

# Test de reexaminare (20.02.2022) la Introducere în programare



\*Required

## Întrebări

Alegeți varianta corectă sau variantele corecte.

Fie `int i,s,S=1,x[5]={2,6,4,5,1}; for (s=0,i=1;i<3;i++) s+=*x+i*i;` La final `S+*(x+1)` va fi: \*

- ☐ 14
- ☐ niciuna din celelalte variante
- ☐ 12
- ☐ 7
- ☐ 11

Fie declarația `struct nod { int umr; int urm; } a, b;`. Care din următoarele atribuiri nu este sintactic corectă? \*

- ☐ `a.urm=b.urm;`
- ☐ niciuna din celelalte variante
- ☐ `a=b;`
- ☐ `a=b.nod;`
- ☐ `a.umr=b.urm;`



Care din următoarele declarații nu este sintactic corectă în C/C++? \*

- ☐ niciuna din celelalte variante
- ☐ `int y=5, x=sizeof(y+2);`
- ☐ `int y=5, x=sizeof(y);`
- ☐ `int y=5, x=sizeof(int);`
- ☐ `int y=5, x=sizeof(int+y);`

Ce va afișa următorul program? \*

```
#include <stdio.h>
int main()
{
    int s=0, i=5, j=4;
    for (i=1;i<=3;i++)
        s=s+2;
    for (j=i+1;j<=7;j++)
        s=s+2;
    printf("%d",s);
    return 0;
}
```

Your answer

---

Presupunând că a,b\_c și d sunt variabile de tip int, care din următoarele atribuiri nu este corectă? \*

- ☐ `a=++d==4;`
- ☐ niciuna din celelalte variante
- ☐ `a=15.0+d++`
- ☐ `a=a+7.0*(d%5);`
- ☐ `a=+-2+b_c-d;`



Fie un vector  $x$  cu  $n$  numere întregi (numerotate de la 1 la  $n$ ). Care din următoarele secvențe de program va afișa un singur DA, dacă toate elementele din vector sunt identice între ele, respectiv un singur NU, în alt caz? \*

- ☐ niciuna din celelalte variante
- ☐ `ok=1; for (i=1;i<n;i++) if (x[i]==x[i+1]) ok=0; if (ok==0) cout<<"DA"; else cout<<"NU";`
- ☐ `for (i=1;i<n;i++) if (x[i]==x[i+1]) cout<<"DA"; else cout<<"NU";`
- ☐ `for (i=1;i<n;i++) if (x[i]!=x[i+1]) ok=1; else ok=0; if (ok) cout<<"NU"; else cout<<"DA";`
- ☐ `for (i=1;i<n;i++) if (x[i]!=x[i+1]) cout<<"NU"; else cout<<"DA";`
- ☐ `ok=1; for (i=1;i<n;i++) if (x[i]!=x[i+1]) ok=0; if (ok) cout<<"DA"; else cout<<"NU";`

Fie un vector  $x$  cu  $n$  numere întregi (numerotate de la 1 la  $n$ ) și un număr  $a$  întreg. Care din următoarele secvențe de program va afișa un singur DA, dacă  $a$  se află în vectorul  $x$  (o dată sau de mai multe ori), respectiv un singur NU, dacă  $a$  nu se află în vectorul  $x$ ? \*

- ☐ `ok=1; for (i=1;i<=n;i++) if (x[i]!=a) ok=0; if (!ok) cout<<"NU"; else cout<<"DA";`
- ☐ `k=0; for(i=1;i<=n;i++) if (x[i]==a) k++; if (k>0) cout<<"DA"; else cout<<"NU";`
- ☐ `ok=0; for (i=1;i<=n;i++) if (x[i]!=a) cout<<"NU"; else ok=1; if (ok) cout<<"DA";`
- ☐ niciuna din celelalte variante
- ☐ `for (i=1;i<=n;i++) if (x[i]==a) cout<<"DA"; else cout<<"NU";`
- ☐ `ok=0; for (i=1;i<=n;i++) if (x[i]==a) ok=1; if (ok) cout<<"DA"; else cout<<"NU";`

Fie  $a, b, c > 0$ . Pentru a exprima că  $a, b, c$  pot fi lungimile laturilor unui triunghi echilateral, putem scrie: \*

- ☐ `a==b && !(c!=a)`
- ☐ niciuna din celelalte variante
- ☐ `a==b && b==c`
- ☐ `a==b || b==c || c==a`
- ☐ `a<b+c && !(b>=c+a || c>=a+b)`
- ☐ `a=b && a=c && c=b`



Care din următoarele afirmații este adevărată? \*

- ☐ O declarație de forma `int x-5.2;` este permisă în C/C++.
- ☐ Variabilele și funcțiile reprezintă același concept.
- ☐ niciuna din celelalte variante
- ☐ Variabilele globale își pot modifica valoarea într-o funcție.
- ☐ Expresia `(16>3) & (3<=10)` este corectă sintactic.

