

Subprogramul **aranjare** are 2 parametri prin care primește un tablou unidimensional cu maximum 100 de numere reale și numărul de elemente din tabloul unidimensional. 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.

**Exemplu:** dacă este transmis ca parametru un tablou unidimensional cu 6 elemente 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**. (10p.)

Subprogramul **f** primește prin intermediul parametrului **n** un număr natural nenul ( $2 \leq n \leq 9$ ), iar prin intermediul parametrului **a**, un tablou unidimensional care conține **n** valori naturale distincte, fiecare dintre acestea având cel mult patru cifre. Subprogramul interschimbă cel mai mic element par și cel mai mare element impar din tabloul **a** și furnizează tabloul modificat tot prin parametrul **a**. Dacă tabloul nu conține niciun element par sau niciun element impar, el va rămâne neschimbat.

**Exemplu:** dacă subprogramul se apelează pentru **n=6** și pentru **a** având valorile (2, 3, 17, 6, 1, 24), tabloul va avea în urma apelului următoarea formă: (17, 3, 2, 6, 1, 24). (10p.)

Subprogramul **nule** are 2 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**.

(10p.)

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. (6p.)

b) Fișierul **NUMERE.IN** conține pe prima linie un număr natural nenul **n** ( $1 \leq n \leq 100$ ) 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 și afișează pe ecran, separate prin câte un spațiu, folosind apeluri utile ale subprogramului **primul**, numerele „aproape prime” de pe linia a doua a fișierului **NUMERE.IN**.

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

6

10 14 21 8 77 35 atunci se afișează pe ecran 14 21 77 35

(10p.)

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`, cele `n` componente ale unui tablou unidimensional. Fiecare componentă a acestui tablou este un număr întreg care are cel mult 4 cifre. Subprogramul permută circular cu o poziție spre stânga, elementele tabloului `x`, și furnizează tabloul modificat tot prin parametrul `x`.

**Exemplu:** dacă înainte de apel `x = (1, 2, 3, 4)`, după apel `x = (2, 3, 4, 1)`. (4p.)

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 citesc 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)`. (6p.)

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 `p2 - p1` în care `p1` și `p2` sunt numere prime și  $p1 \leq n \leq p2$ .

**Exemplu:** dacă `n = 20` atunci `i_prim(10) = 4`, valoare obținută pentru `p1 = 19` și `p2 = 23`. (10p.)

Scrieți definiția completă a subprogramului `nreal` cu doi parametri `x` și `y`, numere naturale din intervalul `[1, 1000]` și returnează un număr real cu proprietatea că partea sa întreagă este egală cu `x` iar cifrele numărului `y`, sunt egale, în ordine, cu cifrele aflate după punctul zecimal.

**Exemplu:** pentru `x = 12` și `y = 543`, subprogramul returnează valoarea `12.543`. (10p.)

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`. (6p.)

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 citesc 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`. (4p.)

a) Scrieți definiția unui subprogram, `nz`, cu un parametru întreg, `n` ( $0 < n \leq 12$ ), care returnează numărul zerourilor de la sfârșitul numărului `n!`. (6p.)

b) Scrieți o secvență de instrucțiuni prin care, fiind dat un număr natural `k` ( $0 < k \leq 1500$ ), să se determine, folosind apeluri ale subprogramului `nz`, cel mai mic număr natural `n` pentru care `n!` are cel puțin `k` zerouri la sfârșit. (4p.)

Scrieți definiția completă a subprogramului `interval` care are 2 parametri prin care primește un tablou unidimensional cu maximum 100 de numere naturale mai mici decât 1000 și numărul 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. (10p.)

Subprogramul `cifrak` are 2 parametri, `n`, prin care primește un număr natural cu maximum 9 cifre și `k`, prin care primește un număr natural ( $k \leq 9$ ). Subprogramul returnează numărul de cifre ale numărului `n` care sunt egale cu valoarea `k`.

