclusiv elementele diagonalei principale
- d) suma elementelor de pe diagonala principală
- e) suma elementelor de sub diagonala principală inclusiv elementele diagonalei principale
- f) suma elementelor de deasupra diagonalei secundare inclusiv elementele diagonalei secundare

9. Se consideră următoarele declarații de tipuri și variabile:

Limbajul C++

```
struct a  
{ int b;  
  char c[10];  
};  
struct d  
{ char e[10];  
  float f;
```

Limbajul Pascal

```
type a=record  
    b:integer;  
    c:string[10]  
end;  
d=record  
    e:string[10];  
    f:real;
```

```

    a g;
} h;

```

```

    g:a
end;
var h:d;

```

Limbajul C

```

typedef struct
{ int b;
  char c[10];
}a;
typedef struct
{ char e[10];
  float f;
  a g;
}d;
d h;

```

Dintre următoarele expresii, de tip caracter este:

- a) **g.e[2]** b) **h.a.c** c) **h.a.c[0]**
d) **h.c[2]** e) **h.g.c[2]** f) **d.e[2]**

10. Se consideră un arbore cu 8 noduri și muchiile [1,2], [2,3], [3,6], [4,3], [5,7], [7,2], [8,2]. Pentru ca arborele să conțină un număr maxim de lanțuri elementare de lungime 3 care nu conțin rădăcina, se poate alege ca rădăcină **oricare** dintre nodurile:

- a) 1, 2, 4, 5 b) 1, 2, 5, 6 c) 1, 3, 6, 7
d) 2, 3, 4, 5, 6 e) 1, 4, 5, 6, 8 f) 2, 3, 4, 7, 8

11. În urma executării următoarei secvențe de program tabloul unidimensional **a**, cu 6 elemente numerotate de la 1 la 6, va conține valorile:

Limbajul C++/C

```

for (i=1;i<=6;i++)
  if (i%2!=0) a[i]=i/2;
  else a[i]=7-i;
for(i=6;i>=3;i--)
  a[a[i]]=2*i%7;

```

Limbajul Pascal

```

for i:=1 to 6 do
  if i mod 2<>0 then a[i]:=i div 2
  else a[i]:=7-i;
for i:=6 downto 3 do
  a[a[i]]:=2*i mod 7;

```

- a) 0 5 6 1 3 5 b) 5 3 1 3 2 1 c) 6 3 6 2 2 3
d) 6 1 5 2 6 3 e) 6 3 1 2 2 1 f) 6 3 1 3 2 1

12. Pentru apelul **s(2020,2)** al subprogramului de mai jos, enunțul adevărat este:

Limbajul C++/C

```

int s(int n, int d)
{
  if(n==1) return 0;
  if (n%d==0)
    return 1+s(n/d,d);
  else
    return s(n,d+1);
}

```

Limbajul Pascal

```

function s(n,d:integer): integer;
begin
  if n=1 then
    s:=0
  else if n mod d=0 then
    s:=1+s(n div d,d)
  else s:=s(n,d+1)
end;

```

- a) $s(2020, 2) = 3$ și reprezintă numărul divizorilor primi ai numărului 2020
- b) $s(2020, 2) = 4$ și reprezintă numărul divizorilor primi ai numărului 2020
- c) $s(2020, 2) = 4$ și reprezintă suma exponenților divizorilor primi din descompunerea în factori primi a numărului 2020
- d) $s(2020, 2) = 6$ și reprezintă suma exponenților divizorilor primi din descompunerea în factori primi a numărului 2020
- e) $s(2020, 2) = 10$ și reprezintă numărul divizorilor proprii ai numărului 2020
- f) $s(2020, 2) = 12$ și reprezintă numărul divizorilor numărului 2020

13. Numărul de grafuri neorientate cu șase noduri, în care nodul 1 are gradul 1 și nodul 2 are gradul 2 este:

- a) 92 b) 1280 c) 1536 d) 1792 e) 1920 f) 2560

14. În urma rulării programului următor vor fi afișate valorile:

```

Limbajul C++
#include <iostream>
using namespace std;
void f (int &a, int b)
{ int x=3;
  a--;
  b++;
  x--;
  cout<<a<<' '<<b<<' '<<x<<' ';
}
int main()
{ int i, x=4, y=6;
  for(i=1;i<=3;i++)
    f(x,x+y);
  cout<<x<<' '<<y;
  return 0;
}

```

```

Limbajul C
#include <stdio.h>
#include <stdlib.h>
void f (int *a, int b)
{ int x=3;
  (*a)--;
  b++;
  x--;
  printf("%d %d %d ",*a,b,x);
}
int main()
{ int i, x=4, y=6;
  for(i=1; i<=3; i++)
    f(&x,x+y);
  printf("%d %d",x,y);
  return 0;
}

```

```

Limbajul Pascal
program main;
var x, y, i: integer;
procedure f (var a: integer; b:integer);
var x:integer;
begin
  x:=3;
  dec(a); inc(b); dec(x);
  write(a,' ',b,' ',x,' ');
end;
begin
  x:=4;
  y:=6;
  for i:=1 to 3 do
    f(x,x+y);
  write(x,' ',y)
end.

```


Varianta 7 online

1. Expresia de mai jos are valoarea **1** (C++/C) respectiv **true** (Pascal) dacă și numai dacă **n** este:

Limbajul C++/C

```
n%2==1 && n*n<100
```

Limbajul Pascal

```
(n mod 2=1) and (n*n<100)
```

- a) număr întreg impar mai mic decât **10**
- b) număr întreg impar, din intervalul **(-10,10)**
- c) număr natural mai mic decât **100**
- d) număr natural impar de o singură cifră
- e) număr întreg par mai mic decât **10**
- f) număr natural impar cu cel mult două cifre

2. Dacă **a** este un tablou bidimensional cu **n** linii și **n** coloane, numerotate de la **1** la **n**, elementul de pe linia **i** și coloana **j** se află pe diagonala secundară dacă între indici există relația:

a) **i<j**

b) **i>j**

c) **i=j**

d) **i+j=n-1**

e) **i+j=n**

f) **i+j=n+1**

3. Graful neorientat complet **G** are **10** noduri. Un enunț adevărat este:

a) **G** este arbore

b) **G** are 50 de muchii

c) **G** nu este graf hamiltonian și nici eulerian

d) **G** este graf hamiltonian dar nu eulerian

e) **G** nu este graf hamiltonian dar este graf eulerian

f) **G** este graf hamiltonian și eulerian

4. Se consideră că **d**, **i**, **k**, **n** sunt variabile de tip întreg și **a** este un tablou unidimensional cu **n** numere întregi numerotate de la **1** la **n**. La finalul execuției secvenței următoare, variabila **k** are valoarea **1** dacă și numai dacă elementele tabloului **a** formează o progresie aritmetică. Expresia corectă care completează punctele de suspensie este:

Limbajul C++/C

```
k=1;
```

```
d=a[2]-a[1];
```

```
for(i=3;i<=n;i++)
```

```
    if ( . . . )
```

```
        k=0;
```

Limbajul Pascal

```
k:=1;
```

```
d:=a[2]-a[1];
```

```
for i:=3 to n do
```

```
    if . . . then
```

```
        k:=0;
```

Limbajul C++/C

a) **a[i+1]-a[i]!=d**

b) **a[i]-a[i+1]!=d**

c) **a[i]-a[i-1]!=d**

d) **a[i+1]-a[i]==d**

e) **a[i]+a[i+1]!=d**

f) **a[i]-a[i-1]==d**

Limbajul Pascal

a) **a[i+1]-a[i]<>d**

b) **a[i]-a[i+1] <>d**

c) **a[i]-a[i-1]<>d**

d) **a[i+1]-a[i]=d**

e) **a[i]+a[i+1]<>d**

f) **a[i]-a[i-1]=d**

5. Se consideră un arbore cu 8 noduri și muchiile [1,2], [2,3], [3,6], [4,3], [5,7], [7,2], [8,2]. Înălțimea arborelui este egală cu lungimea celui mai lung lanț elementar care unește rădăcina de o frunză. Arborele dat are înălțime minimă dacă se va alege ca rădăcină nodul:

a) 1 b) 2 c) 3 d) 5 e) 7 f) 8

6. În urma execuției secvenței următoare, în care toate variabilele sunt de tip întreg, valoarea variabilei **n** este:

Limbajul C++/C

```
n=0; a=11357; b=1426; p=1;
while(a!=b)
{ x=a%10;y=b%10;
  if(x<y) n=n+p*x;
  else n=n+p*y;
  p=p*10;a=a/10;b=b/10;
}
```

Limbajul Pascal

```
n:=0; a:=11357; b:=1426; p:=1;
while a<>b do begin
x:=a mod 10; y:=b mod 10;
if x<y then n:=n+p*x
else n:=n+p*y;
p:=p*10; a:=a div 10; b:=b
div 10
end;
```

a) 1326 b) 1356 c) 6241
d) 11326 e) 11457 f) 62411

7. Fie enunțul: „pentru a sorta descrescător un tablou unidimensional cu 20 de elemente numere reale, utilizând metoda selecției, nu sunt necesare mai mult de **x** determinări ale valorii maxime”. Enunțul este adevărat dacă **x** este egal cu:

a) 0 b) 10 c) 19 d) 20 e) 190 f) 400

8. Matricea alăturată este matricea de adiacență a unui graf:

```
0 1 1 0
1 0 1 0
1 0 0 0
0 1 0 0
```

a) orientat cu 6 noduri și 3 arce b) neorientat cu 4 noduri și 3 muchii
c) orientat cu 4 noduri și 6 arce d) neorientat cu 6 noduri și 6 muchii
e) orientat cu 4 noduri și 3 arce f) neorientat cu 4 noduri și 6 muchii

9. Utilizând un algoritm backtracking se generează în ordine lexicografică toate anagramele cuvântului **roman**. Soluția generată imediat înainte de cuvântul **norma** și soluția generată imediat după cuvântul **norma** sunt:

a) **nramo** și **noram** b) **nramo** și **nrmao** c) **nomra** și **noram**
d) **nomra** și **nramo** e) **noram** și **nramo** f) **nomar** și **nramo**

10. Variabilele **i**, **j**, **k** sunt de tip întreg iar **s** reține un șir de caractere format din litere mici și spații (cuvintele sunt despărțite printr-un singur spațiu). În urma executării următoarei secvențe de program, variabila **k** are valoarea 0 dacă șirul **s** este inițial:

Limbajul C++/C

```
for(i=0;i<strlen(s);i++)
  if(s[i]==' ')
    strcpy(s+i,s+i+1);
```

Limbajul Pascal

```
for i:=1 to length(s) do
  if s[i]=' ' then
    delete(s,i,1);
```

```

i=0;
j=strlen(s)-1;
k=1;
while (i<j)
{
    if (s[i]!=s[j])
        k=0;
    i++;
    j--;
}

```

- a) atasata
d) elisa vasile

```

i:=1;
j:=length(s);
k:=1;
while i<j do
begin
    if s[i]<>s[j] then
        k:=0;
    inc(i);
    dec(j)
end;

```

- b) o rama maro
e) nora aron
c) o rama alba
f) vasile elisav

11. Dacă din programul principal se apelează $f(f(3))$, numărul de autoapeluri ale funcției f , definită mai jos, este:

Limbajul C++/C

```

int f (int a)
{
    if (a<2)
        return 1;
    else
        return f(a-1)+2*f(a-3);
}

```

- a) 8 b) 9 c) 10

Limbajul Pascal

```

function f(a:integer):integer;
begin
    if a<2 then
        f:=1
    else
        f:=f(a-1)+2*f(a-3)
end;

```

- d) 14 e) 15 f) 16

12. Secvența de mai jos construiește tabloul bidimensional a cu n linii și n coloane, numerotate de la 1 la n . Pentru $n=4$, suma elementelor de pe diagonala principală este:

Limbajul C++/C

```

x=1;
y=1;
for (i=1; i<=n; i++)
    for (j=1; j<=n+1-i; j++)
    {
        a[i][j]=x;
        x++;
    }
for (j=n; j>=1; j--)
    for (i=n; i>=n+1-j; i--)
    {
        a[i][j]=y;
        y++;
    }

```

- a) 9 b) 12 c) 14

Limbajul Pascal

```

x:=1;
y:=1;
for i:=1 to n do
    for j:=1 to n+1-i do
        begin
            a[i,j]:=x;
            inc(x)
        end;
for j:=n downto 1 do
    for i:=n downto n+1-j do
        begin
            a[i,j]:=y;
            inc(y)
        end;

