

1. Subprogramul **sub**, cu trei parametri, primește prin intermediul parametrilor:
- **v** un tablou unidimensional cu cel mult 100 de componente ce memorează numere întregi de cel mult 4 cifre fiecare;
 - **n** un număr natural nenul mai mic sau egal cu 100 ce reprezintă numărul efectiv de componente ale tabloului primit prin intermediul parametrului **v**;
 - **a** un număr întreg cu cel mult 4 cifre.

Subprogramul **sub** returnează numărul componentelor tabloului primit prin intermediul parametrului **v** ale căror valori sunt strict mai mici decât valoarea parametrului **a**.

Exemplu: pentru valorile $n=5$, $v=(1, 21, 9, 21, 403)$, $a=20$ ale parametrilor, în urma apelului, subprogramului **sub** va returna valoarea 2.

a) Scrieți definiția completă a subprogramului **sub**.

b) Să se scrie un program C/C++ care să citească de la tastatură un număr natural nenul n ($n \leq 100$) și n numere întregi, fiecare având cel mult 4 cifre, și care, folosind apeluri utile ale subprogramului **sub**, să afișeze pe ecran mesajul **DA** dacă oricare două dintre cele n numere întregi citite sunt distincte două câte două, sau mesajul **NU** în caz contrar. **Exemplu:** pentru $n=6$ și cele n numere citite de la tastatură: 47 183 69 8 134 -56 se va afișa pe ecran mesajul **DA**

2. Un număr n se numește **extraprim** dacă atât el, cât și orice număr obținut prin permutarea cifrelor lui n , sunt numere prime. De exemplu, numărul 113 este un număr **extraprim** deoarece 113, 311, 131 sunt numere prime.

a) Scrieți definiția completă a unui subprogram **f**, cu un parametru, subprogram care:

- primește prin intermediul parametrului **a** un număr natural cu cel mult 3 cifre ($a > 1$)
- returnează suma tuturor exponenților din descompunerea în factori primi a valorii parametrului **a**.

Exemplu: pentru $a=90$ subprogramul va returna valoarea 4, deoarece $a=2 \cdot 3^2 \cdot 5$ și $1+2+1=4$.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural n , $2 \leq n \leq 999$ și, folosind apeluri utile ale subprogramului **f**, verifică dacă n este un număr **extraprim**. În caz afirmativ, programul afișează pe ecran mesajul **DA**, în caz contrar afișând mesajul **NU**.

3. Subprogramul **cif**, cu doi parametri, primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre și prin intermediul parametrului **b** o cifră; subprogramul returnează numărul de apariții ale cifrei **b** în scrierea numărului **a**.

Exemplu: pentru $a=125854$ și $b=5$, subprogramul va returna valoarea 2.

a) Scrieți definiția completă a subprogramului **cif**. (4p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural n cu exact 8 cifre și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului **cif**, cel mai mare număr palindrom ce poate fi obținut prin rearanjarea tuturor cifrelor numărului n . Dacă nu se poate obține un palindrom din toate cifrele numărului n , programul va afișa pe ecran numărul 0. Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă.

Exemplu: dacă $n=21523531$ atunci se va afișa pe ecran numărul 53211235, iar dacă $n=12272351$ atunci se va afișa pe ecran numărul 0. (6p.)

4. Subprogramul **cif**, cu doi parametri, primește prin intermediul parametrului **a** un număr natural cu cel mult 8 cifre și prin intermediul parametrului **b** o cifră; subprogramul returnează numărul de apariții ale cifrei **b** în scrierea numărului **a**.

Exemplu: pentru $a=125854$ și $b=5$, subprogramul va returna valoarea 2.

a) Scrieți definiția completă a subprogramului **cif**. (4p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural n cu **exact 8** cifre, fiecare cifră fiind nenulă, și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului `cif`, cel mai mic număr palindrom ce poate fi obținut prin rearanjarea tuturor cifrelor numărului n . Dacă nu se poate obține un palindrom din toate cifrele numărului n , programul va afișa pe ecran numărul 0. Un număr natural este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor sale în ordine inversă. **Exemplu:** dacă $n=21523531$ atunci se va afișa pe ecran numărul 12355321, iar dacă $n=12272351$ atunci se va afișa pe ecran numărul 0. **(6p.)**

5. Subprogramul `f`, cu un parametru:

- primește prin intermediul parametrului a un număr natural cu cel mult 8 cifre ($a > 1$)
- returnează cel mai mic divizor prim al valorii parametrului a .

Exemplu: pentru valoarea 45 a parametrului a , subprogramul va returna valoarea 3 deoarece $a=32 \cdot 5$, iar cel mai mic divizor prim al său este 3.

a) Scrieți definiția completă a subprogramului `f`. **(4p.)**

b) Scrieți un program C/C++ care să citească de la tastatură un număr natural nenul n ($n \leq 100$) și apoi un șir de n numere naturale de cel mult 8 cifre fiecare, toate numerele din șir fiind strict mai mari decât 1. Folosind apeluri utile ale subprogramului `f`, programul va determina și va afișa pe ecran toate numerele prime din șirul citit. Numerele determinate se vor afișa pe ecran, separate prin câte un spațiu, în ordine crescătoare a valorii lor. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**. **Exemplu:** pentru $n=7$, șirul: 1125, 2, 314, 101, 37, 225, 12 pe ecran se va afișa: 2 37 101 **(6p.)**

6. Se consideră șirul definit de relația de recurență alăturată:

a) Scrieți definiția completă a unui subprogram `sub`, care primește prin intermediul singurului său parametru n un număr natural de maximum 8 cifre, și care returnează cel mai mare termen al șirului `f` care este mai mic sau cel mult egal cu n .

Exemplu: dacă $n=83$ atunci subprogramul va returna valoarea 80. **(4p.)**

b) Scrieți un program C/C++ care citește de la tastatură un număr natural s ($s \leq 10000000$) și determină scrierea lui s ca sumă de termeni distincți ai șirului dat folosind apeluri utile ale subprogramului `sub`. Numerele astfel determinate se vor scrie pe ecran, pe aceeași linie, separate prin câte un spațiu. **Exemplu:** dacă valoarea citită de la tastatură este 63, se va afișa: 40 20 3 **(6p.)**

7. Subprogramul `sub` primește prin intermediul parametrilor:

– n și m două numere naturale ($1 < n < 100$, $1 < m < 100$)

– a și b două tablouri unidimensionale, fiecare având componente numere naturale de maximum patru cifre, **ordonate crescător**; tabloul a conține n numere pare, iar tabloul b conține m numere impare. Subprogramul va afișa pe ecran, în ordine crescătoare, separate prin câte un spațiu, un șir format dintr-un număr maxim de elemente care aparțin cel puțin unuia dintre tablouri, astfel încât orice două elemente aflate pe poziții consecutive să fie de paritate diferită. **Exemplu:** pentru $n=5$, $m=3$ și tablourile $a=(2, 4, 8, 10, 14)$ și $b=(3, 5, 11)$, subprogramul va afișa 2 3 4 5 8 11 14 sau 2 3 4 5 10 11 14.

a) Scrieți definiția completă a subprogramului `sub`, alegând pentru rezolvare un algoritm eficient din punctul de vedere al timpului de executare. **(6p.)**

b) Descrieți succint, în limbaj natural, algoritmul pe baza căruia a fost scris subprogramul de la punctul **a)**, explicând în ce constă eficiența metodei utilizate. **(4p.)**

8. Scrieți definiția completă a subprogramului **sub** cu trei parametri: **n** (număr natural, $5 < n \leq 30000$), **a** și **b**; subprogramul furnizează prin intermediul parametrilor **a** și **b** cele mai mari două numere **prime distincte** mai mici decât **n**. **Exemplu:** dacă **n= 28** la apelul **subprogramului** se va furniza prin parametrul **a** valoarea 23 și prin parametrul **b** valoarea 19.

9. **a)** Scrieți doar antetul funcției **sum** care primește ca parametru un număr natural nenul **x** cu maximum 9 cifre și returnează suma divizorilor numărului **x**.

Exemplu: **sum(6)** are valoarea 12 (=1+2+3+6). **(3p.)**

b) Să se scrie un program **C/C++** care citește de la tastatură un număr natural **n** ($0 < n < 25$) și apoi **n** numere naturale nenule cu maximum 9 cifre fiecare. Programul calculează, folosind apeluri ale funcției **sum**, și afișează pe ecran câte numere prime conține șirul citit. **Exemplu:** pentru **n=5** și valorile 12 3 9 7 1 se va afișa pe ecran valoarea 2 (în șirul dat există două numere prime și anume 3 și 7). **(7p.)**

10. Se consideră subprogramul **P** care are doi parametri:

– **n**, prin intermediul căruia primește un număr natural de cel mult 9 cifre

– **c**, prin intermediul căruia primește o cifră.

Subprogramul va furniza tot prin intermediul parametrului **n** numărul obținut din **n** prin eliminarea tuturor aparițiilor cifrei **c**. Dacă, după eliminare, numărul nu mai conține nicio cifră sau conține doar cifre 0, rezultatul returnat va fi 0.

a) Scrieți doar antetul subprogramului **P**. **(2p.)**

b) Pe prima linie a fișierului text **BAC.IN** se găsesc, separate prin câte un spațiu, mai multe numere naturale de cel mult 9 cifre fiecare. Scrieți programul **C/C++** care citește numerele din acest fișier, utilizând apeluri ale subprogramului **P** elimină toate cifrele impare din fiecare dintre aceste numere și apoi scrie în fișierul text **BAC.OUT** numerele astfel obținute, separate prin câte un spațiu. Dacă un număr din fișierul **BAC.IN** nu conține nicio cifră pară nenulă, acesta nu va mai apărea deloc în fișierul de ieșire. **(8p.)**

Exemplu: dacă fișierul **BAC.IN** conține numerele 25 7 38 1030 45127 0 35 60 15, atunci **BAC.OUT** va avea conținutul: 2 8 42 60.

