



1. Precizati care dintre urmatoarele sechente de cod interschimba variabilele a si b:

A.

```
void interschimbare1(int &a, int &b){
    int aux = a;
    a = b;
    b = aux;
}
```

B.

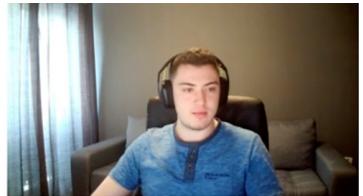
```
void interschimbare2(int &a, int &b){
    a = a + b;
    b = a - b;
    a = a - b;
}
```

C.

```
void interschimbare3(int &a, int &b){
    a = a / b;
    b = a * b;
    a = a * b;
}
```

D.

```
void interschimbare4(int &a, int &b){
    a = b;
    b = a;
}
```

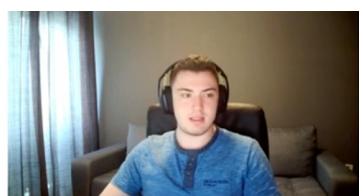


2. Se da urmatorul subprogram:

```
int suma(int n){
    if(n == 1)
        return 0;
    return (1 - n % 2) * n * (n - 1) + suma(n - 1);
}
```

Precizati care dintre informatii sunt adevarate:

- A. Pentru $n = 6$, se returneaza 44
- B. Pentru n – par, $\text{suma}(n)$ si $\text{suma}(n + 1)$ returneaza aceiasi valoare
- C. Pentru n – impar, $\text{suma}(n)$ si $\text{suma}(n + 2)$ returneaza acelasi lucru ca si $\text{suma}(n + 1)$
- D. Suma calculeaza $2 * 1 + 4 * 3 + \dots + n * (n - 1)$, pentru orice $n \geq 6$



$$(x > 5) \&\& (x < 20) \mid\mid (x \neq y)$$

3. Precizati care dintre urmatoarele expresii sunt echivalente cu cea de mai sus:

- A. $!(x \leq 5) \mid\mid (x < 20) \mid\mid (x \neq y)$
- B. $!(x \leq 5 \mid\mid x \geq 20) \mid\mid !(x == y)$
- C. $!(x \leq 5 \mid\mid (x \geq 20) \&\& (x == y))$
- D. $(5 > x) \&\& (20 < x) \mid\mid (x \neq y)$



4. Un fișier Excel conține n înregistrări numerotate de la 1 la n . Aceste înregistrări un fișier Word în care înregistrările se vor aranja în câte r rânduri și c coloane pe fiecare pagină (cu excepția primei și ultimei pagini). Pe prima pagină a documentului Word, datorită prezenței unui antet, numărul de rânduri este r_1 , $r_1 < r$ (numărul de rânduri prezent pe prima pagina este mai mic).

Înregistrările vor fi aranjate în fișierul Word pe fiecare pagină de sus în jos pe fiecare coloană, coloanele fiind completate de la stânga la dreapta: dacă prima înregistrare de pe o pagină are numărul de ordine i , înregistrarea cu numărul de ordine $(i + 1)$ va fi prezentă sub ea, iar înregistrarea cu numărul de ordine $(i + r)$ va fi prima înregistrare de pe coloana 2 de pe pagina respectivă și.a.m.d.

Pentru $n = 5000$, $r = 43$, $r_1 = 14$ și $c = 5$, unde în documentul Word se va regăsi înregistrarea cu numărul de ordine $i = 3245$?

- A. Pagina 15, coloana 4, randul 36
- B. Pagina 15, coloana 4, randul 35
- C. Pagina 16, coloana 4, randul 36
- D. Pagina 16, coloana 4, randul 35

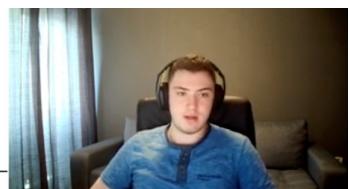


5. Precizați care dintre urmatoarele expresii logice verifică corect dacă numărul N are ultima cifră 3, 5 sau 7:

- A. $N \% 10 == 3 \mid\mid N \% 10 == 5 \mid\mid N \% 10 == 7$
- B. $N \% 3 == 0 \mid\mid N \% 5 == 0 \mid\mid N \% 7 == 0$
- C. $N \% 2 == 1 \&\& (N \% 10 != 1 \mid\mid N \% 10 != 9)$
- D. $N \% 2 != 0 \&\& !(N \% 10 == 1 \mid\mid N \% 10 == 9)$

6. Precizați care dintre următorii algoritmi pot fi implementați într-o complexitate $O(n * \log(n))$:

- A. Algoritmul care calculează numărul minim de subsiruri strict crescătoare în care poate fi partitionat un sir.
- B. Algoritmii de sortare QuickSort și MergeSort.
- C. Algoritmul de verificare a apartenenței a n elementelor la un sir cu n elemente ordonate crescător.
- D. Toate răspunsurile sunt corecte.