**Exemplu:** dacă `n=233433`, iar `k=3`, subprogramul va returna valoarea 4.

a) Scrieți definiția completă a subprogramului `cifrak`. (4p.)

b) În fișierul `numere.txt` sunt memorate mai multe numere naturale (maximum 1000 de numere cu maximum 9 cifre fiecare). Fiecare linie a fișierului conține câte un număr. Scrieți programul `Pascal` care, folosind apeluri ale subprogramului `cifrak`, afișează pe ecran, separate prin câte un spațiu, toate numerele din fișier care conțin exact 3 cifre de 0.

**Exemplu:** dacă fișierul `numere.txt` conține

260070

39008

70009

se vor afișa numerele 260070 70009. (6p.)

a) Scrieți definiția completă a unui subprogram `sterge`, cu trei parametri, care:

- primește prin parametri:

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

- `n` o valoare naturală reprezentând numărul de elemente din tabloul `v`

- `i` o valoare naturală cu  $1 \leq i \leq n$

- elimină din tabloul `v` elementul `v[i]` și actualizează valoarea lui `n`. (10p.)

b) Fișierul `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. (6p.)

Subprogramul `count` are 2 parametri prin care primește un tablou unidimensional cu maximum 100 de numere reale și numărul de elemente din tabloul unidimensional. Subprogramul returnează numărul de elemente din tabloul unidimensional care sunt mai mari sau cel puțin egale cu media aritmetică dintre primul și ultimul element al tabloului.

**Exemplu:** dacă tabloul are 6 elemente și este de forma (12, 7.5, 6.5, 8.5, 7.5, 3), subprogramul va returna valoarea 2.

Scrieți definiția completă a subprogramului `count`. (10p.)

a) Scrieți definiția completă a unui subprogram **P** cu 3 parametrii, care primește prin intermediul primului parametru **a** un vector de numere întregi, prin intermediul celui de al doilea parametru **k** un număr natural ( $k < 101$ ) și furnizează prin intermediul celui de al treilea parametru al său **max** cea mai mare valoare dintre valorile  $a_1, a_2, \dots, a_k$  din vector.

**Exemplu:** pentru  $k=5$  și vectorul  $a=(7,3,8,4,6,\dots)$ , în urma apelului  $P(a,k,max)$  valoarea variabilei **max** este 8. (4p.)

b) Să se scrie un program **C/C++** care citește de la tastatură un număr natural **n** ( $n < 101$ ), apoi **n** numere întregi, având maximum 4 cifre fiecare și construiește în memorie și afișează apoi pe ecran un vector de **n** numere întregi, cu proprietatea că valoarea termenului de pe poziția **i** ( $i=1,2,\dots,n$ ) din acest vector este egală cu cea mai mare dintre primele **i** valori din șirul dat. Se vor folosi apeluri utile ale subprogramului **P**.

**Exemplu:** dacă se citesc de la tastatură  $n=12$  și valorile 4 6 3 7 8 1 6 2 7 9 10 8 se va afișa pe ecran vectorul 4 6 6 7 8 8 8 8 8 9 10 10. (6p.)

Se consideră subprogramul **prim** care primește ca parametru un număr natural **n** ( $n \leq 32000$ ) și care returnează 1 dacă **n** este număr prim și respectiv 0 în caz contrar.

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

b) Scrieți programul **C/C++** care, utilizând apeluri utile ale subprogramului **prim**, afișează pe ecran toate numerele prime de două cifre care, citite invers (de la dreapta la stânga), sunt tot prime. Un astfel de număr este de exemplu 13 pentru că atât 13 cât și 31 sunt numere prime. (6p.)

Se consideră subprogramul **P** care primește ca parametri un număr natural **n** cu maximum 9 cifre și o cifră **c** și care va elimina din numărul **n** toate aparițiile cifrei **c**, furnizând tot prin parametrul **n** numărul obținut.