11. Scrieți definiția completă a subprogramului **multiplu** care are 3 parametri: **a**, prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000, **n**, numărul efectiv de elemente ale tabloului și **k**, un număr natural ($k \leq 9$).

Subprogramul returnează numărul de elemente din tablou care sunt multipli ai numărului **k** și au ultima cifră egală cu **k**. **Exemplu:** dacă **n=6**, **a=(9, 273, 63, 83, 93, 123)**, iar **k=3**, subprogramul va returna valoarea 4.

12. Scrieți definiția completă a subprogramului **interval** care are doi parametri **a** și **n**, prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000 și respectiv numărul efectiv de elemente din tabloul unidimensional. Subprogramul returnează numărul de elemente din tabloul unidimensional care aparțin intervalului închis determinat de primul și respectiv ultimul element al tabloului. **Exemplu:** dacă tabloul are 6 elemente și este de forma (12, 27, 6, 8, 9, 2), subprogramul va returna valoarea 5.

13. Scrieți definiția completă a subprogramului **count** care are doi parametri, **a** și **n**, prin care primește un tablou unidimensional cu maximum 100 de numere reale și respectiv numărul efectiv de elemente din tablou. Subprogramul returnează numărul de elemente din tabloul **a** care sunt mai mari sau cel puțin egale cu media aritmetică a tuturor elementelor din tablou.

Exemplu: dacă tabloul are 6 elemente și este de forma (12, 7.5, 6.5, 3, 8.5, 7.5), subprogramul va returna valoarea 4 (deoarece media tuturor elementelor este 7.5 și numerele subliniate sunt cel puțin egale cu această medie).

14. Subprogramul **aranjare** are doi parametri: **a** prin care primește un tablou unidimensional cu maximum 100 de numere reale nenule și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile negative să se afle pe primele poziții, iar valorile pozitive în continuarea celor negative. Ordinea în cadrul secvenței de elemente pozitive, respectiv în cadrul secvenței de elemente negative, poate fi oricare. Tabloul modificat va fi furnizat tot prin intermediul parametrului **a**.

Exemplu: dacă tabloul are 6 elemente și este de forma (12, -7.5, 6.5, -3, -8, 7.5), după apel, acesta ar putea fi: (-7.5, -3, -8, 12, 6.5, 7.5). Scrieți definiția completă a subprogramului **aranjare**.

15. Subprogramul **nule** are doi parametri: **a**, prin care primește un tablou unidimensional cu maximum 100 de numere întregi, cu cel mult 4 cifre fiecare și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile nule să se afle la sfârșitul tabloului. Ordinea în cadrul secvenței de elemente nenule poate fi oricare. Tabloul modificat este furnizat tot prin parametrul **a**. **Exemplu:** dacă $n=6$, $a=(12, 0, 0, -3, -8, 0)$, după apel, acesta ar putea fi: $a=(12, -3, -8, 0, 0, 0)$. Scrieți definiția completă a subprogramului **nule**.

16. Scrieți definiția completă a unui subprogram **i_prim** care primește prin singurul său parametru, **n**, un număr natural din intervalul [2, 30000] și returnează diferența minimă $p_2 - p_1$ în care p_1 și p_2 sunt numere prime și $p_1 \leq n \leq p_2$.

Exemplu: dacă $n=20$ atunci $i_prim(n)=4$, valoare obținută pentru $p_1=19$ și $p_2=23$

17. a) Scrieți definiția completă a unui subprogram, **nz**, cu un parametru întreg **n** ($0 < n \leq 2000$), care returnează numărul zerourilor de la sfârșitul numărului **n!**.

b) Scrieți programul C/C++ care citește de la tastatură un număr natural **k** ($0 < k \leq 3$) și determină, folosind apeluri ale subprogramului **nz**, cel mai mic număr natural **n** pentru care **n!** are cel puțin **k** zerouri la sfârșit. Numărul determinat se afișează pe ecran.

18. a) Scrieți definiția completă a subprogramului **shift** care primește prin intermediul parametrului **n** o valoare naturală nenulă ($n \leq 100$), iar prin intermediul parametrului **x**, un tablou unidimensional cu maximum 100 de componente. Fiecare componentă a acestui tablou este un număr întreg care are cel mult 4 cifre. Subprogramul permută circular cu o poziție spre stânga primele **n** elemente ale tabloului **x** și furnizează tabloul modificat tot prin parametrul **x**. **Exemplu:** dacă înainte de apel $n=4$ și $x=(1, 2, 3, 4)$, după apel $x=(2, 3, 4, 1)$.

b) Scrieți un program C/C++ care citește de la tastatură o valoare naturală nenulă **n** ($n \leq 100$), apoi cele **n** elemente ale unui tablou unidimensional **x**. Programul va inversa ordinea elementelor tabloului **x** folosind apeluri utile ale subprogramului **shift** și va afișa pe ecran, separate prin câte un spațiu, elementele tabloului rezultat în urma acestei prelucrări.

Exemplu: dacă se citește pentru **n** valoarea 5, iar tabloul **x** este (1, 2, 3, 4, 5) programul va determina ca **x** să devină (5, 4, 3, 2, 1).

19. a) Scrieți definiția completă a subprogramului **p** care primește prin intermediul parametrului **n** un număr natural nenul ($n \leq 100$), iar prin intermediul parametrului **x** un tablou unidimensional cu **n** componente întregi, de maximum patru cifre fiecare. Subprogramul furnizează prin intermediul parametrului **mini** valoarea minimă din tabloul **x**, prin intermediul parametrului **maxi** valoarea maximă din **x**, iar prin intermediul parametrului **sum** suma elementelor din tabloul **x**.

b) Scrieți un program C/C++ care citește de la tastatură o valoare naturală nenulă **n**,

($3 \leq n \leq 100$), apoi cele n elemente distincte ale unui tablou unidimensional x . Fiecare dintre aceste elemente este un număr natural având cel mult patru cifre. Folosind apeluri utile ale subprogramului p , programul calculează și afișează pe ecran media aritmetică a elementelor care ar rămâne în tabloul x dacă s-ar elimina valoarea minimă și valoarea maximă din tablou. Valoarea afișată va avea cel mult 3 cifre după virgulă. **Exemplu:** dacă se citește pentru n valoarea 5, iar pentru tabloul x valorile (1, 9, 4, 8, 5), programul va afișa una dintre valorile 5.667 sau 5.666.

20. Subprogramul f primește prin intermediul parametrului n un număr natural nenul ($1 \leq n \leq 9$), iar prin intermediul parametrului a , un tablou unidimensional care conține n valori naturale, fiecare dintre acestea reprezentând câte o cifră a unui număr. Astfel, a_0 reprezintă cifra unităților numărului, a_1 cifra zecilor etc. Subprogramul furnizează prin parametrul k o valoare naturală egală cu numărul obținut din cifrele pare reținute în tabloul a sau valoarea -1 dacă în tablou nu există nicio cifră pară. Scrieți definiția completă a subprogramului f . **Exemple:** dacă subprogramul se apelează pentru $n=6$ și pentru tabloul a având valorile (2, 3, 5, 6, 4, 1), parametrul k va furniza valoarea 462. Dacă subprogramul se apelează pentru $n=4$ și pentru a reținând valorile (0, 0, 1, 1), k va furniza valoarea 0. Dacă subprogramul se apelează pentru $n=3$ și pentru a reținând valorile (3, 7, 1), k va furniza valoarea -1 .

21. a) Scrieți definiția completă a subprogramului $sterge$, care primește prin cei 4 parametri v, n, i, j :

- v , un tablou unidimensional cu maximum 100 de elemente întregi din intervalul $[-1000, 1000]$

- n , un număr natural reprezentând numărul de elemente din tabloul v

- i și j două valori naturale cu $1 \leq i \leq j \leq n$

și elimină din tabloul v elementele v_i, v_{i+1}, \dots, v_j actualizând valoarea parametrului n .
Tabloul modificat este furnizat tot prin parametrul v .

b) Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul n ($1 \leq n \leq 100$) și pe următoarea linie n numere întregi din intervalul $[-1000; 1000]$, separate prin câte un spațiu. Scrieți un program C/C++ care citește din fișierul **NUMERE.IN** numărul natural n , construiește în memorie un tablou unidimensional v cu cele n numere întregi aflate pe linia a doua în fișier și utilizează apeluri utile ale subprogramului $sterge$ pentru a elimina din tablou un număr minim de elemente astfel încât să nu existe două elemente alăturate cu aceeași valoare. Elementele tabloului obținut se afișează pe ecran, separate prin câte un spațiu.

Exemplu: Dacă fișierul **NUMERE.IN** are conținutul:

12

10 10 2 2 19 9 9 9 9 15 15 15 atunci se afișează 10 2 19 9 15.