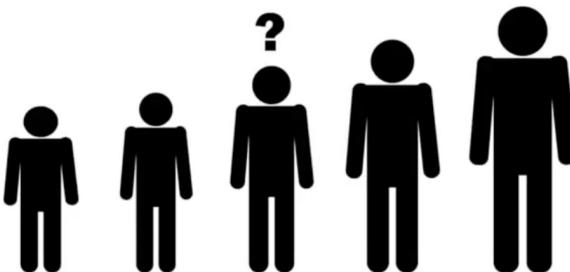
7. Se consideră următoarea secvență de cod în care n este un număr natural și $c[]$ – vectorul caracteristic:

```
void ce_face(int n){  
    c[1] = c[0] = 1;  
    for(int i = 2; i * i <= n; ++i)  
        if(c[i] == 0)  
            for(int j = i; i * j <= n; ++j)  
                c[i * j] = 1;  
}
```

Care dintre următoarele afirmații sunt adevărate?

- A. Algoritmul stochează în vectorul c , toate numerele prime.
- B. Algoritmul marchează cu 1 în vectorul c toate numerele prime.
- C. Algoritmul marchează cu 1 în vectorul c toate numerele care au cel puțin un divizor propriu.
- D. Algoritmul parcurge în ordine numerele prime.

8. Copilul mijlociu

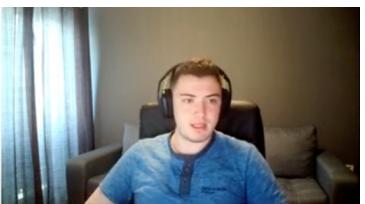


Copilul mijlociu

- A. Liviu
- B. Diana
- C. Marius
- D. Laura

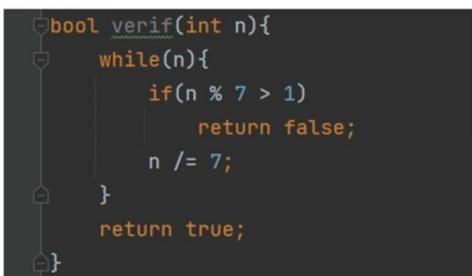
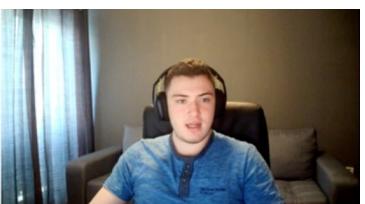


Marius este mai mic decât Liviu, dar mai mare decât Vlad. Laura este mai mare decât Diana, care la rândul ei este mai mare decât Marius. Liviu este mai mare decât Laura. Cine este copilul mijlociu?



9. Utilizând metoda backtracking se generează toate grupele de accesorii multimea {cască, costum de înot, cleme, dopuri de urechi, plută, mască de snorkeling, ochelari, snorkel}. Accesoriile au prețurile următoare, exprimate în lei: cască - 40, costum de înot - 400, cleme - 30, dopuri de urechi - 30, plută - 10, mască de snorkeling - 400, ochelari - 70, snorkel - 60. Într-o grupă accesoriile sunt distincte, nu contează ordinea lor și costă, în total, exact 500 de lei. Primele trei soluții generate sunt, în această ordine: (cască, costum de înot, cleme, dopuri de urechi), (cască, costum de înot, snorkel), (cască, cleme, dopuri de urechi, mască de snorkeling). A cincea soluție generată este:

- A. (cască, mască de snorkeling, snorkel)
- B. (costum de înot, cleme, plută, snorkel)
- C. (cleme, mască de snorkeling, ochelari)
- D. (dopuri de urechi, plută, mască de snorkeling, snorkel)