```

- d) 16 e) 28 f) 30

13. Pentru funcția dată mai jos, $f(95)$ și $f(59)$ au valorile:

Limbajul C++/C

```

int f (int x)
{

```

Limbajul Pascal

```

function f (x:integer) :
integer;

```

```

if (x>=100)
    return x+2;
else
    return f (f(x+2)+1);
}

```

```

begin
    if x>=100 then
        f:=x+2
    else
        f:=f(f(x+2)+1)
end;

```

a) 103 și 146 b) 109 și 162 c) 110 și 163
d) 103 și 163 e) 112 și 157 f) 112 și 166

14. Sortând crescător prin metoda selecției, cu număr minim de interschimbări (se interschimbă doar elemente distincte), tablourile unidimensionale $v=(3, 8, 2, 7)$, $x=(4, 5, 1, 7)$, $y=(4, 7, 9, 6)$ și $z=(6, 3, 2, 9)$ se calculează numărul operațiilor (comparări și atribuiri) efectuate. Afirmația adevărată este:
- a) Pentru v și y s-a realizat un număr egal de operații
 - b) Pentru v și z s-a realizat un număr egal de operații
 - c) Cel mai mare număr de operații s-a efectuat pentru x
 - d) Cel mai mare număr de operații s-a efectuat pentru y
 - e) Cel mai mic număr de operații s-a efectuat pentru z
 - f) Cel mai mic număr de operații s-a efectuat pentru y

15. În urma executării secvenței de program de mai jos se afișează:

```

Limbajul C++/C
int f (int a, int b, int e)
{
    int x;
    if(a<2)
        return e+1;
    if(a%b==0)
    {
        if(e==0)
            cout<<b<<' ';
        |printf("%d ",b);
        e++;
        return f(a/b,b,e);
    }
    else
    {
        x=e+1;
        e=0;
        b++;
        return x*f(a,b,e);
    }
}
int main()
{ int x,y,e;
  cin>>x; |scanf("%d",&x);
  y=2;
  e=0;

```

```

Limbajul Pascal
program p;
var x,y,e: integer;
function f(a,b,e:integer) :integer;
var x:integer;
begin
    if a<2 then
        f:=e+1
    else
        begin
            if a mod b=0 then
                begin
                    if e=0 then
                        write(b,' ');
                    inc(e);
                    f:=f(a div b,b,e)
                end
            else
                begin
                    x:=e+1;e:=0;inc(b);
                    f:=x*f(a,b,e)
                end
            end
        end
end;
end;
begin
    read(x);
    y:=2;

```



```
    cout<<f(x,y,e) ;           |   e:=0 ;
|printf("%d",f(x,y,e)) ;     |   writeln(f(x,y,e))
}                             |end.
```

- a) divizorii proprii ai numărului x
- b) numărul de divizori proprii ai numărului x
- c) divizorii proprii și numărul divizorilor proprii ai numărului x
- d) divizorii primi ai lui x și numărul tuturor divizorilor lui x
- e) divizorii proprii ai numărului x și produsul exponenților divizorilor primi din descompunerea în factori primi a numărului x
- f) divizorii primi ai numărului x și produsul exponenților divizorilor primi din descompunerea în factori primi a numărului x

Varianta 8 online

1. Se dă o variabilă **a** care reține un număr natural nenul. Expresia care are valoarea **0/false** pentru orice număr natural nenul **a** este:

- a) C++/C: $(a/3+a/7)\%9$
Pascal: $(a \text{ DIV } 3+a \text{ DIV } 7) \text{ MOD } 9$
- b) C++/C: $(a\%10+a\%100/10)/10$
Pascal: $((a \text{ MOD } 10)+(a \text{ MOD } 100) \text{ DIV } 10) \text{ DIV } 10$
- c) C++/C: $((10-a\%10)+(10-a\%100/10))/10$
Pascal: $((10-(a \text{ MOD } 10))+(10-(a \text{ MOD } 100) \text{ DIV } 10)) \text{ DIV } 10$
- d) C++/C: $(a\%5+a\%7)/10$
Pascal: $(a \text{ MOD } 4+a \text{ MOD } 6) \text{ DIV } 10$
- e) C++/C: $(a\%3+a\%7)/9$
Pascal: $(a \text{ MOD } 3+a \text{ MOD } 7) \text{ DIV } 9$
- f) C++/C: $(a\%10+a/10)/9$
Pascal: $(a \text{ MOD } 10+a \text{ DIV } 10) \text{ DIV } 9$

2. Se dă un tablou unidimensional $\mathbf{v}=(3, 5, 8, 4, 2, 6, 9, 1)$ în care primul element se află pe poziția 0 și **i** o variabilă de tip întreg. În urma executării secvenței de instrucțiuni, elementele tabloului unidimensional **v** sunt:

Limbajul C++/C

```
i=0;
while (i<=6)
{
j=i+1;
v[i]=v[i]+v[j];
v[j]=v[i]-v[j];
v[i]=v[i]-v[j];
i=i+2;
}
```

Limbajul Pascal

```
i:=0;
while i<=6 do
begin
j:=i+1;
v[i]:=v[i]+v[j];
v[j]:=v[i]-v[j];
v[i]:=v[i]-v[j];
i:=i+2;
end;
```

- a) $\mathbf{v}=(5, 8, 4, 2, 6, 9, 1, 3)$
- b) $\mathbf{v}=(5, 3, 4, 8, 6, 2, 1, 9)$
- c) $\mathbf{v}=(11, 3, 20, 8, 10, 2, 19, 9)$
- d) $\mathbf{v}=(5, -7, 4, 0, 6, 10, 1, 6)$
- e) $\mathbf{v}=(3, 1, 9, 6, 2, 4, 8, 5)$
- f) $\mathbf{v}=(9, 1, 2, 6, 8, 4, 3, 5)$

3. Știind că variabila **i** este de tip întreg și variabila **a** de tip șir de caractere reține cuvântul **politehnica**, în urma executării instrucțiunilor se va afișa:

Limbajul C++/C

```
for (i=0; i<=7; i++)
if (a[i]<'n')
a[i]='A'-'a'+a[i];
cout<<a; | printf("%s", a);
```

Limbajul Pascal

```
for i:=1 to 8 do
begin
if a[i]<'n' then
a[i]:=upcase(a[i])
end;
```