22. Scrieți definiția completă a subprogramului $nreal$ cu doi parametri x și y , numere naturale din intervalul $[1; 1000]$ ce returnează un număr real cu proprietatea că partea sa întreagă este egală cu x , iar numărul format din zecimalele sale, în aceeași ordine, este egal cu y . **Exemplu:** pentru $x=12$ și $y=543$, subprogramul returnează valoarea 12.543.

23. a) Scrieți definiția completă a unui subprogram $primul$, care

- primește prin singurul său parametru, a , o valoare naturală din intervalul $[2, 10000]$

- returnează o valoare naturală reprezentând cel mai mic divizor al numărului a mai mare strict decât 1.

b) Fișierul text **NUMERE.IN** conține pe prima linie un număr natural nenul ($1 \leq n \leq 100$) și pe următoarea linie n numere naturale din intervalul $[2, 10000]$ separate prin câte un spațiu. Un număr natural n se numește „aproape prim” dacă este egal cu produsul a două numere

prime distincte. De exemplu, numărul 14 este „aproape prim” pentru că este egal cu produsul numerelor prime 2 și 7. Scrieți un program C/C++ care determină, folosind apeluri utile ale subprogramului `primul`, cel mai mare număr „aproape prim” de pe linia a doua a fișierului `NUMERE.IN`. În cazul în care există un astfel de număr se afișează pe ecran mesajul `DA`, urmat de numărul determinat, iar în caz contrar mesajul `NU`. **Exemplu:** dacă fișierul `NUMERE.IN` are conținutul:

6

100 14 21 8 77 35

atunci se afișează pe ecran `DA 77` pentru că numărul 77 este cel mai mare dintre numerele „aproape prime” din fișier ($14=7*2$, $21=7*3$, $77=7*11$, $35=7*5$).

24. Scrieți definiția completă a subprogramului `multipli`, cu trei parametri `a, b, c` ($a \leq b$), numere naturale din intervalul $[1, 10000]$ ce returnează numărul multiplilor lui `c` din intervalul $[a; b]$. **Exemplu:** pentru $a=10$, $b=27$, $c=5$ subprogramul returnează valoarea 4.

25. Scrieți definiția completă a subprogramului `suma`, care primește prin cei 4 parametri, `v, n, i, j`:

- `v`, un tablou unidimensional cu maximum 100 de elemente întregi din intervalul $[-1000, 1000]$, numerotate de la 1 la `n`;

- `n`, un număr natural reprezentând numărul de elemente din tabloul `v`;

- `i` și `j`, două valori naturale cu $1 \leq i \leq j \leq 100$

și returnează suma elementelor `v1, ..., vi-1, vj+1, ..., vn` din tabloul `v`.

26. Scrieți definiția completă a subprogramului `suma` care primește ca parametru un tablou unidimensional `x` cu cel mult 100 de elemente, numere reale, un număr natural `n` ce reprezintă numărul efectiv de elemente ale tabloului `x` ($n \leq 100$), și un număr natural `m` ($n \geq m$). Subprogramul returnează suma obținută din cele mai mici `m` elemente ale tabloului `x`.

27. Scrieți definiția completă a subprogramului `nr_prim` care are ca parametru un număr natural `x` și returnează cel mai mic număr prim, strict mai mare decât `x`.

Exemplu: pentru $x=25$ subprogramul returnează numărul 29, iar pentru $x=17$ valoarea returnată va fi 19.

28. Subprogramul `sum3` primește prin parametrul `x` un tablou unidimensional, cu cel mult 100 de elemente, numere întregi cu cel mult 4 cifre fiecare, iar prin parametrul `n` un număr natural ce reprezintă numărul efectiv de elemente ale tabloului `x` ($n \leq 100$). Scrieți definiția completă a subprogramului `sum3` care returnează suma elementelor tabloului care sunt divizibile cu 3. Dacă tabloul nu conține nicio valoare divizibilă cu 3, subprogramul va returna 0.

29. a) Subprogramul `max` primește ca parametru un tablou unidimensional `x` cu cel mult 100 de elemente numere întregi, care sunt, în ordine, termenii unei progresii aritmetice și un număr natural `n`, care reprezintă dimensiunea tabloului. Scrieți definiția completă a subprogramului `max` care returnează cel mai mare termen al progresiei aritmetice. Alegeți un algoritm de rezolvare eficient din punct de vedere al timpului de executare.

b) Explicați în limbaj natural metoda utilizată justificând eficiența acesteia.

c) Pe prima linie a fișierului `numere.txt` se află un număr natural `n` ($n \leq 100$), iar pe următoarele `n` linii, câte `n` numere întregi cu cel mult 4 cifre fiecare. Scrieți programul C/C++ care citește din fișier datele existente, determină liniile din fișier pe care s-au memorat în ordine termenii unei progresii aritmetice și afișează pe ecran, folosind apeluri ale

subprogramului **max** cel mai mare număr (diferit de cel situat pe prima linie) din fișier, care în plus este termenul unei progresii aritmetice.

Exemplu: dacă fișierul **numere.txt** are conținutul:

```
5
5 7 3 1 9
-9 -7 -5 -3 -1
2 5 8 14 11
50 40 30 20 10
18 17 16 15 14
```

se va afișa 50, deoarece progresiile aritmetice sunt:

(-9 -7 -5 -3 -1), (50 40 30 20 10) și (18 17 16 15 14)

30. a) Scrieți definiția completă a unui subprogram **sum** care primește prin parametrul **x** un număr natural de cel mult 4 cifre și returnează suma divizorilor numărului **x**, diferiți de 1 și de el însuși.

Exemplu: dacă **x=10** se va returna valoarea 7 ($7=2+5$).

b) Scrieți programul C/C++ care citește de la tastatură un număr natural **n** ($0 < n < 100$), apoi **n** numere naturale (cu cel mult 4 cifre fiecare). Programul determină, folosind apeluri utile ale subprogramului **sum**, pentru fiecare număr natural citit, suma divizorilor săi proprii și afișează pe ecran sumele determinate, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu.

Exemplu: dacă **n=5** și numerele citite sunt 10 2 33 6 11 valorile afișate pe ecran vor fi: 0 0 5 7 14 deoarece suma divizorilor lui 10 este 7, suma divizorilor lui 2 este 0, suma divizorilor lui 33 este 14, suma divizorilor lui 6 este 5, suma divizorilor lui 11 este 0.

31. Subprogramul **cifra** primește prin intermediul parametrului **a** un număr natural cu cel mult 4 cifre și returnează ultima cifră pară a sa. Dacă numărul nu conține cifre pare, subprogramul returnează valoarea -1. De exemplu, dacă **a=8345**, subprogramul va returna 4.

a) Să se scrie definiția completă a subprogramului **cifra**.

b) Pe prima linie a fișierului **bac.in** se află un număr natural nenul **n** ($n \leq 15000$), iar pe a doua linie a fișierului se află un șir de **n** numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran, folosind apeluri utile ale subprogramului **cifra**, cel mai mare număr care se poate forma cu ultimele cifre pare ale fiecărui element, dacă acestea există. Alegeți o metodă de rezolvare eficientă ca timp de executare. Dacă toate numerele de pe a doua linie a fișierului au numai cifre impare, programul va afișa mesajul **NU EXISTA**.

Exemplu: dacă fișierul **bac.in** are conținutul alăturat,

```
7
369 113 2 0 33 1354 42
```

pe ecran se va afișa: 64220

c) Descrieți succint în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 - 4 rânduri).

32. a) Scrieți doar antetul subprogramului **nrdiv**, care primește prin intermediul parametrului **x** un număr natural nenul cu cel mult 4 cifre, și returnează numărul de divizori primi ai lui **x**.

b) Pe prima linie a fișierului **bac.in** se află un număr natural nenul **n** ($n \leq 1000$), iar pe a doua linie a fișierului se află un șir format din **n** numere naturale nenule, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care citește numerele din fișier și care afișează pe ecran, folosind apeluri utile ale subprogramului **nrdiv**, prima și ultima valoare din șirul celor **n** numere citite, care au un număr par de divizori primi. Numerele afișate vor fi separate printr-un spațiu. **Exemplu:** dacă fișierul **bac.in** are conținutul alăturat,

7

30 105 20 140 7 10 5
pe ecran se va afișa: 20 10

33. Scrieți în limbajul C/C++ definiția completă a subprogramul `Del` care are doi parametri: `x`, un număr întreg de cel mult 9 cifre, și `y`, un număr natural nenul de o cifră. Subprogramul determină eliminarea tuturor cifrelor lui `x` mai mari strict decât `y` și furnizează numărul obținut tot prin intermediul parametrului `x`. Dacă toate cifrele lui `x` sunt mai mari strict decât `y`, atunci `x` va primi valoarea -1.

Exemplu: dacă `x=37659` și `y=6`, după apel `x=365`, iar `y=6`.

34. Se consideră subprogramul `inter`, cu doi parametri: `x` și `y` (numere întregi formate din cel mult patru cifre fiecare); subprogramul interschimbă valorile a două variabile transmise prin intermediul parametrilor `x` și `y`.

a) Scrieți în limbajul C/C++ definiția completă a subprogramului `inter`.

b) Pe prima linie a fișierului `bac.in` se află un număr natural nenul $n \leq 1000$, iar pe a doua linie a fișierului se află un șir de `n` numere naturale nenule, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult 4 cifre. Scrieți un program C/C++ care afișează pe ecran, în ordine crescătoare, numerele aflate pe a doua linie a fișierului. Numerele vor fi afișate pe o singură linie, iar între două numere se va lăsa un spațiu. Se vor folosi apeluri utile ale subprogramului `inter`.