Care expresie are valoarea 1 dacă și numai dacă  $x$  aparține reuniunii de intervale  $[-3,-1] \cup [1,3]$  \*

- ☐ `x>=-3 && x<=-1 && x>=1 && x<=3`
- ☐ `!(x<-3 || x>-1) || !(x<1 || x>3)`
- ☐ niciuna din celelalte variante
- ☐ `!(x<-3 && x>3 && x>-1 || x<1)`

Fie declarația `struct nod { int inf; nod * st, *dr; } *a, b;`. Care din următoarele atribuiri nu sunt sintactic corecte? \*

- ☐ `(*a).inf=b.inf;`
- ☐ `a=b;`
- ☐ `a->st=&b;`
- ☐ `a->st=b.dr;`
- ☐ niciuna din celelalte variante



Ce afișează programul următor? \*

```
#include <stdio.h>
int main()
{
    int s=0, i=5, j=4;
    for (i=1;i<=3;i++)
        s=s+2;
    for (j=i+1;j<=7;j++)
        s=s+2;
    printf("%d",s);
    return 0;
}
```

Your answer

---

Fie: float x,y=2; x=3; x=5,y; La final, valoarea lui x va fi: \*

- ☐ 5.2
- ☐ niciuna din celelalte variante
- ☐ 3
- ☐ 5
- ☐ 5.3



Ce afișează programul următor \*

```
#include <iostream>
using namespace std;
int main()
{
    int i=2, j=1, k=1, z=1;
    for (; i<=10; i++)
        while (j<10)
        {
            j++;
            do
            {
                k++;
                z++;
            }
            while (k<=10) ;
        }
    cout<<z;
    return 0;
}
```

Your answer

---



Răspundeți la întrebarea din imagine \*

Fie funcția **int f(int x, int &y)** și declarația **int a=6,y=82;** în funcția **main**.

Care din următoarele apeluri ale funcției **f**, din funcția **main**, nu este sintactic corect?

- ☐ `a=f(y,a)==f(a,y);`
- ☐ `y=a+f(a+1+1+1+1+y,a);`
- ☐ `y=f(y+a,y);`
- ☐ `y=a+f(a+y+1+1,y+a+1);`
- ☐ niciuna din celelalte variante

Putem găsi într-un arbore binar un nod 5 care are fiul stâng 6 și fiul drept 4? \*

- ☐ uneori da, alteori nu
- ☐ nu
- ☐ da
- ☐ niciuna din celelalte variante



Care va fi valoarea afișată de următorul program? \*

```
#include <iostream>
using namespace std;
int main()
{
    int i=10,s=3,x[5]={2,3,4,5,1};
    for (int s=0,i=1;i<3;i++) s+=*x+i*i;
    cout<<s+i;
    return 0;
}
```

Your answer

---

Fie  $n$  un număr natural. Care din următoarele secvențe de cod afișează DA dacă numărul este prim, respectiv NU, dacă numărul nu este prim? \*

- ☐ `ok=0; for (d=2;d<n;d++) if (x%d!=0) ok=1; if (ok) cout<<"DA"; else cout<<"NU";`
- ☐ `ok=1; for (d=2;d<=n/2;d++) if (x%d==0) ok=0; if (ok) cout<<"DA"; else cout<<"NU";`
- ☐ `ok=1; for (d=2;d*d<=n;d++) if (x%d==0) ok=0; if (ok) cout<<"DA"; else cout<<"NU";`
- ☐ `ok=0; for (d=1;d<=n;d++) if (x%d==0) ok++; if (ok==2) cout<<"DA"; else cout<<"NU";`
- ☐ niciuna din celelalte variante
- ☐ `ok=1; for (d=2;d*d<n;d++) if (x%d==0) ok=0; if (ok) cout<<"DA"; else cout<<"NU";`





Care este valoarea returnată de funcția F pentru parametrul x=20? \*

```
unsigned F(unsigned x)
{ unsigned c=0;
  while (x!=1) {
    if (x%2!=17%4)
      x/=2;
    else
      x=x*3+1;
    c++;
  }
  return c; }
```

Choose ▼

Fie un vector x cu n numere întregi (numerotate de la 1 la n). Care din următoarele secvențe de program va determina valoarea cea mai mică din x? \*

- ☐ for (i=1;i<=n-1;i++) if (x[i]>x[i+1]) { aux=x[i]; x[i]=x[i+1]; x[i+1]=aux; } cout<<x[n];
- ☐ niciuna din celelalte variante
- ☐ minim=10000; for (i=1;i<=n;i++) if (x[i]<minim) minim=x[i]; cout<<minim;
- ☐ minim=x[1]; for (i=2;i<=n;i++) if (x[i]<minim) minim=x[i]; cout<<minim;
- ☐ minim=-1; for (i=1;i<=n;i++) if (x[i]<minim) minim=x[i]; cout<<minim;
- ☐ minim=x[1]; for (i=1;i<=n;i++) if (x[i]<minim) minim=x[i]; cout<<minim;



Care din următoarele afirmații nu este corectă? \*

- ☐ Dacă variabila i are valoarea 2, atunci i rămâne 2 după operația i+=0;
- ☐ Operatorii de deplasare pe biți sunt asociativi de la stânga la dreapta.
- ☐ ! (22<3 || 2>x) este echivalentă cu (22>=3) && (2<=x)
- ☐ Datele de tip char se reprezintă pe 8 bytes.
- ☐ niciuna din celelalte variante

Back

Submit

Clear form

Never submit passwords through Google Forms.

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