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

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 un program **C/C++** care citește numerele din acest fișier, elimină toate cifrele impare din fiecare dintre aceste numere și apoi scrie în fișierul text **BAC.OUT** numerele astfel obținute. Se vor folosi apeluri utile ale subprogramului **P**. Dacă un număr din fișierul **BAC.IN** conține doar cifre impare și cifra 0, acesta nu va mai apărea deloc în fișierul de ieșire. (6p.)

**Exemplu:** dacă fișierul **BAC.IN** conține numerele 25 7 38 130 45127 0 35 60 15 atunci **BAC.OUT** va avea conținutul: 2 8 42 60.

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 2 cifre ( $a > 1$ )
- returnează suma exponenților divizorilor primi 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  $1+2+1=4$ . (4p.)

b) Scrieți un program  $C/C++$  care citește de la tastatură un număr natural  $n$ ,  $2 \leq n \leq 99$ , și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului  $f$ , mesajul DA dacă  $n$  este un număr **extraprim**, altfel va afișa mesajul NU. (6p.)

Subprogramul  $sub$ , cu trei parametri, primește prin intermediul parametrului:

- $v$  un tablou unidimensional cu cel mult 100 de componente ce memorează numere întregi cu cel mult 4 cifre
- $n$  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 egale cu valoarea parametrului  $a$ .

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

a) Scrieți definiția completă a subprogramului  $sub$ . (4p.)

b) Scrieți 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 numere dintre cele  $n$  numere 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 (6p.)

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

a) Scrieți definiția completă a subprogramului  $div$  (4p.)

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 construiește în memorie un tablou unidimensional ale cărui elemente memorează în ordine crescătoare, toți divizorii lui  $n$  din intervalul închis determinat de  $a$  și  $b$  folosind apeluri utile ale subprogramului  $div$ . Intervalul închis determinat de  $a$  și  $b$  este  $[a, b]$  dacă  $a < b$  sau  $[b, a]$  dacă  $b \leq a$ .

**Exemplu:** pentru  $a=65$ ,  $b=11$  și  $n=140$  tabloul va conține: 14 20 28 35 (6p.)

Se consideră subprogramul `div` care primește prin parametrii `x` și `y` două valori întregi pozitive ( $0 < x < 100$  și  $0 < y < 100$ ) și returnează cel mai mare divizor comun al lor.

a) Scrieți doar antetul subprogramului `div`. (4p.)

b) Scrieți un program `C/C++` care citește de la tastatură trei numere naturale nenule distincte `a`, `b` și `n`, cu cel mult 2 cifre fiecare și care construiește în memorie un tablou unidimensional ale cărui elemente memorează în ordine crescătoare, toate numerele naturale cuprinse în intervalul închis determinat de `a` și `b`, care nu au niciun divizor strict mai mare decât 1 comun cu `n`, folosind apeluri utile ale subprogramului `div`. Intervalul închis determinat de `a` și `b` este `[a, b]` dacă  $a < b$  sau `[b, a]` dacă  $b \leq a$ .

**Exemplu:** pentru `a=60`, `b=32` și `n=36` tabloul va conține: 35 37 41 43 47 49 53 55 59 (6p.)

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`. (4p.)

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 determină și 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 (6p.)

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`. (4p.)

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 folosind apeluri utile ale subprogramului `cmax`. Cifra determinată se va afișa pe ecran.

**Exemplu:** dacă fișierul `bac.txt` conține valorile: 23 12 64 12 72 345 67 23 71 634 atunci pe ecran se afișează 7. (6p.)

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, fiecare element având cel mult patru cifre.

Funcția returnează media aritmetică a elementelor impare din tablou sau valoarea 0 dacă nu există elemente impare. (10p.)