35. Se consideră subprogramul `pr`, care primește prin intermediul parametrului `a` un număr natural nenul cu cel mult 9 cifre și returnează 1 dacă numărul este prim și 0 în caz contrar.

a) Scrieți numai antetul subprogramului `pr`.

b) Considerăm un număr natural nenul $n > 99$ cu cel mult 9 cifre. Din `n` se obține un șir de valori prin eliminarea succesivă a ultimei cifre, apoi a ultimelor două cifre, apoi a ultimelor trei cifre etc., până ce se obține un număr de două cifre, ca în exemplu. Să se realizeze un program C/C++ care citește de la tastatură numărul `n` și care, folosind apeluri utile ale subprogramul `pr`, afișează pe ecran, separate prin câte un spațiu, doar valorile prime din șirul numerelor obținute din `n`, prin procedeul descris mai sus.

Exemplu: pentru `n=193124` se obține șirul de valori 19312, 1931, 193, 19. din care se vor afișa pe ecran doar valorile 1931 193 19 (nu neapărat în această ordine).

36. a) Scrieți definiția completă a subprogramului `sdiv` care primește prin intermediul parametrului `y` un număr natural cu cel mult 6 cifre și returnează suma tuturor divizorilor numărului `y`.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` ($n < 10000$) și care, folosind apeluri ale subprogramului `sdiv` verifică dacă suma divizorilor lui `n` este un număr prim. În caz afirmativ, programul va afișa pe ecran mesajul `Da` și în caz contrar va afișa mesajul `NU`. **Exemplu:** dacă `n=206`, atunci programul va afișa: `NU` ($1+2+103+206=312$, iar 312 nu este un număr prim).

37. Scrieți un subprogram `DIST`, cu doi parametri, care primește prin intermediul parametrului `a` un tablou unidimensional cu cel mult 100 de elemente, numere naturale de cel mult 4 cifre fiecare, și prin intermediul parametrului `n` un număr natural nenul, $n < 100$, ce reprezintă numărul de elemente din tablou. Subprogramul returnează valoarea 1 dacă toate elementele tabloului `a` sunt distincte și dacă diferența absolută a oricăror două elemente vecine din tablou este diferită de 1, altfel returnând valoarea 0.

38. Scrieți definiția completă a unui subprogram **P**, cu trei parametri, care primește prin intermediul primului parametru, **a**, un tablou unidimensional de cel mult 100 de numere întregi, cu cel mult 4 cifre fiecare, prin intermediul celui de-al doilea parametru, **n**, numărul efectiv de elemente ale tabloului, iar prin parametrul **k**, un număr natural ($k < 101$) și returnează cea mai mare sumă cu **k** termeni care se poate obține adunând **k** elemente ale tabloului. **Exemplu:** dacă $n=6$ și $k=4$, iar șirul este format din elementele (5, 2, 5, 4, 1, 3), atunci la apel se va returna 17.

39. Scrieți definiția completă a unui subprogram **P** cu doi parametri, care primește prin intermediul primului parametru, **n**, un număr natural nenul ($1 \leq n \leq 100$) și prin intermediul celui de-al doilea parametru, **a**, un tablou unidimensional cu elementele numerotate de la 1 la **n**, numere întregi de cel mult 4 cifre fiecare. Subprogramul returnează suma tuturor numerelor **impere** aflate pe poziții **pare** din tablou.

Exemplu: dacă $n=6$, iar șirul **a** este format din elementele (3, 12, 7, 1, 4, 3), atunci la apel se va returna 4.

40. Scrieți definiția completă a unui subprogram **P**, cu doi parametri, **a** și **b**, numere naturale cu cel mult 4 cifre fiecare, care afișează pe ecran, separate prin câte un spațiu, numerele aflate în intervalul închis determinat de valorile **a** și **b**, care sunt pătratele unor numere prime. **Exemplu:** pentru $a=40$ și $b=1$ se vor afișa valorile: 4 9 25 (nu neapărat în această ordine).

41. **a)** Scrieți în limbajul C/C++ doar antetul subprogramului **cifre**, care prin intermediul parametrului **nr** primește un număr natural de cel mult 9 cifre și furnizează prin intermediul parametrilor **nc** și **sc** numărul de cifre și respectiv suma cifrelor din scrierea lui **nr**.

b) Scrieți programul C/C++ care citește de la tastatură un număr natural **n** ($10 \leq n \leq 109$) și verifică, folosind apeluri utile ale subprogramului **cifre**, dacă în scrierea în baza 10 a lui **n** se găsește cel puțin o cifră care să fie media aritmetică a celorlalte cifre din componența lui **n**. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz contrar.

Exemplu: pentru $n=27989$ programul va afișa mesajul **DA**, deoarece în scrierea lui **n** apare cifra 7 care este media aritmetică a celorlalte cifre din scrierea lui **n**: $7 = (2+9+8+9) / 4$. Pentru $n=7351$ se va afișa mesajul **NU**.

42. **a)** Scrieți în limbajul C/C++ doar antetul unui subprogram **cif**, care primește prin intermediul primului parametru, **nr**, un număr natural cu cel mult 9 cifre și furnizează prin intermediul celui de-al doilea parametru, **s**, suma cifrelor din scrierea lui **nr**.

b) Scrieți programul C/C++ care citește de la tastatură un număr natural **n** ($0 < n < 25$), apoi un șir de **n** numere naturale nenule cu cel mult 9 cifre fiecare și care afișează pe ecran, separate prin câte un spațiu, numerele din șir care au suma cifrelor maximă, folosind apeluri utile ale subprogramului **cif**. **Exemplu:** dacă pentru $n=8$ se citește șirul de numere 274 56018 354 8219 293 287 932 634 atunci, pe ecran, se afișează numerele 56018 8219.

43. Se consideră subprogramul **cmax** care prin parametrul **a** primește un număr natural nenul mai mic decât 30000, iar prin parametrul **b** furnizează cifra maximă din numărul **a**.

a) Scrieți, folosind limbajul C/C++, doar antetul subprogramului **cmax**.

b) Fișierul **bac.txt** conține cel mult 1000 numere naturale nenule, mai mici decât 30000 fiecare, separate prin câte un spațiu. Scrieți programul C/C++ care citește din fișierul **bac.txt** toate numerele și care determină cea mai mare cifră din scrierea lor și cel mai mic dintre numerele care conțin această cifră, folosind apeluri utile ale subprogramului **cmax**. Cifra și numărul determinate se vor afișa pe ecran, separate printr-un spațiu. **Exemplu:** dacă fișierul **bac.txt** conține valorile: 23 12 64 12 72 345 67 23 71 634 atunci pe ecran se afișează 7 67.

44. Se consideră subprogramul `divxy` care primește prin parametri `x` și `y` două valori întregi pozitive ($0 < x < 1000$ și $0 < y < 1000$) și returnează valoarea 1 dacă `y` este divizor al lui `x` sau `x` este divizor al lui `y` și returnează valoarea 0 în caz contrar.

a) Scrieți definiția completă a subprogramului `divxy`.

b) Scrieți un program C/C++ care citește de la tastatură trei numere naturale nenule `a`, `b` și `n`, cu cel mult 3 cifre fiecare și care afișează pe ecran toți divizorii lui `n` din intervalul închis determinat de `a` și `b` folosind apeluri utile ale subprogramului `divxy`. Intervalul închis determinat de `a` și `b` este `[a,b]` dacă $a < b$ sau `[b,a]` dacă $b \leq a$. Numerele afișate sunt separate prin câte un spațiu. Dacă nu există niciun astfel de număr se afișează mesajul `NU EXISTA`. **Exemplu:** pentru `a=85`, `b=10` și `n=40` se afișează: 10 20 40 (nu neapărat în această ordine).

45. Se consideră subprogramul `dist2`, care primește prin intermediul parametrilor `xa`, `ya` și respectiv `xb`, `yb`, coordonatele carteziene întregi (abscisă, ordonată) pentru două puncte din plan, `A` și respectiv `B`. Subprogramul returnează pătratul distanței dintre cele două puncte.

a) Scrieți în limbajul C/C++ definiția completă a subprogramului `dist2`.

b) Scrieți un program C/C++ care citește de la tastatură 8 valori întregi reprezentând coordonatele carteziene pentru patru puncte din plan și afișează mesajul `Da` dacă cele patru puncte pot fi vârfurile unui pătrat, iar în caz contrar afișează mesajul `Nu`, folosind apeluri utile ale subprogramului `dist2`. **Exemplu:** dacă coordonatele punctelor sunt cele alăturate

0 0

3 0

3 3

0 3

atunci se va afișa mesajul `Da`

46. Subprogramul `mult`, cu doi parametri, primește prin intermediul primului parametru, `n`, un număr natural nenul cu maximum trei cifre și prin intermediul celui de-al doilea parametru, `a`, un tablou unidimensional având `n` componente numere naturale cu cel mult 8 cifre fiecare. Subprogramul returnează valoarea `true` dacă cele `n` componente ale lui `a` pot forma o mulțime și returnează `false` în caz contrar.

a) Scrieți definiția completă a subprogramului `mult`.

b) Fișierul text `date.in` conține cel mult 400 de numere naturale având maximum 8 cifre fiecare. Scrieți un program C/C++ care, folosind apeluri utile ale subprogramului `mult`, afișează pe ecran valoarea maximă `k`, astfel încât primele `k` numere succesive din fișier să poată forma o mulțime. **Exemplu:** dacă fișierul `date.in` conține

16 17 8 31 8 2 10

atunci se va afișa 4 (deoarece primele patru numere din fișier pot forma o mulțime și acesta este cardinalul maxim posibil în condițiile impuse de enunțul problemei)

47. Se consideră subprogramul `cmmdc`, care primește prin intermediul a doi parametri, `a` și `b`, două numere naturale nenule, cu maximum 8 cifre fiecare, și returnează cel mai mare divizor comun al valorilor parametrilor `a` și `b`.

a) Scrieți definiția completă a subprogramului `cmmdc`.

b) Scrieți un program `Pascal` care citește de la tastatură un număr natural `n` ($n < 300$), și, cu ajutorul subprogramului `cmmdc`, determină numărul perechilor de valori naturale (a, b) , $1 < a < b < n$, cu proprietatea că `a` și `b` nu au niciun divizor comun în afară de 1. Numărul obținut se va afișa pe ecran. **Exemplu:** dacă se citește `n=6`, atunci se va afișa 6 (deoarece perechile $(2, 3)$, $(2, 5)$, $(3, 4)$, $(3, 5)$, $(4, 5)$, $(5, 6)$ satisfac condițiile din enunț).