- | | | |
|----------------|------------------------|----------------|
| | <code>write(a);</code> | |
| a) poLIteHnICA | b) POLiTehnica | c) POLiTehnica |
| d) POLiTehNICA | e) poliTEHNICA | f) poLIteHnica |

4. Fie un tablou bidimensional **A**, cu 4 linii și 4 coloane numerotate de la 0 la 3 care conține elemente de tip întreg și două variabile **i** și **j** de tip întreg. Valorile ce vor fi reținute în tabloul bidimensional **A** după executarea următoarelor instrucțiuni sunt:

Limbajul C++/C

```

i=3;
while(i>=0)
{
  j=3;
  while(j>=0)
  {
    if((i+j)%2==0)
      A[i][j]=i+j;
    else
      if(i>j) A[i][j]=i;
      else A[i][j]=j;
    j--;
  }
  i--;
}

```

Limbajul Pascal

```

i:=3;
while i>=0 do
begin
  j:=3;
  while j>=0 do
  begin
    if (i+j) MOD 2 =0 then
      A[i,j]:=i+j
    else
      if i>j then
        A[i,j]:=i
      else
        A[i,j]:=j;
    j:=j-1
  end;
  i:=i-1
end;

```

$$a) A = \begin{pmatrix} 0 & 0 & 2 & 0 \\ 0 & 2 & 1 & 4 \\ 2 & 1 & 4 & 2 \\ 0 & 4 & 2 & 6 \end{pmatrix}$$

$$b) A = \begin{pmatrix} 0 & 1 & 0 & 3 \\ 1 & 1 & 3 & 1 \\ 0 & 3 & 2 & 5 \\ 3 & 1 & 5 & 3 \end{pmatrix}$$

$$c) A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 1 & 3 & 3 \\ 2 & 3 & 2 & 5 \\ 3 & 3 & 5 & 3 \end{pmatrix}$$

$$d) A = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 2 & 2 & 4 \\ 2 & 2 & 4 & 3 \\ 3 & 4 & 3 & 6 \end{pmatrix}$$

$$e) A = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 3 & 3 \\ 1 & 3 & 2 & 5 \\ 3 & 3 & 5 & 3 \end{pmatrix}$$

$$f) A = \begin{pmatrix} 0 & 2 & 1 & 3 \\ 2 & 1 & 3 & 5 \\ 1 & 3 & 5 & 3 \\ 3 & 5 & 3 & 3 \end{pmatrix}$$

5. Fie o stivă inițial vidă. Cu ajutorul subprogramelor **Ad(x)**, respectiv **E1()** este adăugat elementul **x**, respectiv șters un element din stivă. Suma elementelor din stivă după executarea operațiilor următoare este:

Ad(3) Ad(7) Ad(5) E1() E1() Ad(8)

- a) 3 b) 7 c) 10 d) 11 e) 12 f) 13

6. Se dă mulțimea $A = \{1, 4, 5, 8, 9\}$. Un algoritm generează în ordine crescătoare, toate numerele naturale de n cifre, folosind cifre distincte din mulțimea A , care nu au alăturate cifre de aceeași paritate. Dacă pentru $n=4$, primele patru soluții generate sunt: **1458, 1498, 1854, 1894**, numărul de soluții pe care le va genera algoritmul este:

a) 12 b) 16 c) 20 d) 24 e) 28 f) 30

7. Șirul care poate reprezenta valorile gradelor nodurilor unui graf neorientat cu 6 noduri este:

a) 2 5 0 2 1 2 b) 2 2 1 1 1 2 c) 2 2 7 2 2 1
d) 2 2 0 0 4 2 e) 2 3 1 1 2 2 f) 2 2 2 1 1 2

8. Șirul de valori care poate fi vectorul de tați al unui arbore cu 8 noduri este:

a) $T = (0\ 1\ 8\ 3\ 2\ 5\ 5\ 4)$ b) $T = (0\ 3\ 7\ 2\ 4\ 5\ 8\ 3)$
c) $T = (0\ 1\ 1\ 2\ 4\ 5\ 8\ 7)$ d) $T = (0\ 5\ 7\ 3\ 1\ 3\ 1\ 2)$
e) $T = (0\ 1\ 0\ 2\ 4\ 6\ 3\ 3)$ f) $T = (8\ 5\ 7\ 3\ 1\ 3\ 1\ 2)$

9. Știind că i, j, s și a sunt patru variabile de tip întreg, pentru orice valoare naturală nenulă a variabilei a , după executarea instrucțiunilor, valoarea afișată corespunde formulei matematice:

Limbajul C++/C

```
s=0;
for(i=1;i<=a;i++)
{
    j=1;
    while(j<=i)
    {
        s++;
        j++;
    }
    j=i+1;
    do
    {
        s++;
        j++;
    }while(j<=a);
}
cout<<s--; | printf("%d", s--);
```

Limbajul Pascal

```
s:=0;
for i:=1 to a do
begin
    j:=1;
    while j<=i do
    begin
        s:=s+1;
        j:=j+1
    end;
    j:=i+1;
    repeat
        s:=s+1;
        j:=j+1;
    until j>a;
end;
write(s);
```

a) $a(a+1)$ b) a^2+1 c) a^2
d) a^2-1 e) $a(a-1)$ f) $2a^2-1$

10. Subprogramul **afis** primește ca parametru un tablou bidimensional v cu n linii și n

coloane, numerotate de la 1 la n, unde

$$v = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 4 & 5 & 6 & 7 \\ 8 & 9 & 0 & 1 \\ 2 & 3 & 4 & 5 \end{pmatrix}, n=4 \text{ și } k=2 \cdot n.$$

Pentru valorile date, **afis(v,n,k)** va afișa:

Limbajul C++/C

```
void afis(int v[100][100],int n,int k)
{int i;
if(k>1)
  { for(i=n;i>=1;i--)
    if(k-i<=n && k-i>0)
      cout<<v[i][k-i]; | printf("%d", v[i][k-i]);
afis(v,n,k-2);}
}
```

Limbajul Pascal

```
type matrice = array [1..100,1..100] of integer;
procedure afis(var v:matrice; n:integer; k:integer);
var i: integer;
begin
if k<>1 then
begin
  for i:=n downto 1 do
    if(k-i<=n) AND (k-i>0) then
      write(v[i,k-i]);
  afis(v,n,k-2)
end
end;
```

a) 08523075

b) 57032580

c) 02587035

d) 53078520

e) 35087250

f) 70358520

11. Știind că subprogramul **functie** corespunde funcției matematice $f(x)=3 \cdot x-1$, pentru orice x număr întreg, **abc(t,c)** va calcula:

Limbajul C++/C

```
int functie(int x)
{return 3*x-1;}

int abc(int t, int c)
{ if(c==0) return t;
  else
    return abc(functie(t),c-1);}
```

Limbajul Pascal

```
function functie(var x:integer):integer;
```

```

begin
    functie:=3*x-1
end;

function abc(t,c:integer):integer;
begin
    if c=0 then
        abc:=t
    else
        abc:=abc(functie(t),c-1)
    end;
end;

```

- | | |
|---|---|
| a) $\underbrace{f(t) \circ \dots \circ f(t)}_{c-1}$ | b) $\underbrace{f(t) + \dots + f(t)}_{c-1}$ |
| c) $f(t)^c$ | d) $c * f(t)$ |
| e) $(c - 1) * f(t)$ | f) $\underbrace{f(t) \circ \dots \circ f(t)}_c$ |

12. După executarea următoarelor instrucțiuni se va afișa:

<pre> Limbajul C++/C char a[20][20]; int i; strcpy(a[1], "bacalaureat"); strcpy(a[2], "liceu"); strcpy(a[3], "examene"); strcpy(a[4], "politehnica"); for (i=1; i<=4; i++) cout<<a[i][2*i]; printf("%d", a[i][2*i]); </pre>	<pre> Limbajul Pascal var a:array[1..20] of string; i:integer; begin a[1]:='bacalaureat'; a[2]:='liceu'; a[3]:='examene'; a[4]:='politehnica'; for i:=1 to 4 do write(a[i,2*i+1]) end. </pre>
--	---

- | | | |
|---------|---------|---------|
| a) aenn | b) teen | c) cunc |
| d) cuei | e) bceh | f) ceen |

13. Următoarele instrucțiuni vor afișa:

<pre> Limbajul C++ int f1(int x, int &y) { x=x+2; y=y-1; return x+y; x=x+1; } int main() </pre>	<pre> Limbajul C int f1(int x, int *y) { x=x+2; *y=*y-1; return x+*y; x=x+1; } int main() </pre>
---	--

```

{
int n=3,m=6;
cout<<f1(f1(m,n),m);
cout<<" "<<m;
}

```

```

{
int n=3,m=6;
printf("%d ",
f1(f1(m,&n),&m));
printf("%d", m);
}

```

Limbajul Pascal

```

function f1(x:integer; var y:integer):integer;
begin
  x:=x+2;
  y:=y-1;
  f1:=x+y;
  x:=x+1
end;
var m,n: integer;
begin
m:=6;
n:=3;
write(f1(f1(m,n),m),' ',m)
end.

```

a) 17 5 b) 17 6 c) 10 5 d) 10 6 e) 11 6 f) 10 7

14. Valorile care vor fi memorate în tabloul bidimensional **b**, cu liniile și coloanele numerotate de la 1 la **n**, după apelul **matrice(a,b,n,q)**, unde

$$a = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{pmatrix}, \quad b = \mathbf{0}_3, \quad n=3, \quad q=2, \text{ sunt:}$$

Limbajul C++/C

```

void matrice(int a[][100], int b[][100], int n, int q)
{int i,j,k;
  if(q>1)
  {
    for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
        for(k=1;k<=n;k++)
          b[i][j]=b[i][j]+a[i][k]*a[k][j];
    matrice(a,b,n,q-1);
  }
}

```

Limbajul Pascal

```

type matrix = array [1..100,1..100] of integer;
procedure matrice(a:matrix; b:matrix; n:integer;
q:integer);
var i,j,k: integer;
begin

```

```

if q>1 then
begin
for i:=1 to n do
  for j:=1 to n do
    for k:=1 to n do
      b[i,j]:=b[i,j]+a[i,k]*a[k,j];
    matrice(a,b,n,q-1)
  end
end
end;

```

$$a) \mathbf{b} = \begin{pmatrix} 2 & 3 & 4 \\ 3 & 4 & 5 \\ 4 & 5 & 6 \end{pmatrix}$$

$$b) \mathbf{b} = \begin{pmatrix} 47 & 38 & 29 \\ 62 & 50 & 38 \\ 77 & 62 & 47 \end{pmatrix}$$

$$c) \mathbf{b} = \begin{pmatrix} 4 & 6 & 8 \\ 6 & 8 & 10 \\ 8 & 10 & 12 \end{pmatrix}$$

$$d) \mathbf{b} = \begin{pmatrix} 4 & 9 & 16 \\ 9 & 16 & 25 \\ 16 & 25 & 36 \end{pmatrix}$$

$$e) \mathbf{b} = \begin{pmatrix} 29 & 38 & 47 \\ 38 & 50 & 62 \\ 47 & 62 & 77 \end{pmatrix}$$

$$f) \mathbf{b} = \begin{pmatrix} 77 & 62 & 47 \\ 62 & 50 & 38 \\ 47 & 38 & 29 \end{pmatrix}$$

15. În urma executării programului de mai jos se afișează:

```

Limbajul C++
#include <iostream>
using namespace std;
void functie(int &a,int &b)
{ b=3*b;
  a=2*a;}
int main()
{ int n=4;
  functie(n,n);
  cout<<n;}

```

```

Limbajul C
#include <stdio.h>
void functie(int *a,int *b)
{ *b=3*(*b);
  *a=2*(*a);}
int main()
{ int n=4;
  functie(&n,&n);
  printf("%d", n);
}

```

```

Limbajul Pascal
procedure functie (var a:integer; var b:integer);
begin
  b:=3*b;
  a:=2*a
end;

var n:integer;
begin
  n:=4;
  functie(n,n);
  write(n)
end.