Subprogramul `ordonat` are 4 parametri:

- `n`, prin care primește un număr natural ( $1 \leq n \leq 100$ )
- `v`, prin care primește un tablou unidimensional cu `n` elemente, numerotate de la 1 la `n`, numere naturale cu cel mult patru cifre fiecare.
- `k1` și `k2` – două numere întregi ( $1 \leq k1 \leq k2 \leq n$ )

Subprogramul returnează valoarea 1 dacă elementele din tablou, cu indici în intervalul `[k1, k2]`, se află în ordine crescătoare, și 0 în caz contrar.

a) Scrieți numai antetul subprogramului `ordonat`. (4p.)

b) Scrieți un program `C/C++` care citește de la tastatură un număr natural `n` ( $1 \leq n \leq 100$ ) și `n` numere naturale, cu cel mult 4 cifre fiecare, apoi determină și afișează pe ecran, separate printr-un spațiu, două valori `k1` și `k2` astfel încât secvența formată din elementele cu indici în intervalul `[k1, k2]`, pe poziții consecutive în tablou, să fie ordonată crescător și să aibă o lungime maximă. Dacă sunt mai multe secvențe de lungime maximă se vor afișa indicii corespunzători unei valori minime a lui `k1`. Se vor utiliza apeluri utile ale subprogramului `ordonat`.

**Exemplu:** dacă `n=7` și tabloul este (4, -6, 1, 2, 1, 4, 5) se va afișa 2 4. (10p.)

a) Scrieți definiția completă a unui subprogram `max_neg` cu trei parametri, care primește prin intermediul parametrului `n` ( $0 < n < 100$ ) un număr natural, prin intermediul parametrului `v` un tablou unidimensional cu `n` numere întregi cu cel mult 4 cifre fiecare și furnizează prin intermediul parametrului `max` cea mai mare valoare strict negativă dintre numerele din tablou sau valoarea 0 dacă nu există astfel de numere. (4p.)

b) Scrieți un program `C/C++` care citește de la tastatură un număr natural `n` ( $3 < n < 100$ ) și apoi `n` numere întregi cu cel mult 4 cifre fiecare și afișează pe ecran cea mai mare valoare strict negativă și cea mai mică valoare strict pozitivă dintre cele citite utilizând pentru ambele valori apeluri utile ale subprogramului `max_neg`. Dacă nu există valori negative se va afișa mesajul `NU EXISTA VALORI NEGATIVE`, respectiv mesajul `NU EXISTA VALORI POZITIVE` dacă nu există valori pozitive.

**Exemplu:** pentru `n=12` și numerele 11 2 2 -3 2 2 3 -13 2 3 2 10 se va afișa:

-3 2 (6p.)

Scrieți definiția completă a unui subprogram `max_d` cu trei parametri, care primește prin intermediul numărului natural `n` un număr natural cu cel mult 9 cifre și cel puțin două cifre și care returnează prin intermediul parametrilor `m1` și `m2` cele mai mari două cifre ale numărului `n`. (10p.)

Scrieți definiția completă a unui subprogram `patrat` care primește prin intermediul parametrului `n` ( $n < 100$ ) un număr natural și prin intermediul parametrului `v` un vector cu `n` elemente numere naturale și afișează pe ecran acele elemente ale tabloului care sunt pătrate perfecte. Numerele vor fi afișate câte 10 pe linie iar pe aceeași linie despărțite prin câte un spațiu.

**Exemplu:** pentru `n=10` și elementele tabloului `v=(3,4,25,100,7,20,64,9,5,50)` pe ecran vor fi afișate numerele 4 25 100 64 9. (10p.)

a) Scrieți definiția completă a unui subprogram `ordonat` care primește prin intermediul singurului său parametru `n` un număr natural cu cel mult 9 cifre și returnează valoarea 1 dacă numărul are cifrele ordonate strict crescător sau strict descrescător și valoarea 0 în caz contrar. (4p.)

Scrieți definiția completă a subprogramului `numar`, cu doi parametri, care primește prin intermediul parametrilor `a` și `b` două numere naturale, `a` fiind format din cel mult 9 cifre iar `b` este un număr natural strict mai mic decât numărul de cifre ale lui `a`. Subprogramul înlocuiește cu 0 ultimele `b` cifre ale numărului `a` și returnează valoarea astfel obținută.