48. Se consideră subprogramul `inv`, care primește prin intermediul primului parametru `a` un număr natural, cu minimum două cifre și maximum 8 cifre, și furnizează prin intermediul celui de-al doilea parametru, `b`, valoarea numărului natural format cu aceleași cifre ca și `a`, considerate în ordine inversă. De exemplu, pentru `a=3805`, după apel `b` va avea valoarea 5083., iar dacă `a=3800`, după apel `b` va avea valoarea 83.

a) Scrieți definiția completă a subprogramului `inv`.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` de minimum două și maximum 8 cifre și afișează pe ecran un număr palindrom cu valoarea cea mai apropiată de valoarea lui `n` citită. În cazul în care există două astfel de numere, se va afișa cel mai mic dintre ele. Se vor folosi apeluri utile ale subprogramului `inv`. Spunem că un număr natural `x` este palindrom dacă numărul format din cifrele lui `x` considerate de la stânga către dreapta este același cu numărul format din cifrele lui `x` considerate de la dreapta către stânga.

Exemplu: dacă `n=16`, atunci se afișează valoarea 11, dacă `n=126`, atunci se afișează 121, iar dacă `a=33`, atunci se afișează 33.

49. Se consideră subprogramul `cifre`, care primește prin intermediul primului parametru, `a`, un număr natural cu maximum 8 cifre nenule și returnează, prin intermediul celui de-al doilea parametrul `b`, cel mai mic număr care se poate forma cu toate cifrele distincte ale lui `a`.

a) Scrieți definiția completă a subprogramului `cifre`.

b) Se consideră fișierul text `date.in` ce conține pe prima linie un număr natural nenul `n` ($n \leq 100$), iar pe a doua linie `n` numere naturale, separate prin câte un spațiu, fiecare număr având maximum 8 cifre nenule. Scrieți un program C/C++ care citește toate numerele din fișierul text `date.in` și afișează pe ecran, despărțite prin câte un spațiu, numerele situate pe a doua linie a fișierului, formate numai din cifre distincte ordonate strict crescător, folosind apeluri utile ale subprogramului `cifre`. În cazul în care nu există niciun astfel de număr se va afișa valoarea 0.

Exemplu: dacă fișierul `date.in` are conținutul alăturat, 6

16 175 333 242477 321 269 atunci se vor afișa numerele: 16 269

50. Se consideră subprogramele

– `prim`, care primește prin intermediul unicului său parametrului `x` un număr natural nenul de cel mult 4 cifre și returnează valoarea 1 dacă `x` este un număr prim și 0 în caz contrar;
– `numar`, care primește prin intermediul parametrului `x` un număr natural nenul de cel mult 4 cifre și furnizează prin intermediul parametrului `nrp` numărul de numere prime mai mici decât `x`.

a) Scrieți numai antetul subprogramului `prim` și definiția completă a subprogramului `numar`.

b) Scrieți un programul C/C++ în care se citesc de la tastatură două numere naturale nenule de cel mult 4 cifre, `a` și `b` ($a < b$), și, prin apeluri utile ale subprogramului `numar`, se verifică dacă intervalul închis `[a, b]` conține cel puțin un număr prim. Progr. va afișa pe ecran, în caz afirmativ, mesajul `DA`, iar în caz contrar, mesajul `NU`.

51. Se consideră subprogramul `multiplu`, cu doi parametri, care:

– primește prin intermediul parametrilor `a` și `k` două numere întregi de cel mult 4 cifre;
– returnează cel mai mic multiplu al lui `k` mai mare sau egal cu `a`.

a) Scrieți numai antetul subprogramului `multiplu`.

b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură trei numere naturale nenule `x`, `y`, `z`, de cel mult 4 cifre fiecare, ($x \leq y$), și care, prin apeluri utile ale subprogramului `multiplu`, verifică dacă intervalul `[x, y]` conține cel puțin un multiplu al lui `z`. Programul va afișa pe ecran, în caz afirmativ, mesajul `DA`, iar în caz contrar mesajul `NU`.

52. Se consideră subprogramul `divizor`, care:

– primește prin intermediul parametrului `a` un număr natural strict mai mare decât 1, de cel mult 4 cifre;

– furnizează prin intermediul parametrului `d` cel mai mare divizor al lui `a` strict mai mic decât `a`.

a) Scrieți numai antetul subprogramului `divizor`.

b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură un număr natural nenul `x`, de cel mult 4 cifre și, prin apeluri utile ale subprogramului `divizor`, verifică dacă `x` este număr prim. Programul va afișa pe ecran în caz afirmativ mesajul `DA`, iar în caz contrar mesajul `NU`.

53. Se consideră subprogramul `radical`, care:

– primește prin intermediul parametrului `a`, un număr natural nenul de cel mult 4 cifre;

– furnizează prin intermediul parametrului `x` cel mai mare număr natural cu proprietatea că x^2 este mai mic sau egal cu `a`; de exemplu, dacă `a=20`, subprogramul va furniza prin `x` valoarea 4.

a) Scrieți numai antetul subprogramului `radical`.

b) Scrieți declarațiile de date și programul principal C/C++ care citește de la tastatură un număr natural nenul de cel mult 4 cifre, `n`, și prin apeluri utile ale subprogramului `radical`, verifică dacă `n` este pătrat perfect. Programul va afișa pe ecran în caz afirmativ mesajul `DA`, iar în caz contrar mesajul `NU`.

54. Subprogramul `cifra` primește prin parametrul `x` un număr real nenul pozitiv și furnizează prin parametrul `y` valoarea cifrei unităților părții întregi a lui `x`.

Exemplu: dacă `x=34.567`, după apel `y=4`.

a) Scrieți definiția completă a subprogramului `cifra`.

b) Scrieți un program C/C++ care citește de la tastatură două numere reale cu cel mult două zecimale, numere reprezentând mediile semestriale obținute de un elev. Programul stabilește, folosind apeluri utile ale subprogramului `cifra`, dacă cele două medii citite se află în aceeași categorie de medii sau nu. Precizăm că orice medie, în funcție de intervalul în care se încadrează, face parte din una dintre categoriile: `[3,3.99]`, `[4,4.99]`, `[5,5.99]`, `[6,6.99]`, `[7,7.99]`, `[8,8.99]` sau `[9,10]`. În cazul în care ambele medii fac parte din aceeași categorie, programul va afișa mesajul `Da`, altfel va afișa mesajul `Nu`.

55. Scrieți definiția completă a unui subprogram `fibonacci` cu doi parametri, `n` și `v`, care primește prin intermediul parametrului `n` un număr natural ($1 < n < 30$) și returnează prin intermediul parametrului `v` un tablou unidimensional care conține primii `n` termeni **impari** ai șirului lui Fibonacci (amintim că șirul lui Fibonacci este: `1, 1, 2, 3, 5, 8, 13, 21, ...`).

56. Subprogramul `verif` primește prin singurul său parametru, `x`, un număr natural nenul cu cel mult 9 cifre și returnează valoarea 1 dacă numărul conține cel puțin o secvență de 3 cifre impare alăturate și 0 în caz contrar.

Exemplu: dacă `x=7325972` se va returna valoarea 1.

a) Scrieți definiția completă a subprogramului `verif`. (6p.)

b) Scrieți un program C/C++ care citește de la tastatură un număr natural nenul `n` cu exact 6 cifre și, folosind apeluri utile ale subprogramului `verif`, verifică dacă `n` are primele trei cifre impare. Programul afișează pe ecran mesajul `Da` în caz afirmativ și mesajul `Nu` în caz contrar.

57. Subprogramul `diviz` primește prin intermediul parametrului `n` un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului `a`, un tablou unidimensional care conține `n` valori naturale nenule, fiecare dintre acestea având cel mult patru cifre. Elementele tabloului sunt numerotate de la 1 la `n`. Subprogramul returnează o valoare egală cu numărul de perechi (a_i, a_j) , $1 \leq i < j \leq n$, în care `ai` este divizor al lui `aj`, sau `aj` este divizor al lui `ai`. Scrieți definiția completă a subprogramului `diviz`, în limbajul C/C++. **Exemplu:** pentru `n=5` și `a=(4,8,3,9,4)` subprogramul returnează valoarea 4.