10. Care afirmații legate de aceasta secvență de cod sunt false?

- A. Algoritmul efectueaza descompunerea in baza 7 a numarului n.
- B. Algoritmul returneaza true daca si numai daca n contine doar cifrele 0 si 1 in scrierea sa in baza 7.
- C. Algoritmul verifică daca numarul n poate fi scris ca suma de puteri distincte ale lui 7.
- D. Algoritmul returneaza false pentru orice numar care nu este multiplu de 7.



```

void aranjare(int a[], int n){
    int p = 1;
    for(int i = 1; i <= n; ++i)
        if(a[i] % 2 == 0) {
            a[i] = a[i] + a[p];
            a[p] = a[i] - a[p];
            a[i] = a[i] - a[p];
            p = p + 1;
        }
}

```

11. Precizati care este efectul subalgoritmului alaturat:
- Sorteaza in ordine crescatoare elementele vectorului a.
 - Rearanjeaza elementele vectorului in asa fel incat cele impare sa fie inaintea celor pare.
 - Algoritmul nu face nimic concluziv.
 - Rearanjeaza elementele in asa fel incat cele pare sa fie inaintea celor impare.



```

int ce_face1(int a[], int n){
    int max = -1000000000;
    int s = 0;
    for(int i = 1; i <= n; ++i){
        s += a[i];
        if(s > max)
            max = s;
        if(s < 0)
            s = 0;
    }
    return s;
}

int ce_face2(int a[], int n){
    int s[n + 1] = {0};
    for(int i = 1; i <= n; ++i)
        s[i] = s[i-1] + a[i];
    int max = -1000000000;
    for(int i = 1; i <= n; ++i)
        for(int j = i; j <= n; ++j)
            if(s[j] - s[i-1] > max)
                max = s[j] - s[i-1];
    return max;
}

```

12. Se considera cele 2 subprogram alaturate. Precizati care dintre urmatoarele informatii sunt adevărate:
- Cele 2 secvențe de cod nu returnează același lucru pentru seturi de date identice.
 - Ambele secvențe de cod identifică secvența de suma maximă dintr-un vector.
 - Cele 2 secvențe de cod au complexitate similară.
 - Secvențele de cod calculează același lucru, utilizând algoritmi diferiți, de complexități diferite, primul fiind mai eficient decât al doilea.



```

int f(int n){
    int j = n;
    while(j > 1){
        int i = 1;
        while(i < j * j)
            i *= 4;
        j /= 3;
    }
    return j;
}

```

13. Precizati in care dintre urmatoarele clase de complexitate se incadreaza algoritmul alaturat:
- $O(\log_2^2 n^3)$
 - $O(\log_{12} n^2)$
 - $O(\log_3 n \cdot \lg n^2)$
 - $O(\log_3 n \log_4 n)$



```

int f(int n, int k){
    if(n == k || k == 0)
        return 1;
    else{
        cout << "Zecelainfo";
        return f( n: n-1, k: k-1 ) + f( n: n-1, k );
    }
}

```

14. Precizati de cate ori se va afisa mesajul “Zecelainfo” in urma executiei secventei de cod de mai sus:

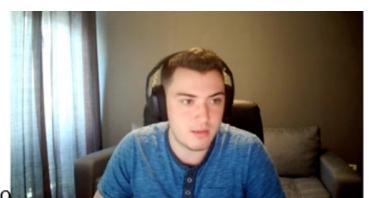
- A. $C_n^k \cdot 2 + 1$
- B. $C_n^k \cdot 2 - 1$
- C. $A_n^k - 1$
- D. $C_n^k - 1$



15. In legatura cu grila anteroara. Care dintre urmatoarele afirmatii sunt adevarate?

- A. Numarul total de pasi pe care ii va face subprogramul este $C_n^k \cdot 2 + 1$.
- B. Pentru $n = 10$ si $k = 5$, numarul total de pasi va fi 503.
- C. Pentru $n = 15$ si $k = 3$, numarul total de pasi va fi 907.
- D. Numarul total de pasi pe care ii va face subprogramul este $C_n^k - 1$

* Prin pas a se inteleaga un apel al functiei (autoapel + apelul initial).



```

void f(int a[], int n, int b[], int m, int c[], int &k){
    int i=1, j=1;
    k=0;
    while(i<=n && j<=m){
        if(a[i]<b[j])
            c[++k]=a[i++];
        else if(a[i]>b[j])
            c[++k]=b[j++];
        else i++, j++;
    }
    while(i<=n)
        c[++k]=a[i++];
    while(j<=m)
        c[++k]=b[j++];
}

```

16. Se da secventa de cod a si b reprezinta cate o multime matematica cu cardinalele egale cu n si, respective, m . Precizati ce calculeaza aceasta secventa de cod:

- A. INTERSECTIA ($A \cap B$) celor 2 multimi si o stocheaza in vectorul c cu k elemente.
- B. DIFERENTA ($A \setminus B$) celor 2 multimi si o stocheaza in vectorul c cu k elemente.
- C. DIFERENTA SIMETRICA ($(A \setminus B) \cup (B \setminus A)$) a celor 2 multimi si o stocheaza in vectorul c su k elemente.
- D. REUNIUNEA celor 2 multimi si o stocheaza in vectorul c cu k elemente.

```

void f1(int n, int& prod, int d = 2, int p = 0){
    if(n == 1)
        return ;
    if(n % d != 0){
        if(d * d > n)
            f1(n, &prod, d, p);
        else f1(n, &prod, d+1, p);
    }
    else{
        p++;
        n /= d;
        if(n % d != 0)
            prod *= (p + 1);
        f1(n, &prod, d, p);
    }
}

```

```

void f2(int n, int& cnt, int d = 1){
    if(d * d >= n)
        cnt++;
    return ;
}
if(n % d == 0)
    cnt+=2;
f2(n, &cnt, d+1);

```



17. Se consideră subprogramele f_1 și f_2 , definite anterior. Subprogramul f_1 primește ca și parametrii, 2 variabile, n , un număr natural și un $prod$, initializat înainte de apel cu valoarea 1. Subprogramul f_2 primește ca și parametrii, 2 variabile, n , un număr natural și un cnt , initializat înainte de apel cu valoarea 0. Precizați care dintre următoarele afirmații sunt corecte:

- A. Pentru același parametru de intrare n , valorile $prod$ și cnt vor avea aceleasi valori după execuția celor 2 funcții.
- B. Cele 2 secvențe de cod calculează același lucru.
- C. Prima funcție are o complexitate mai bună decât a 2-a.
- D. Prima funcție calculează produsul puterilor factorilor primi din descompunerea în factori primi ai lui n , iar cea de-a doua funcție calculează numarul de divizori ai lui n .

```

int suma1(int a[][101], int n){
    int sum = 0;
    for(int i = 2; i <= n - 1; ++i)
        for(int j = max(n - i + 1, i) + 1; j <= n; ++j)
            sum += a[i][j];
    return sum;
}

int suma2(int a[][101], int n){
    int sum = 0;
    for(int i = 1; i <= n / 2; ++i)
        for(int j = i + 1; j < n - i + 1; ++j)
            sum += a[i][j];
    return sum;
}

```

```

int suma3(int a[][101], int n){
    int sum = 0;
    for(int i = 2; i < n; ++i)
        for(int j = 1; j < min(n-i+1, i); ++j)
            sum += a[i][j];
    return sum;
}

int suma4(int a[][101], int n){
    int sum = 0;
    for(int i = n / 2 + 2; i <= n; ++i)
        for(int j = n - i + 2; j < i; ++j)
            sum += a[i][j];
    return sum;
}

```

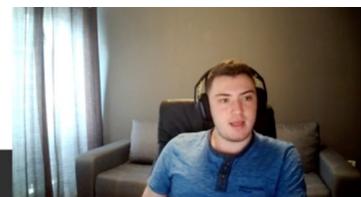




18. Se dă cele 4 funcții de deasupra, fiecare calculând niste sume intr-o matrice patratică cu latura egală cu n . Precizați care dintre următoarele afirmații sunt corecte:

- A. Funcția `suma1` și `suma3` calculează suma elementelor din ZONA 2.
- B. Funcția `suma4` calculează suma elementelor din ZONA 3.
- C. Funcția `suma2` și `suma3` calculează suma elementelor din ZONA 1.
- D. Nicio afirmație nu este corectă.

* În desenul de deasupra puteți vedea cele 4 zone delimitate de cele 2 diagonale.

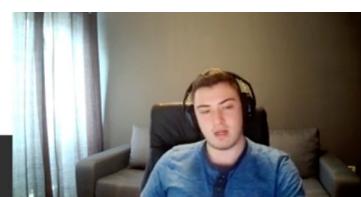


```

bool verif(int a[], int st, int dr){
    if(st == dr)
        return true;
    int mij = (st + dr) / 2;
    return verif(a, st, mij) && verif(a, mij + 1, dr) && a[mij] <= a[mij + 1];
}
  
```

19. Precizați care dintre următoarele informații sunt false despre subprogramul `verif`:

- A. Complexitatea acestui program este $O(\log_2 n)$.
- B. Subprogramul returnează adevarat dacă cele 2 elemente din mijlocul sirului sunt în ordine crescătoare.
- C. Subprogramul utilizează conceptul de Divide et Impera.
- D. Dacă sirul conține un singur element se returnează false.



```

int f(int a[], int n, int i){
    if(i > n)
        return 0;
    return max(f(a, n, i + 1), f(a, n, i + 2) + a[i]);
}
  