```

a) eroare b) 4 c) 6 d) 12 e) 14 f) 24

Varianta 9 online

1. Rezultatul expresiei de mai jos este:

Limbajul C++/C

```
16 / (-5 % 3) * 3
```

a) 48 b) -24 c) -48

Limbajul Pascal

```
16 div (-5 mod 3) * 3
```

d) -2 e) 2 f) 6

2. Expresia corespunzătoare penultimei cifre a numărului natural având cel puțin două cifre reținut de variabila întregă n este:

Limbajul C++/C

a) $n / 10 / 10$

b) $n / 10 \% 10$

c) $n \% 10 \% 10$

d) $n \% 10 / 10$

e) $n \% 10 / 100$

f) $n / 100 \% 10$

Limbajul Pascal

a) $n \text{ div } 10 \text{ div } 10$

b) $n \text{ div } 10 \text{ mod } 10$

c) $n \text{ mod } 10 \text{ mod } 10$

d) $n \text{ mod } 10 \text{ div } 10$

e) $n \text{ mod } 10 \text{ div } 100$

f) $n \text{ div } 100 \text{ mod } 10$

3. Afirmatia adevărată în privința secvenței de instrucțiuni de mai jos este:

Limbajul C++/C

```
d = 1;
```

```
while (d * d <= n)
```

```
{
```

```
    if (n % d == 0)
```

```
    {
```

```
        d1 = d;
```

```
        d2 = n / d;
```

```
    }
```

```
    d++;
```

```
}
```

Limbajul Pascal

```
d := 1;
```

```
while d * d <= n do
```

```
begin
```

```
    if n mod d = 0 then
```

```
    begin
```

```
        d1 := d;
```

```
        d2 := n div d
```

```
    end;
```

```
    d := d + 1
```

```
end
```

- a) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține un număr prim.
b) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține cubul unui număr prim.
c) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține un număr impar.
d) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține un număr par.
e) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține un număr impar.
f) La final $d1$ și $d2$ vor fi egale **doar** dacă n reține un număr pătrat perfect.

4. Secvența de instrucțiuni de mai jos ordonează crescător cele n elemente ale tabloului unidimensional v , în care primul element este memorat pe poziția 0, dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
for (i = 0; i < n - 1; i++)
```

```
{
```

```
    ...
```

```
    {
```

```
        if (v[j] > v[j+1])
```

Limbajul Pascal

```
for i := 0 to n - 2 do
```

```
begin
```

```
    ...
```

```
begin
```

```
    if v[j] > v[j+1] then
```

<pre> { aux = v[j]; v[j] = v[j+1]; v[j+1] = aux; } } </pre>	<pre> begin aux := v[j]; v[j] := v[j+1]; v[j+1] := aux end end </pre>
---	---

Limbajul C++/C

- | | |
|-------------------------------|-------------------------------|
| a) for (j = n-2; j >= i; j--) | b) for (j = 0; j <= i; j++) |
| c) for (j = n-i; j >= i; j--) | d) for (j = 1; j < i; j++) |
| e) for (j = n-1; j > i; j--) | f) for (j = 1; j <= i+1; j++) |

Limbajul Pascal

- | | |
|-------------------------------|-------------------------|
| a) for j := n-2 downto i do | b) for j := 0 to i do |
| c) for j := n-i downto i do | d) for j := 1 to i-1 do |
| e) for j := n-1 downto i+1 do | f) for j := 1 to i+1 do |

5. Subprogramul **f** este definit mai jos. O condiție necesară și suficientă pentru ca numărul natural mai mare strict ca **1** reținut de variabila **n** să fie prim este:

Limbajul C++/C

```

int f(int d, int n)
{
  do
  {
    d++;
  }
  while (n % d != 0);
  return d;
}

```

Limbajul Pascal

```

function f(d, n: integer):
integer;
begin
  repeat
    d := d + 1;
  until n mod d = 0;
  f := d
end;

```

Limbajul C/C++

- | | |
|-----------------------|-----------------------|
| a) $f(2, n) == n$ | b) $f(2, n) == 2$ |
| c) $f(1, n) == n$ | d) $f(1, n) == 1$ |
| e) $f(1, n - 1) == n$ | f) $f(2, n - 1) == 2$ |

Limbajul Pascal

- | | |
|----------------------|----------------------|
| a) $f(2, n) = n$ | b) $f(2, n) = 2$ |
| c) $f(1, n) = n$ | d) $f(1, n) = 1$ |
| e) $f(1, n - 1) = n$ | f) $f(2, n - 1) = 2$ |

6. Numărul de muchii care trebuie adăugate unui **arbore** cu **10** vârfuri astfel încât acesta să devină **graf complet** este:

- | | | | | | |
|-------------|--------------|--------------|--------------|--------------|--------------|
| a) 9 | b) 10 | c) 11 | d) 35 | e) 36 | f) 37 |
|-------------|--------------|--------------|--------------|--------------|--------------|

7. Suma elementelor aflate pe diagonala principală a matricei **a**, cu **5** linii și **5** coloane numerotate de la **0** la **4**, ale cărei elemente sunt actualizate în secvența de instrucțiuni de mai jos este:

Limbajul C++/C

```

n = 5;
for (i = 0; i < n; i++)

```

Limbajul Pascal

```

n := 5;
for i := 0 to n - 1 do

```

```

{
  for (j = 0; j < n; j++)
  {
    a[i][j] = (n - i) * n - j;
  }
}

```

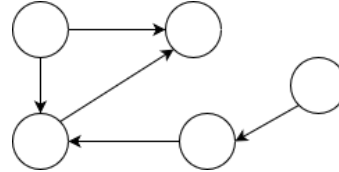
```

begin
  for j := 0 to n - 1 do
  begin
    a[i,j] := (n-i)*n - j
  end
end

```

a) 15 b) 20 c) 35 d) 55 e) 65 f) 70

8. O variantă care poate corespunde șirului gradelor interne ale vârfurilor grafului orientat alăturat este:



- a) (2, 1, 1, 1, 0) b) (1, 1, 1, 1, 0)
c) (2, 1, 0, 2, 0) d) (2, 0, 2, 2, 0)
e) (2, 0, 0, 3, 0) f) (2, 0, 1, 1, 0)
9. Un algoritm Backtracking generează **ultimele două** soluții **pilo** și **poli**, având ca date de intrare cuvântul **poli**. O variantă care poate reprezenta descrierea algoritmului este:
- a) Algoritmul generează în ordine invers lexicografică anagramele cuvântului citit.
b) Algoritmul generează în ordine lexicografică anagramele cuvântului citit.
c) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au vocale pe poziții alăturate.
d) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au consoane pe poziții alăturate.
e) Algoritmul generează în ordine lexicografică anagramele cuvântului citit care nu au vocale pe ultima poziție.
f) Algoritmul generează în ordine invers lexicografică anagramele cuvântului citit care nu au consoane pe ultima poziție.

10. Programul de mai jos afișează pe ecran textul **Politehnică 2020** dacă punctele de suspensie sunt înlocuite cu:

```

Limbajul C++/C
#include <stdio.h>
#include <string.h>
int main()
{
  char s[256], t[256];
  strcpy(s, "Politehnică 2020");
  ...
  strcpy(s + 4, t);
  puts(s);
  return 0;
}

```

```

Limbajul Pascal
var s, t: string;

begin
  s:='Politehnică 2020';
  ...
  s:=copy(s, 1, 4) + t;
  writeln(s)
end.

```

Limbajul C++/C

- a) `strcpy(t, strchr(s, ' '));` b) `strcpy(t, strcpy(s, ' '));`
c) `strcat(t, strchr(s, '2'));` d) `strcpy(t, strchr(s, " "));`

e) `strcat(t, strcpy(s, "2"));` f) `strcpy(t, strchr(s, "2"));`
 Limbajul Pascal
 a) `t:=copy(s, pos(' ', s), 5);` b) `t:=copy(s, copy(' ', s), 4);`
 c) `t:=s + pos('2', s);` d) `t:=copy(s, pos(" ", s), 5);`
 e) `t:=copy(s, copy("2", s), 4);` f) `t:=copy(s, pos("2", s), 5);`

11. Subprogramul **f** este definit mai jos. Valoarea returnată la apelul **f(24,34)** este:

Limbajul C++/C

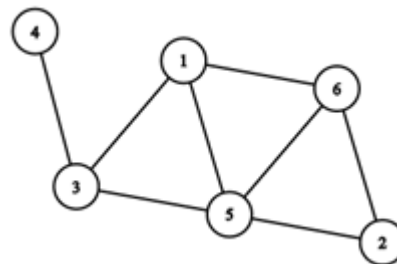
```
int f(int a, int b)
{
  int r;
  if (a >= b)
  {
    r = a;
  }
  else if (a % 10 == b % 10)
  {
    r = 2 + f(a + 1, b);
  }
  else if (a % 3 == b % 3)
  {
    r = 1 + f(a + 1, b - 1);
  }
  else
  {
    r = f(a, b - 2);
  }
  return r;
}
```

Limbajul Pascal

```
function f(a, b: integer):
integer;
var r: integer;
begin
  if a >= b then
  begin
    r := a;
  end
  else
  if a mod 10 = b mod 10 then
  begin
    r := 2 + f(a + 1, b)
  end
  else if a mod 3 = b mod 3
  then
  begin
    r := 1 + f(a + 1, b - 1)
  end
  else
  begin
    r := f(a, b - 2)
  end;
  f := r
end;
```

a) 30 b) 31 c) 32 d) 33 e) 34 f) 35

12. Numărul maxim de muchii care pot fi eliminate din graful neorientat alăturat astfel încât acesta să conțină cel puțin trei cicluri elementare distincte este:



a) 1 b) 6 c) 2 d) 4 e) 5 f) 3

13. Se generează în ordine lexicografică vectorii de tați corespunzători tuturor arborilor cu rădăcină având exact **6** noduri. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț elementar care unește rădăcina cu un alt nod. **A doua** soluție corespunzătoare unui arbore cu înălțimea **3** este:

- a) 0 1 2 3 1 1
- c) 0 1 1 1 2 6
- e) 0 1 1 1 1 2

- b) 0 1 1 1 2 5
- d) 0 1 1 1 3 5
- f) 0 1 1 1 4 5

14. Subprogramul **rad** de mai jos calculează și returnează cel mai mic număr care ridicat la pătrat este mai mare sau egal cu numărul natural reținut de **x** (partea întreagă superioară a lui radical din **x**) dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
int rad(int s, int d, int x)
{
    int rez, m;
    if (s == d)
    {
        rez = s;
    }
    else
    {
        m = (s + d) / 2;
        if (...)
        {
            rez = rad(s, m, x);
        }
        else
        {
            rez = rad(m + 1, d, x);
        }
    }
    return rez;
}
```

Limbajul Pascal

```
function rad(s,d,x: integer)
:integer;
var m, rez: integer;
begin
    if s = d then
        begin
            rez := s
        end
    else
        begin
            m := (s + d) div 2;
            if ... then
                begin
                    rez := rad(s, m, x)
                end
            else
                begin
                    rez := rad(m+1,d, x)
                end
            end;
        rad := rez
    end;
```

Limbajul C/C++

- a) $m * m == x$
- c) $m * m <= x$
- e) $m * m < x$

- b) $m * m >= x$
- d) $m * m > x$
- f) $m * m != x$

Limbajul Pascal

- a) $m * m = x$
- c) $m * m <= x$
- e) $m * m < x$

- b) $m * m >= x$
- d) $m * m > x$
- f) $m * m <> x$

15. Fie un tablou unidimensional **v** care reține **n** numere naturale: **v[0]**, **v[1]**, ..., **v[n-1]** și un număr întreg **t**. Secvența de instrucțiuni de mai jos are ca efect obținerea lungimii maxime **lmax** a unei subsecvențe **v[k]**, **v[k+1]**, ..., **v[k+lmax-1]** având suma elementelor mai mică sau egală cu **t** dacă punctele de suspensie sunt înlocuite cu:

Limbajul C++/C

```
s = 0;
j = 0;
lmax = 0;
```

Limbajul Pascal

```
s := 0;
j := 0;
lmax := 0;
```

```

for (i = 0; i < n; i++)
{
    s += v[i];
    while (j <= i && s > t)
    {
        ...
        j++;
    }
    if (i - j + 1 > lmax)
    {
        lmax = i - j + 1;
    }
}

```

```

for i := 0 to n-1 do
begin
    s := s + v[i];
    while(j<=i) and (s>t) do
    begin
        ...
        j := j + 1
    end;
    if i - j + 1 > lmax then
    begin
        lmax := i - j + 1
    end;
end;

```

Limbajul C++/C

- a) `s += v[j];`
- c) `s -= v[i];`
- e) `s += v[i];`

- b) `i--;`
- d) `s -= v[j];`
- f) `i++;`

Limbajul Pascal

- a) `s := s+v[j];`
- c) `s := s-v[i];`
- e) `s := s+v[i];`

- b) `i := i-1;`
- d) `s := s-v[j];`
- f) `i := i+1;`

6. Pentru funcția f definită mai jos, valoarea returnată de apelul $f(2019, 2347)$; este:

Limbajul C++/C

```
int f(int a, int b)
{
    int cif;
    if (a+b>0)
    {
        cif=a%10;
        if (cif<b%10)
            cif=b%10;
        return f(a/10, b/10)*10+cif;
    }
    return 0;
}
```