58. Subprogramul `sfx` primește prin singurul său parametru, `x`, un număr natural din intervalul `[100, 200000000]` și returnează valoarea 1 dacă ultimele trei cifre ale numărului sunt în ordine strict descrescătoare sau valoarea 0 în caz contrar.

Exemplu: dacă `x=24973` se va returna valoarea 1.

a) Scrieți definiția completă a subprogramului `sfx`.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` format din exact 6 cifre și verifică, utilizând apeluri ale subprogramului `sfx`, dacă acest număr are toate cifrele în ordine strict descrescătoare. Programul va afișa mesajul `Da` în caz afirmativ și mesajul `Nu` în caz contrar.

Exemplu: dacă `n=756543` se va afișa `Nu`, iar dacă `n=976532` se va afișa `Da`.

59. Să se scrie în limbajul C/C++ definiția completă a subprogramului `calcu1`, care primește prin intermediul parametrului `n` un număr natural nenul ($1 \leq n \leq 10000$), iar prin intermediul parametrului `a`, un tablou unidimensional care conține `n` valori naturale, fiecare dintre aceste valori având cel mult 9 cifre. Subprogramul returnează cel mai mare divizor comun al elementelor tabloului `a`. **Exemplu:** în urma apelului, pentru `n=5` și tabloul unidimensional `(12,36,48,6,60)` se va returna 6.

60. Să se scrie în limbajul C/C++ definiția completă a subprogramului `calcu1`, care primește prin intermediul parametrului `n` un număr natural nenul ($1 \leq n \leq 10000$), iar prin intermediul parametrului `a`, un tablou unidimensional care conține `n` valori naturale, fiecare dintre aceste valori având cel mult 9 cifre. Subprogramul returnează numărul de numere prime din tablou.

Exemplu: pentru `n=5` și tabloul unidimensional `(12,37,43,6,71)` în urma apelului se va returna 3.

61. a) Să se scrie definiția completă a subprogramului `calcu1`, care primește prin intermediul celor doi parametri ai săi două numere întregi, `n` și `k` ($1 \leq n \leq 100000000$ și $1 \leq k \leq 9$), și returnează cifra de rang `k` a numărului `n`. Rangul unei cifre este numărul său de ordine, numerotând cifrele de la dreapta la stânga; cifra unităților având rangul 1. Dacă numărul `k` este mai mare decât numărul de cifre ale lui `n`, atunci funcția return. valoarea -1. **Exemplu:** dacă `n=9243` și `k=3`, în urma apelului se va returna 2.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` cu cel mult 8 cifre. Programul va verifica, utilizând apeluri ale subprogramului `calcu1`, dacă orice cifră a lui `n` are rangul cifrei mai mare sau egal cu valoarea cifrei respective și va afișa mesajul `Da` în caz afirmativ și mesajul `Nu` în caz contrar.

Exemplu : pentru `n=4160` se va afișa `Nu`.

rang 4 3 2 1

cifră 4 1 6 0

62. **a)** Să se scrie definiția completă a subprogramului `ca1cu1`, care primește prin intermediul parametrului întreg `n` un număr natural de cel mult 9 cifre și returnează valoarea absolută a diferenței dintre numărul de cifre pare și numărul de cifre impare conținute de `n`.
Exemplu: dacă `n=92465`, în urma apelului se va returna valoarea 1 (2 cifre impare, 3 cifre pare).

b) Să se scrie în limbajul C/C++ un program care citește de la tastatură un număr natural `n` de cel mult 7 cifre și, utilizând apeluri ale subprogramului `ca1cu1`, determină și afișează pe ecran cel mai mic număr natural `m`, $m \geq n$, care are tot atâtea cifre pare câte cifre impare.

Exemple: dacă se citește `n=5513`, atunci se afișează `m=5520`, iar dacă se citește `n=311`, atunci se afișează `m=1001`.

63. Să se scrie în limbajul C/C++ definiția completă a subprogramului `ca1cu1`, care primește prin parametrul `n` un număr natural nenul de cel mult 9 cifre și furnizează prin parametrul `x` numărul obținut prin alăturarea cifrelor pare ale lui `n` considerate de la dreapta către stânga. Dacă `n` nu conține nicio cifră pară, `x` primește valoarea 0.

Exemplu: în urma apelului `ca1cu1(9278, x)`, `x` primește valoarea 82.

64. Un număr natural se numește palindrom dacă numărul citit de la stânga la dreapta este egal cu numărul citit de la dreapta la stânga.

a) Scrieți definiția completă a subprogramului `Palindrom` care primește prin intermediul parametrului `n` un număr natural de cel mult nouă cifre și returnează 1 dacă acesta este palindrom și 0 în caz contrar.

b) Fișierul text `NUMERE.IN` conține cel mult 100000 numere naturale de cel mult nouă cifre fiecare, numerele fiind despărțite prin câte un spațiu. Cel puțin unul dintre numere este palindrom. Scrieți programul C/C++ care citește numerele din fișierul `NUMERE.IN` și, folosind apeluri utile ale subprogramului `Palindrom` determină în mod eficient, din punct de vedere al memoriei utilizate și al timpului de executare, care este cel mai mare număr palindrom citit și de câte ori apare el în fișierul `NUMERE.IN`. Programul scrie în fișierul text `NUMERE.OUT` numărul astfel determinat precum și numărul de apariții ale acestuia, pe rânduri diferite.

Exemplu: dacă `NUMERE.IN` conține numerele: 23 565 78687 7887 7865 78687 7887 23 78687 98798

atunci `NUMERE.OUT` va conține: 78687 3 (pe randuri diferite).

c) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul **b**, explicând în ce constă eficiența ei (3 – 4 rânduri).

65. **a)** Scrieți definiția completă a subprogramului `Ecuatie` care primește prin parametrii `a`, `b` și `c` trei numere întregi, $a \neq 0$, de cel mult patru cifre fiecare, reprezentând coeficienții ecuației de gradul al II-lea: $ax^2 + bx + c = 0$. În funcție de soluțiile ecuației subprogramul va returna:
– cea mai mare dintre soluții dacă ecuația are două soluții reale distincte, dintre care cel puțin una pozitivă.

– una dintre soluții dacă ecuația are două soluții egale și pozitive.

– -32000 în celelalte cazuri.

b) Se consideră șirul `s`: 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, 1, 2, 3, 4, 5, 1, 2, 3, 4, 5, 6, 1, 2, ...

Pentru un număr natural `k`, $0 < k \leq 10000$, se cere să se determine valoarea elementului ce se află pe poziția `k` în șirul `s`.

Exemplu: pentru `k=5` numărul cerut este 2. Scrieți un program C/C++ care citește de la tastatură valoarea numărului natural `k` și, prin apeluri utile ale funcției `Ecuatie`, determină valoarea elementului ce se află pe poziția `k` în șirul `s`, folosind un algoritm eficient din punctul de vedere al spațiului de memorie alocat și al timpului de executare. Valoarea astfel determinată se va scrie în fișierul text `sir.out`.

c) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul **b**, explicând în ce constă eficiența ei (3 – 4 rânduri)

66. **a)** Scrieți definiția completă a subprogramului **Cutare**, cu trei parametri, **n**, **x** și **v**, care primește prin parametrul **n** un număr natural ($1 \leq n \leq 1000$), prin parametrul **x** un tablou unidimensional format din **n** componente (numere întregi de cel mult patru cifre fiecare: **x1**, **x2**, ..., **xn**) memorate în ordine crescătoare și prin parametrul **v** un număr întreg de cel mult patru cifre, diferit de oricare dintre elementele tabloului unidimensional **x**. Subprogramul va căuta, în mod eficient din punct de vedere al timpului de executare, poziția pe care ar trebui inserată valoarea **v** în șirul **x** astfel încât să se obțină tot un șir ordonat și returnează această poziție.

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri).

c) Fișierul text **sir.in** conține cel mult 1000 numere naturale de maximum patru cifre fiecare, numerele fiind diferite două câte două și despărțite prin câte un spațiu.

Scrieți un program C/C++ care citește numerele din fișierul **sir.in** și, folosind apeluri utile ale subprogramului **Cutare**, construiește în memorie un tablou unidimensional care va conține toate numerele din fișierul **sir.in** ordonate crescător. Programul scrie în fișierul text **sir.out** șirul obținut, câte 10 elemente pe un rând, elementele de pe același rând fiind despărțite printr-un singur spațiu.

Exemplu: dacă fișierul **sir.in** conține numerele: 7 -5 635 -456 0 8 587 -98 65 3 -8 atunci după executarea programului fișierul **sir.out** va conține:

```
-456 -98 -8 -5 0 3 7 8 65 587
635
```

67. **a)** Scrieți definiția completă a subprogramului **Cifra**, cu doi parametri, **n** și **x**, care primește prin intermediul parametrului **n** un număr natural de cel mult nouă cifre și furnizează prin parametrul **x** cea mai mare cifră a numărului transmis prin parametrul **n**.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural **n**, $n < 1000000000$, și afișează pe ecran mesajul **Da** în cazul în care numărul citit este format doar din cifre aparținând mulțimii $\{0, 1, 2, 3, 4\}$ și afișează mesajul **Nu** în caz contrar.

68. **a)** Scrieți definiția completă a funcției **UltimaCifra** care primește prin cei doi parametri **a** și **b** câte un număr natural ($0 < a < 1000000$, $0 < b < 1000000$), calculează în mod eficient din punct de vedere al timpului de executare și returnează ultima cifră a numărului a^b (**a** la puterea **b**).