```

20. Se consideră subprogramul de mai sus. Precizați care este efectul subprogramului dacă se apelează cu următorii parametrii: $a[] = \{2, 4, 7, 3, 10, 1, 4, 8, 3\}$, $n = 9$, $k = 1$:

- A. 42
- B. 29
- C. 27
- D. 35





```
int cb1(int a[], int n, int val){  
    int st = 1, dr = n, poz = 0;  
    while(st <= dr)  
    {  
        int mij = (st + dr) / 2;  
        if(a[mij] <= val){  
            poz = max(poz, mij);  
            st = mij + 1;  
        }  
        else dr = mij - 1;  
    }  
    if(poz == 0)  
        return -1;  
    return poz;  
}
```

```
int cb2(int a[],  
        int st = 1,  
        while(st <= dr)  
        {  
            int mij = (st + dr) / 2;  
            if(a[mij] >= val){  
                poz = min(poz, mij);  
                dr = mij - 1;  
            }  
            else st = mij + 1;  
        }  
        if(poz == n+1)  
            return -1;  
        return poz;  
}
```



21-22. Sa dau cele 2 functii de mai sus: cb1 si cb2 care primesc ca parametrii un sir de numere naturale ordonate crescator si o valoare numar natural.

21. Precizati care dintre urmatoarele informatii sunt adevarate:

- A. Cele 2 subprograme cauta binar valoarea val in sirul a.
- B. Cele 2 subprograme returneaza aceiasi valoare in cazul in care valoarea val apare in sir.
- C. Cele 2 subprograme nu returneaza aceiasi valoare decat daca valoarea val apare in mod unicat in sir.
- D. In cazul in care valoarea val nu apare in sir, subprogramele returneaza -1.

22. Referitor la cele 2 sechente de cod, precizati in care dintre urmatoarele situatii este necesar sa le folosim pe ambele pentru a ajunge la un rezultat corect:

- A. Pentru a verifica daca o valoare apare intr-un sir ordonat crescator.
- B. Pentru a numara de cate ori apare o valoare intr-un sir.
- C. Pentru a vedea cate valori dintr-un sir, apartin unui interval de numere.
- D. Pentru a identifica cel mai apropiat numar ca valoare care apare in sir, de o anumita valoare care nu apare neaparat in sir.



```
void afisare(int n){  
    if(n == 0)  
        return ;  
    cout << n << ' ' ;  
    afisare( n : n - 1 );  
    cout << n - 1 << ' ' ;  
    afisare( n : n - 1 );  
    cout << n << ' ' ;  
}
```

23. Se considera suprogramul alaturat. Precizati care dintre urmatoarele afirmatii sunt adevarate referitor la acest subprogram:

- A. Pentru n = 3, se afiseaza un sir cu 20 de elemente.
- B. Pentru n = 3, se afiseaza un sir cu 21 de elemente.
- C. Oricare ar fi n, numarul de valori afisate este multiplu de 3.
- D. Pentru n = 2, se afiseaza: 2 1 0 1 0 1 0 1 2



```
int f1(int a, int b){  
    if(b == 0)  
        return 1;  
    int val = f1(a, b: b / 2);  
    if(b % 2 == 1)  
        return a * val * val;  
    else return val * val;  
}  
  
int f2(int n, int i, int j){  
    int p = f1(a: 10, b: i);  
    return n / p * p * 10 + j * p + n % p;  
}
```

24. Se da secevta de cod alaturata. Precizati pentru care dintre urmatoarele seturi de date, se afiseaza ceea ce scrie la fiecare?
- A. f2(123, 2, 3) -> 1323
 - B. f2(123, 2, 3) -> 1233
 - C. f1(4, 3) -> 16
 - D. f2(32468234, 6, 9) -> 329468234

Raspunsuri:

- | | |
|----------|-------------|
| 1. A, B | 13. A, C, D |
| 2. A, B | 14. D |
| 3. B, C | 15. B |
| 4. C | 16. C |
| 5. A, D | 17. A, B |
| 6. D | 18. B |
| 7. C | 19. A, B, D |
| 8. B | 20. C |
| 9. B | 21. C |
| 10. A, D | 22. B, C, D |
| 11. D | 23. B, C |
| 12. A | 24. A, D |