**Exemplu:** pentru `a=184465709` și `b=5`, valoarea returnată va fi `184400000`.

(10p.)

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 egală cu suma cifrelor care aparțin intervalului închis `[c1; c2]`.

**Exemplu:** pentru `a=123`, `c1=2` și `c2=7` valoarea furnizată prin `x` va fi 6.

(10p.)

Subprogramul `par` primește prin singurul său parametru, `n`, un număr natural nenul cu cel mult 4 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`.

(2p.)

b) Scrieți un program `c/c++` care citește de la tastatură un număr natural nenul `n` cu cel mult 9 cifre, apoi un șir de `n` numere naturale, cu exact 8 cifre fiecare, și afișează pe ecran, numărul de valori din șirul citit care au cel puțin o cifră pară printre primele 4 poziții, considerate de la stânga spre dreapta. Se vor utiliza apeluri utile ale subprogramului `par`.

**Exemplu:** dacă `n=3`, iar șirul citit este `12386406`, `37152073`, `73543157` se va afișa 2 (numerele `12386406` și `73543157` respectă condiția cerută).

(8p.)

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`.

(4p.)

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`.

(6p.)

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.

Subprogramul returnează o valoare egală cu numărul de perechi  $(a_i, a_j)$ ,  $1 \leq i < j \leq n$ , în care  $a_i$  este divizor a lui  $a_j$ , sau  $a_j$  este divizor a lui  $a_i$ .

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.

(10p.)



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 funcției `multiplu`. (4p.)

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, 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`. (6p.)

Se consideră subprogramul `divizor`, care:

- primește prin intermediul parametrului `a` un număr natural nenul 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`. (4p.)

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`. (6p.)

Se consideră subprogramul `divizor`, care:

- primește prin intermediul parametrului `a` un număr natural nenul de cel mult 4 cifre, strict mai mare ca 1;
- furnizează prin intermediul parametrului `d`, cel mai mic divizor al lui `a` strict mai mare decât 1;

a) Scrieți numai antetul subprogramului `divizor`. (4p.)

b) Scrieți declarațiile de date și programul principal `c/c++` care citește de la tastatură un număr natural `x` ( $x > 1$ ), 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`. (6p.)

Scrieți definiția completă a subprogramului `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. Subprogramul returnează valoarea 1 dacă diferența dintre elementul maxim și elementul minim este mai mare decât 1000, altfel valoarea returnată va fi 0. (10p.)

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:** la apelul `verif(7325972)` se va returna valoarea 1.

**a) Scrieți definiția completă a subprogramului `verif`. (10p.)**

**b)** Fișierul text `date.txt` conține pe prima linie un număr natural nenul `n` cu cel mult 4 cifre și pe fiecare dintre următoarele `n` linii câte un număr natural, cu exact 6 cifre. Scrieți un program C/C++ care citește numerele din fișierul `date.txt` și afișează pe ecran, separate prin câte un spațiu, acele numere care au primele 3 cifre impare. Se vor utiliza apeluri utile ale subprogramului `verif`. Dacă nu există niciun număr cu această proprietate, se va afișa mesajul `nu`. Alegeți o metodă eficientă din punctul de vedere al memoriei utilizate.

**Exemplu:** dacă fișierul `date.txt` conține

3  
133579  
345796  
973314

Pe ecran se afișează:

133579 973314

(6p.)

Scrieți definiția completă a subprogramului `numar`, cu doi parametri, care primește prin intermediul parametrului `n` un număr natural format din cel mult 9 cifre, iar prin intermediul parametrului `c` o cifră nenulă; subprogramul returnează numărul obținut prin înlocuirea fiecărei apariții ale cifrei `c` în numărul `n` cu partea întreagă a valorii  $c/2$ . Dacă `c` nu apare în `n`, subprogramul returnează valoarea `n`.