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri)

c) Fișierul text **SIR.IN** conține pe prima sa linie un număr natural **n** ($0 < n < 1001$), iar pe fiecare dintre următoarele **n** linii câte o pereche de numere naturale, **xi yi** ($1 \leq i \leq n$, $x_i \leq 30000$, $y_i \leq 30000$). Scrieți programul C/C++ care citește numerele din fișierul **SIR.IN** și scrie în fișierul text **SIR.OUT** ultima cifră expresiei:

$x_1^{y_1} + x_2^{y_2} + \dots + x_n^{y_n}$ folosind apeluri ale funcției **UltimaCifra**.

Exemplu: dacă fișierul **SIR.IN** are conținutul alăturat,

```
3
25 6
8 10
1 4589
```

atunci **SIR.OUT** va conține cifra 0.

69. Scrieți definiția completă a unui subprogram **s** cu trei parametri care primește prin intermediul parametrului **n** un număr natural de maximum 9 cifre, prin intermediul parametrului **c** o cifră și furnizează prin intermediul parametrului **k** numărul de cifre ale numărului **n** care aparțin intervalului $[c-1, c+1]$. **Exemplu:** pentru $n=1233$ și $c=3$, **k** va avea valoarea 3, iar pentru $n=650$ și $c=3$, **k** va avea valoarea 0.

70. a) Scrieți doar antetul unui subprogram `prim` cu doi parametri, care primește prin intermediul parametrului `n` un număr natural cu cel mult patru cifre și returnează prin intermediul parametrului `p` valoarea 1 dacă `n` este prim și 0 în caz contrar.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural `n` ($3 < n < 10000$) și afișează pe ecran, despărțite prin câte un spațiu, primele `n` numerele prime, folosind apeluri utile ale subprogramului `prim`.

Exemplu: pentru `n=4` pe ecran vor fi afișate numerele 2 3 5 7

71. Scrieți definiția completă a subprogram `max`, cu trei parametri, `a`, `b`, `c`, care primește prin intermediul parametrilor `a` și `b` două numere reale cu exact două cifre la partea întregă și exact două zecimale fiecare. Subprogramul determină cel mai mare număr real dintre următoarele patru valori: `a`, `b` și numerele reale obținute din `a` și `b` prin interschimbarea părții întregi cu partea fracționară în cadrul aceluiași număr. Această valoare este furnizată prin intermediul parametrului real `c`.

Exemplu: dacă `a=33.17` și `b=15.40`, `c` va avea valoarea 40.15 (cea mai mare valoare dintre 33.17, 15.40, 17.33 și 40.15)

72. a) Scrieți definiția completă a unui subprogram `divi` cu doi parametri, care primește prin intermediul parametrului `n` un număr natural nenul cu cel mult 6 cifre și returnează prin intermediul parametrului `d` cel mai mic divizor propriu al lui `n` sau 0 în cazul în care `n` nu are niciun divizor propriu. De exemplu, pentru `n=15`, `d` va avea valoarea 3.

b) Scrieți programul C/C++ care citește de la tastatură un număr natural nenul `n` ($1 < n < 1000000$) și afișează pe ecran cel mai mare număr natural mai mic decât `n` care îl divide pe `n`. Se vor folosi apeluri utile ale subprogramului `divi`.

Exemple: dacă `n=120`, se afișează pe ecran 60, iar dacă `n=43`, se afișează 1.

73. Scrieți în C/C++ definiția completă a subprogramului `medie` care are doi parametri:

- `n`, prin care primește un număr natural ($1 \leq n \leq 100$);

- `v`, prin care primește un tablou unidimensional cu `n` elemente, numere naturale, fiecare element având cel mult patru cifre.

Subprogramul returnează media aritmetică a elementelor din tablou.

74. Fișierul text `NUMERE.IN` conține, pe mai multe linii, cel mult 30000 de numere naturale nenule mai mici sau egale cu 500, numerele de pe fiecare linie fiind despărțite prin câte un spațiu. Fișierul conține cel puțin două numere distincte, fiecare având două cifre.

a) Scrieți programul C/C++ care citește toate numerele din fișierul `NUMERE.IN` și creează fișierul text `NUMERE.OUT` care să conțină pe prima linie cel mai mare număr de două cifre din fișierul `NUMERE.IN`, și de câte ori apare el în acest fișier, iar pe a doua linie, cel mai mic număr de două cifre din fișierul `NUMERE.IN` și de câte ori apare el în acest fișier. Alegeți o metodă de rezolvare eficientă din punct de vedere al memoriei utilizate și al timpului de executare.

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită la punctul **a)**, explicând în ce constă eficiența ei (3 – 4 rânduri).

Exemplu: dacă fișierul `NUMERE.IN` are conținutul alăturat:

```
2 253 34 3
```

```
6 88 9 2 3
```

```
4 54 34 88
```

atunci fișierul `NUMERE.OUT` va avea următorul conținut:

```
88 2
```

```
34 2
```

75. Scrieți în C/C++ definiția completă a subprogramului `suma` care are doi parametri:

- `n`, prin care primește un număr natural ($1 \leq n \leq 100$);
- `v`, prin care primește un tablou unidimensional cu `n` elemente, numere întregi, fiecare având exact trei cifre. Funcția returnează suma elementelor din tablou care au prima cifră egală cu ultima cifră.

76. Se consideră subprogramul `CMMC` care primește prin cei doi parametri, `x` și `y`, două numere naturale ($1 \leq x \leq 10000$, $1 \leq y \leq 10000$) și returnează cel mai mic multiplu comun al lor.

a) Scrieți numai antetul subprogramului `CMMC`.

b) Fișierul text `NUMERE.IN` conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât 10000, despărțite printr-un spațiu. Scrieți un program C/C++ care, pentru fiecare linie `k` din fișierul `NUMERE.IN`, citește cele două numere de pe această linie și scrie în fișierul text `NUMERE.OUT`, tot pe linia `k`, cel mai mic multiplu comun al acestora, ca în exemplu. Se vor utiliza apeluri utile ale subprogramului `CMMC`. **Exemplu:** dacă fișierul `NUMERE.IN` are conținutul alăturat:

```
12 14
11 12
```

```
4 8
```

atunci fișierul `NUMERE.OUT` va avea următorul conținut:

```
84
132
8
```

77. Se consideră subprogramul `CMMDC` care primește prin cei doi parametri, `x` și `y`, două numere naturale ($1 \leq x \leq 10000$, $1 \leq y \leq 10000$) și returnează cel mai mare divizor comun al lor.

a) Scrieți numai antetul subprogramului `CMMDC`.

b) Fișierul text `NUMERE.IN` conține, pe fiecare linie, câte două numere naturale nenule mai mici sau egale decât 10000, despărțite printr-un spațiu, reprezentând numitorul și numărătorul câte unei fracții. Scrieți un program C/C++ care, pentru fiecare linie `k` din fișierul `NUMERE.IN`, citește numitorul și numărătorul fracției de pe această linie și scrie în fișierul text `NUMERE.OUT`, tot pe linia `k`, numitorul și numărătorul acestei fracții, adusă la forma ireductibilă, ca în exemplu. Se vor utiliza apeluri utile ale subprogramului `CMMDC`.

Exemplu: dacă fișierul `NUMERE.IN` are conținutul alăturat:

```
12 14
11 12
2 2
4 8
```

atunci fișierul `NUMERE.OUT` va avea următorul conținut:

```
6 7
11 12
1 1
1 2
```

78. Scrieți definiția completă a subprogramului `numar`, cu trei parametri, care primește prin intermediul parametrului `n` un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor `c1` și `c2` câte o cifră nenulă; subprogramul returnează numărul obținut prin înlocuirea în numărul primit prin parametrul `n` a fiecărei apariții a cifrei `c1` cu cifra `c2`. Dacă `c1` nu apare în `n`, subprogramul returnează valoarea `n`.

Exemplu: pentru `n=12445`, `c1=4` și `c2=7` valoarea returnată va fi 12775.

79. Scrieți definiția completă a subprogramului **reduce**, cu doi parametri, care primește prin intermediul parametrilor **a** și **b** două numere naturale formate din cel mult 9 cifre fiecare. Funcția returnează o valoare obținută din numărul **a** prin însumarea acelor cifre diferite de 0 ale numărului **a** care **NU** divid numărul **b**. Dacă nu există asemenea cifre, se va returna valoarea 0.

Exemplu: pentru **a=184465709** și **b=18**, cifrele corespunzătoare cerinței sunt 8, 4, 4, 5 și 7, deci valoarea returnată va fi 28 ($28=8+4+4+5+7$). Dacă **a=2402804** și **b=8000**, valoarea returnată va fi 0.

80. Scrieți definiția completă a subprogramului **numar**, cu patru parametri, care primește prin intermediul parametrului **n** un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor **c1** și **c2** câte o cifră cu proprietatea $c1 < c2$; subprogramul furnizează prin intermediul celui de al patrulea parametru, **x**, o valoare obținută prin eliminarea din numărul primit prin parametrul **n** a fiecărei cifre cuprinse în intervalul închis $[c1, c2]$. Dacă toate cifrele lui **n** aparțin acestui interval, valoarea furnizată prin **x** va fi 0. **Exemplu:** pentru **n=162448**, **c1=4** și **c2=7**, valoarea furnizată prin **x** va fi 128.