Limbajul Pascal

```
function f(a,b: integer):integer;
var cif:integer;
begin
    if (a+b>0) then
    begin
        cif:=a mod 10;
        if (cif < b mod 10) then
            cif:=b mod 10;
        f:=f(a div 10, b div 10)*10+cif
    end
    else
        f:=0
    end;
```

a) 349

b) 2017

c) 2349

d) 7102

e) 9432

f) 9743

7. Pentru tabloul unidimensional (4, 6, 14, 25, 61, 73, 82, 87, 95, 96, 98) numărul minim de elemente ale tabloului care trebuie verificate până este găsit elementul 82 este:

a) 7

b) 6

c) 5

d) 3

e) 2

f) 1

8. În urma executării programului de mai jos, variabila k are valoarea:

Limbajul C++/C

```
#include <iostream>
int k=1;
int f(int n)
{
    int k;
```

Limbajul Pascal

```
program p;
function f(n: integer):integer;
var k:integer;
begin
    k:=k+2;
```



```

    k=k+2;
    return k;
}
int main()
{
    k=f(k);
    return 0;
}

```

```

    f:=k
end;
var k:integer;
begin
    k:=1;
    k:=f(k)
end.

```

- a) 0 b) 1 c) 2
d) 3 e) nedefinită f) nicio valoare, programul are erori

9. Numărul elementelor care se găsesc strict deasupra diagonalei secundare a unui tablou bidimensional cu **20** de linii și **20** de coloane este:

- a) **180** b) **190** c) **200**
d) **210** e) **380** f) **400**

10. Problema *Turnurile din Hanoi*:

Se dau 3 tije. Pe prima tijă se găsesc discuri de diametre diferite, așezate în ordinea descrescătoare a diametrelor privite de jos în sus. Se cere să se mute discurile de pe prima tijă pe cea de-a doua, utilizând ca tijă intermediară cea de-a treia, respectând următoarele reguli:

- la fiecare pas se mută un singur disc;
- nu este permis să se așeze un disc cu diametrul mai mare peste un disc cu diametrul mai mic.

Numărul minim de mutări necesare rezolvării problemei *Turnurile din Hanoi* pentru **10** discuri este:

- a) **99** b) **100** c) **1022**
d) **1023** e) **1024** f) **1025**

11. Într-un graf orientat cu **56** de arce, în care oricare arc are extremități distincte și oricare două arce diferă prin cel puțin una dintre extremități, numărul minim de vârfuri este:

- a) **6** b) **7** c) **8**
d) **28** e) **56** f) **112**

12. Fie problema:

Se dau **$n-1$** numere distincte de la **1** la **n** ($1 < n < 10^5$). Se cere un algoritm care să determine numărul lipsă.

Fie algoritmi:

A₁: Se verifică prin câte o parcurgere prezența fiecărui număr de la **1** la **n** în șir.

A₂: Numărul lipsă este egal cu diferența dintre $[n \cdot (n + 1) / 2]$ și suma numerelor din șir.

A₃: Se sortează numerele și se determină pentru ce valori consecutive în șirul sortat

diferența este diferită de **1**.

A₄: Se sortează crescător numerele și se determină prima valoare din șirul sortat care este diferită de poziția în șir.

Este adevărat enunțul:

- a) Algoritmii A₁ și A₂ rezolvă problema pentru anumite date de intrare.
- b) Algoritmii A₂ este cel mai puțin eficient din punctul de vedere al timpului de executare.
- c) Algoritmii A₄ este cel mai eficient din punctul de vedere al timpului de executare.
- d) Algoritmii A₄ rezolvă problema doar dacă numărul lipsă este cel mai mare din șir.
- e) Cel puțin unul dintre algoritmi nu rezolvă problema.
- f) Doi dintre algoritmi nu diferă ca eficiență din punctul de vedere al timpului de executare.

13. Fie enunțurile:

E₁: orice graf neorientat conex **G** cu cel puțin **2** noduri, conține cel puțin un nod **k** care poate fi eliminat (și muchiile incidente cu el) obținându-se un subgraf **G'** conex;

E₂: un graf neorientat cu **n** (**n > 2**) noduri și **n** muchii conține cel puțin un ciclu;

E₃: orice arbore cu **n** (**n > 1**) noduri conține cel puțin două noduri cu gradul **1**.

Enunțurile adevărate sunt:

- a) doar E₁
- b) doar E₂
- c) doar E₁ și E₂
- d) doar E₁ și E₃
- e) doar E₂ și E₃
- f) E₁, E₂ și E₃

14. În urma executării unui program pentru generarea permutărilor elementelor unui șir de caractere ce conține duplicate, numărul de cuvinte distincte, anagrame ale cuvântului "**caracter**", este:

- a) **120**
- b) **2520**
- c) **5040**
- d) **10080**
- e) **20160**
- f) **40320**

15. Fie următoarele formule:

$$1. F(n) = \begin{cases} \left[2 \cdot F\left(\frac{n}{2} - 1\right) + F\left(\frac{n}{2}\right) \right] \cdot F\left(\frac{n}{2}\right), & n \text{ este par}, n > 3 \\ \left[F\left(\frac{n+1}{2}\right) \right]^2 + \left[F\left(\frac{n-1}{2}\right) \right]^2, & n \text{ este impar}, n > 2 \end{cases}$$
$$2. F(n) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^n + \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^n$$
$$3. F(n) = \frac{1}{\sqrt{5}} \cdot \left(\frac{1+\sqrt{5}}{2}\right)^{n/2} + \frac{1}{\sqrt{5}} \cdot \left(\frac{1-\sqrt{5}}{2}\right)^{n/2}$$

Știind că $F(1) = 1, F(2) = 1$, pentru a determina al **n**-lea (**n > 2**) termen din șirul lui Fibonacci (**1, 1, 2, 3, 5, 8, ...**) se poate folosi:

- a) niciuna dintre cele trei formule
- b) doar formula **1**
- c) doar formula **2**
- d) doar formula **1** și formula **2**
- e) doar formula **3**
- f) toate cele trei formule