**Exemplu:** pentru `a=1525735`, `c=5` valoarea returnată va fi 1222732. (10p.)

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 fiecare apariție a cifrei `c1` în `n`, și dacă aceasta apare, o înlocuiește cu `c2`. Subprogramul furnizează tot prin `n` numărul astfel obținut. Dacă cifra `c1` nu apare în `n`, atunci valoarea lui `n` rămâne nemodificată.

**Exemplu:** pentru `a=149448`, `c1=4` și `c2=2` valoarea furnizată prin `n` va fi 129228. (10p.)

Scrieți programul C/C++ care citește de la tastatură un număr natural `n` ( $1 \leq n \leq 100$ ), un șir de câte `n` numere întregi, cu cel mult 5 cifre fiecare, notat `a1, a2, a3, ..., an`, apoi un al doilea șir de `n` numere întregi, cu cel mult 5 cifre fiecare, notat `b1, b2, b3, ..., bn`. Programul construiește în memorie și afișează pe ecran un șir `c` format din `n` numere calculate astfel: `c[i] = a[i] + b[i]`, pentru `i=1, 2, 3, ..., n`. Numerele afișate vor fi separate prin câte un spațiu.

**Exemplu:** pentru `n=4` și numerele 2, 3, 7, 8 respectiv 43, 3, 1, 8 se afișează 45 6 8 9. (10p.)

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 lui `a`.

a) Scrieți definiția completă a subprogramului `cifre`. (4p.)

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 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 printr-un spațiu, numerele situate pe a doua linie a fișierului, formate numai din cifre ordonate 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, atunci se vor afișa numerele: 16 333 269 (6p.)

6
16 175 333 242477 321 269

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`. (4p.)

Să se scrie în limbajul `c/c++` definiția completă a subprogramului `calcul`, 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 întregi, fiecare dintre aceste valori întregi având cel mult 9 cifre. Subprogramul returnează numărul de numere prime din tablou. (10p.)

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

Să se scrie în limbajul `c/c++` definiția completă a subprogramului `calcul`, 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`. (10p.)

**Exemplu :** în urma apelului `calcul(92465)` se va returna valoarea 1.

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

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. (10p.)

Să se scrie în limbajul `c/c++` definiția completă a subprogramului `calcul`, care primește prin parametrul `n` un număr natural nenul de cel mult 9 cifre și furnizează prin parametrul `x`, numărul obținut prin citirea cifrelor pare ale lui `n` de la dreapta la stânga. Dacă `n` nu conține nici o cifră pară `x` primește valoarea 0. (10p.)

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

Să se scrie în limbajul C/C++ definiția completă a subprogramului `calcul`, care primește prin intermediul celor doi parametri ai săi două numere întregi  $n$  și  $k$  ( $1 \leq n \leq 10000$ ) ( $1 \leq k \leq 5$ ) 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 are rangul 1.

Dacă numărul  $k$  este mai mare decât numărul de cifre al lui  $n$  atunci funcția returnează valoarea -1. (10p.)

**Exemplu :** în urma apelului `calcul(9243, 3)` se va returna 2

Scrieți în limbajul C/C++ definiția completă a subprogramului `Invers` cu doi parametri  $n$  și  $x$ , care primește prin intermediul parametrului  $n$  un număr natural de cel mult nouă cifre și furnizează prin parametrul  $x$  numărul obținut prin inversarea ordinii cifrelor sale, ca în exemplu.

**Exemplu:** dacă  $n=78904$  atunci funcția va returna 40987. (10p.)

Subprogram `sfx` primește prin singurul său parametru,  $x$ , un număr natural din intervalul  $[100, 2000000000]$  și returnează valoarea 1 dacă ultimele trei cifre ale numărului sunt în ordine strict descrescătoare sau valoarea 0 în caz contrar.

**Exemplu:** la apelul `sfx(24973)` se va returna valoarea 1.

a) Scrieți definiția completă a subprogramului `sfx`. (10p.)