81. Scrieți definiția completă a subprogramului **numar**, cu trei parametri, care primește prin intermediul parametrului **n** un număr natural format din cel mult 9 cifre, iar prin intermediul parametrilor **c1** și **c2** câte o cifră nenulă. Subprogramul caută prima apariție (de la stânga spre dreapta) a cifrei **c1** în **n**, și dacă aceasta apare, o înlocuiește cu **c2**, iar următoarele cifre, dacă există, sunt înlocuite cu câte o cifră 0. Subprogramul furnizează tot prin **n** numărul astfel obținut. Dacă cifra **c1** nu apare în **n**, atunci valoarea lui **n** rămâne nemodificată.

Exemplu: pentru **n=162448**, **c1=4** și **c2=7** valoarea furnizată prin **n** va fi 162700.

82. Funcția **verif** primește prin intermediul parametrului **n** un număr natural format din cel mult 9 cifre, și prin intermediul parametrului **a**, un număr natural nenul ($2 \leq a \leq 9$). Funcția returnează valoarea 1 dacă **n** este un număr format din cifre aparținând intervalului închis $[0, a]$ și valoarea 0 în caz contrar.

a) Scrieți definiția completă a funcției **verif**.

b) Spunem că **n** poate fi o reprezentare în baza **b** ($1 < b \leq 10$), dacă toate cifrele lui **n** sunt strict mai mici decât **b**. Scrieți un program care citește de la tastatură o valoare naturală **n** cu cel mult 9 cifre și, utilizând apeluri ale funcției **verif**, afișează pe ecran, în ordine crescătoare, cu spații între ele, toate valorile lui **b** pentru care valoarea citită **nu** poate fi o reprezentare în baza **b**.

Exemplu: Pentru **n=4101**, se afișează 2 3 4.

83. Subprogramul **par** primește prin singurul său parametru, **n**, un număr natural nenul cu cel mult 8 cifre și returnează valoarea 1 dacă **n** conține cel puțin o cifră pară, sau returnează valoarea 0 în caz contrar.

Exemplu: pentru **n=723** subprogramul va returna valoarea 1.

a) Scrieți numai antetul subprogramului **par**.

b) Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** cu cel mult trei cifre, apoi un șir de **n** numere naturale, cu cel puțin două și cel mult 8 cifre fiecare, și afișează pe ecran numărul de valori din șirul citit care au numai cifra unităților pară, celelalte cifre fiind impare. Se vor utiliza apelurile utile ale subprogramului **par**. **Exemplu:** dacă **n=4**, iar șirul citit este 7354, 123864, 51731, 570 se va afișa 2 (numerele 7354 și 570 respectă condiția cerută).

84. Subprogramul `ordonare` primește prin parametrul `x` un tablou unidimensional cu cel mult 100 de elemente numere reale, iar prin parametrul `n` un număr întreg ce reprezintă numărul efectiv de elemente ale tabloului `x`. Subprogramul ordonează crescător elementele tabloului și furnizează, tot prin intermediul parametrului `x`, tabloul ordonat.

a) Scrieți numai antetul acestui subprogram.

b) Scrieți un program C/C++ care citește de la tastatură două numere naturale, `n` și `m` ($1 \leq n \leq 100$ și $m \leq n$), și apoi un șir de `n` numere reale distincte. Folosind apeluri utile ale subprogramului `ordonare`, programul afișează pe prima linie a ecranului, cele mai mari `m` elemente din șirul citit (în ordine crescătoare a valorilor lor), iar pe a doua linie de ecran, cele mai mici `m` elemente din șir (în ordine descrescătoare a valorilor lor). Numerele afișate pe aceeași linie vor fi separate prin câte un spațiu.

Exemplu : dacă `n=9`, `m=3`, iar șirul este (14.2, 60, -7.5, -22, 33.8, 80, 4, 10, 3) se va afișa pe ecran: 33.8 60 80

3 -7.5 -22

85. a) Scrieți definiția completă a subprogramului `dcm`, cu doi parametri, care:

- primește prin parametrii `a` și `b` două valori naturale din intervalul $[1, 30000]$

- returnează o valoare naturală reprezentând cel mai mare număr care este atât divizor al lui `a` cât și divizor al lui `b`.

Exemplu: dacă `a=100` și `b=120`, subprogramul returnează valoarea 20.

b) Scrieți un program C/C++ care citește de la tastatură două numere naturale `a` și `b`, numere din intervalul $[1, 30000]$ și determină, folosind apeluri utile ale subprogramului `dcm`, cel mai mare număr care este divizor al fiecăruia dintre numerele `a` și `b` și are proprietatea că este un produs de două sau mai multe numere prime distincte. Programul afișează pe ecran numărul cu proprietatea cerută, iar dacă nu există un astfel de număr, afișează mesajul `nu exista`.

Exemplu: dacă `a=60` și `b=72`, atunci se afișează 6

iar dacă `a=100` și `b=75`, atunci se afișează `nu exista`.

86. Scrieți definiția completă a funcției `f`, care primește prin intermediul parametrului `n` un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului `a` un tablou unidimensional care conține `n` valori întregi, fiecare dintre aceste valori întregi având cel mult patru cifre.

Funcția returnează valoarea 1 dacă elementele tabloului formează un șir crescător, valoarea 2 dacă elementele tabloului formează un șir descrescător, valoarea 0 dacă elementele tabloului formează un șir constant și valoarea -1 în rest.

87. Funcția `f` primește prin intermediul parametrului `n` un număr natural nenul ($2 \leq n \leq 200$), iar prin intermediul parametrului `a` un tablou unidimensional care conține `n` valori întregi nenule (fiecare dintre aceste valori întregi având cel mult patru cifre).

Funcția returnează valoarea -1 dacă numărul de valori negative din tabloul `a` este strict mai mare decât numărul de valori pozitive din tablou, valoarea 0 dacă numărul de valori negative din `a` este egal cu numărul de valori pozitive din tablou și valoarea 1 dacă numărul de valori pozitive din tabloul `a` este strict mai mare decât numărul de valori negative din `a`. Scrieți definiția completă a funcției `f`.

89. Funcția `verif` primește prin intermediul a trei parametri, notați `a`, `b` și `c`, trei valori naturale nenule, fiecare de maximum patru cifre. Funcția returnează valoarea 1 dacă cele trei valori pot constitui laturile unui triunghi și valoarea 0 în caz contrar.

a) Scrieți definiția completă a funcției `verif`.

b) Scrieți un program C/C++ care citește de la tastatură șase valori naturale nenule, fiecare de maximum patru cifre, apoi verifică, utilizând apeluri utile ale funcției `verif`, dacă primele

trei numere citite pot constitui laturile unui triunghi și dacă ultimele trei numere citite pot constitui laturile unui triunghi; în caz afirmativ, programul afișează pe ecran mesajul **congruente** dacă cele două triunghiuri sunt congruente sau mesajul **necongruente** dacă cele două triunghiuri nu sunt congruente; dacă cel puțin unul dintre cele două triplete de valori nu pot constitui laturile unui triunghi, programul va afișa pe ecran mesajul **nu**.

90. Tabloul unidimensional **v**, declarat global, memorează exact 50 de numere întregi: **v1, v2, ..., v50**. Subprogramul **Calcul** primește prin intermediul parametrului **k** un număr natural nenul ($k \leq 50$) și furnizează prin intermediul parametrului **s** suma tuturor elementelor pozitive, din tabloul **v**, cu indici mai mari sau egali cu **k** sau 0 dacă toate elementele menționate sunt negative.

a) Scrieți doar antetul subprogramului **Calcul**.

b) Scrieți un program în limbajul C/C++ care citește de la tastatură cele 50 de componente întregi ale tabloului **v** și două numere naturale nenule **x** și **y** ($x < y \leq 50$). Programul afișează suma elementelor pozitive din tablou, cu indici cuprinși între **x** și **y** inclusiv, sau 0 dacă toate elementele menționate sunt negative, folosind apeluri utile la subprogramul **Calcul**.

91. Scrieți în limbajul C/C++ definiția completă a subprogramului **numar**, cu exact doi parametri, care primește prin intermediul parametrului **x** un număr natural nenul de cel mult 2 cifre, și prin intermediul parametrului **y** un număr natural nenul de cel mult 9 cifre. Subprogramul returnează cel mai mare număr natural **z** pentru care există un număr natural **k** astfel încât $z = xk$ și $z \leq y$. **Exemplu:** pentru $y=18$ și $x=2$ subprogramul va returna valoarea 16 ($=24 < 18$).

92. Subprogramul **Nr** are un singur parametru, **k**, prin intermediul căruia primește un număr natural de cel puțin 3 cifre și cel mult 9 cifre, cu toate cifrele nenule. Subprogramul furnizează tot prin intermediul parametrului **k**, valoarea obținută prin eliminarea primei și ultimei cifre a numărului transmis la apel.

Exemplu: dacă subprogramul primește prin intermediul parametrului **k** valoarea 12438, în urma apelului subprogramului **Nr**, **k** va primi valoarea 243.

Scrieți, în limbajul C/C++, definiția completă a subprogramului **Nr**.

93. Scrieți definiția completă a subprogramului **divizor**, cu trei parametri, prin care primește 3 numere naturale nenule cu cel mult 9 cifre fiecare și returnează numărul divizorilor comuni tuturor celor 3 numere.

Exemplu: dacă numerele primite ca parametri sunt 24, 20 și 12, subprogramul returnează valoarea 3 (divizorii comuni sunt 1, 2 și 4).