Scrieți în limbajul C/C++ definiția completă a subprogramului `Cifre` care primește prin parametrii  $a$  și  $b$  două numere naturale ( $0 < a < 200000000$ ,  $0 < b < 200000000$ ) și returnează numărul de cifre comune ale celor două numere.

**Exemplu:** dacă  $a = 123446$  și  $b = 248766$  atunci subprogramul va returna 3, iar dacă  $a = 1244$  și  $b = 4456$  subprogramul va returna 1. (10p.)

Se consideră subprogramul `numar`, care:

- primește prin intermediul parametrului  $x$  un număr natural nenul de cel mult 4 cifre;
- furnizează prin intermediul parametrului  $nrp$  numărul de numere prime mai mici sau egale cu  $x$ ;

a) Scrieți numai antetul subprogramului `numar`. (4p.)

b) Scrieți declarațiile de date și programul principal C/C++ în care se citesc de la tastatură două numere naturale nenule de cel mult 4 cifre,  $a$  și  $b$  și prin apeluri utile al subprogramului `numar`, se verifică dacă intervalul determinat de  $a$  și  $b$  conține cel puțin un număr prim. Programul va afișa pe ecran în caz afirmativ mesajul DA, iar în caz contrar mesajul NU. Prin intervalul determinat de  $a$  și  $b$  se înțelege intervalul  $[a, b]$  dacă  $a < b$  și  $[b, a]$  în caz contrar. (6p.)

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  $n$ . (10p.)

Scrieți definiția completă a unui subprogram `s` care primește prin intermediul parametrului  $n$  un număr natural de maximum 9 cifre, prin intermediul parametrului  $c$  o cifră și returnează 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. (10p.)

a) Scrieți doar antetul unui subprogram `prim` cu doi parametri, care primește prin intermediul parametrului `n` un număr natural și returnează prin intermediul parametrului `p` valoarea 1 dacă `n` este prim și 0 în caz contrar. (2p.)

b) Scrieți un program `C/C++` care citește de la tastatură un număr natural `n` ( $3 < n < 100$ ) și apoi `n` numere naturale cu maximum 4 cifre fiecare și afișează pe ecran, despărțite prin câte un spațiu, toate numerele prime dintre cele `n` numere citite, în ordinea inversă față de cum au fost citite, utilizând apeluri utile ale subprogramului `prim`. (8p.)

**Exemplu:** pentru `n=12` și numerele 1 2 2 3 7 2 9 3 3 9 7 1 se vor afișa valorile :  
7 3 3 7 3 2 2

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 numai antetul subprogramului `cmmdc`. (4p.)

b) Fișierul text `date.in` conține pe prima linie un număr natural nenul `n` ( $n \leq 100$ ), iar pe a doua linie un șir format din `n` numere naturale nenule, separate prin câte un spațiu, fiecare număr având maximum 8 cifre. Scrieți un program `C/C++` care citește toate numerele din fișierul text `date.in` și afișează pe ecran numărul perechilor de elemente aflate pe poziții consecutive în șir, care sunt prime între ele, folosind apeluri utile ale subprogramului `cmmdc`.

**Exemplu:** dacă fișierul `date.in` are conținutul alăturat, se va afișa 4 pentru că pechile (16,25), (12,7), (7,4), (4, 5) sunt formate din numere prime între ele. (6p.)

Se consideră subprogramul `divizor`, care primește prin intermediul primului parametru, `a`, un număr natural ( $1 < a < 10000$ ) și returnează cel mai mic divizor prim al numărului `a`.

a) Scrieți în limbajul `C/C++` definiția completă a subprogramului `divizor`. (4p.)

b) Fișierul text `date.in` conține pe prima linie un număr natural nenul `n` ( $n \leq 100$ ), iar pe a doua linie `n` numere naturale mai mari decât 1, separate prin câte un spațiu, fiecare număr având maximum 4 cifre. Scrieți un program `C/C++` care citește toate numerele din fișierul text `date.in` și afișează pe ecran suma obținută adunând, pentru fiecare dintre cele `n` numere citite de pe a doua linie din fișierul `date.in`, cel mai mic divizor prim, folosind apeluri utile ale subprogramului `divizor`.

**Exemplu:** dacă fișierul `date.in` are următorul conținutul alăturat, se va afișa 20 ( $20 = 2 + 5 + 2 + 7 + 2 + 2$ ). (6p.)

Subprogramul `dist` primește prin intermediul parametrului `a` un număr natural cu maximum 8 cifre și returnează 1 dacă cifrele numărului `a` sunt distincte, altfel returnează 0.

a) Scrieți definiția completă a subprogramului `dist`. (4p.)

b) Fișierul text `date.in` 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. Scrieți un program `C/C++` care citește toate numerele din fișier și afișează pe ecran, despărțite prin câte un spațiu, numerele de pe a doua linie a fișierului `date.in`, ce au toate cifrele distincte, folosind apeluri utile ale subprogramului `dist`. În cazul în care nu există niciun astfel de număr se va afișa, pe ecran, valoarea 0.

**Exemplu:** dacă fișierul `date.in` are conținutul alăturat, atunci se vor afișa numerele:  
16 123 (6p.)

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 cifre a numărului transmis la apel.

**Exemplu:** dacă subprogramul primește prin intermediul parametrului `k` valoarea 12438, atunci în urma apelului subprogramului `nr`, `k` va primi valoarea 2438.

Scrieți, în limbajul C/C++, definiția completă a subprogramului `nr`.

(10p.)

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`.

Subprogramul `suma` cu trei parametri, primește prin intermediul parametrilor `x` și `y` două numere naturale de cel mult 4 cifre fiecare și furnizează, prin intermediul parametrului `z`, suma numerelor pare cuprinse între `x` și `y` inclusiv.

**Exemplu:** dacă la apel sunt transmise valorile `x=12` și `y=23`, atunci subprogramul `suma` va returna, prin intermediul parametrului `z`, valoarea 102.

Scrieți, în limbajul C/C++, definiția completă a subprogramului `suma`.

(10p.)

Scrieți în limbajul C/C++ definiția completă a subprogramului `numar`, cu exact doi parametri, care primește prin intermediul parametrului `x` un număr natural 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  $z = x^k$  și  $z < y$ .

**Exemplu:** pentru `y=18` și `x=2` subprogramul va returna valoarea 16 ( $=2^4 < 18$ ).

(10p.)

Scrieți definiția completă a subprogramului `divizor`, cu trei parametri, prin care primește 3 numere naturale 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).

(10p.)

a) Scrieți definiția completă a unui subprogram `cmdc`, 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 divizor comun al numerelor `a` și `b`

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

(6p.)



Se consideră funcția  $f$  care primește prin intermediul parametrului  $n$  un număr natural nenul ( $2 \leq n \leq 200$ ), prin intermediul parametrului  $a$  un tablou unidimensional care conține  $n$  valori întregi (fiecare dintre aceste valori întregi având cel mult patru cifre), iar prin intermediul parametrilor  $p1$  și  $p2$  două valori naturale reprezentând două poziții din tablou ( $0 \leq p1 \leq p2 < n$ ). Numerotarea pozițiilor din tablou începe de la 0. Subprogramul returnează valoarea -1 dacă cele mai multe valori din tabloul  $a$ , aflate între pozițiile  $p1$  și  $p2$  inclusiv, sunt strict negative, valoarea 0 dacă cele mai multe valori din  $a$ , aflate între pozițiile  $p1$  și  $p2$  inclusiv, sunt nule sau valoarea 1 dacă cele mai multe valori din tabloul  $a$  aflate între pozițiile  $p1$  și  $p2$  inclusiv, sunt strict pozitive.

a) Scrieți definiția completă a funcției  $f$ .

(5p.)

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, fiecare element având cel mult patru cifre.

Funcția returnează media aritmetică a elementelor pare din tablou sau valoarea 0 dacă nu există elemente pare.

(10p.)