

4. 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=(1,21,9,21,403)**, **a=21** ale parametrilor, în urma apelului, subprogramul **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 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.)
4. 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**, iar altfel afișează mesajul **NU**. (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 cel mult 8 cifre, dintre care cel puțin una impară, și care determină și afișează pe ecran, folosind apeluri utile ale subprogramului **cif**, cel mai mare număr natural care poate fi obținut utilizând toate cifrele impare ale numărului **n**.
- Exemplu:** dacă **n=2152331** atunci se va afișa pe ecran numărul 53311. (6p.)
4. 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 $a=3^2 \cdot 5$, 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 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 o singură linie a ecranului, separate prin câte un spațiu. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.
- Exemplu:** pentru **n=8**, șirul: 1125, 2, 314, 101, 37, 225, 15, 12 se va afișa:
- 2 101 37 (6p.)
4. Se consideră șirul definit de următoarea relație de recurență:
- $$f_n = \begin{cases} n, & \text{dacă } n \leq 5 \\ 2 * f_{n-1}, & \text{dacă } n > 5 \end{cases}$$
- a) Scrieți numai antetul unui subprogram **p**, care primește prin intermediul parametrului **n** un număr natural de maximum 8 cifre, și care returnează cel mai mare termen al șirului de mai sus 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 realizează scrierea lui **s** ca sumă de termeni distincți ai șirului dat, folosind apeluri utile ale subprogramului **p**. Numerele se vor scrie în fișierul **Numere.txt**, pe prima linie a acestuia, separate prin câte un spațiu.
- Exemplu:** dacă valoarea citită de la tastatură este 63, atunci fișierul **Numere.txt** va avea următorul conținut: 40 20 3. (6p.)

4. Se consideră definite următoarele subprograme:
 - **p1** care primește prin intermediul parametrului **n** un număr natural cu cel mult 8 cifre și returnează suma cifrelor numărului primit prin parametrul **n**
Exemplu: dacă **n** este egal cu 1234 valoarea returnată de subprogram va fi 10.
 - **p2** care primește prin intermediul parametrului **n** un număr natural cu cel mult 8 cifre, elimină ultima cifră a acestui număr și returnează noul număr obținut.
Exemplu: dacă **n** este egal cu 1234 valoarea returnată de subprogram va fi 123.
 - a) Scrieți numai antetul subprogramelor **p1** și **p2**. (4p.)
 - b) Scrieți un program C/C++ care citește de la tastatură un număr natural **nenul** **n** cu cel mult 8 cifre și determină, prin apeluri utile ale subprogramelor **p1** și **p2**, fără a accesa cifrele numărului **n**, numărul de cifre egale cu 0 din scrierea lui **n**. Programul va afișa pe ecran numărul obținut.
Exemplu: dacă **n** este 102030, programul va afișa valoarea 3. (6p.)
3. a) Scrieți definiția completă a unui subprogram **P** cu 3 parametri, care primește prin intermediul primului parametru, **a**, un tablou unidimensional de cel mult 100 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 dintre valorile a_1, a_2, \dots, a_k din tablou.
Exemplu: pentru $k=5$ și tabloul $a = (7, 3, 8, 4, 6, 9, \dots)$, 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 tablou unidimensional de **n** numere întregi, cu proprietatea că valoarea termenului de pe poziția **i** ($i = 1, 2, \dots, n$) din acest tablou 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 tabloul 4 6 6 7 8 8 8 8 8 9 10 10. (6p.)
4. Se consideră subprogramul **nr** cu doi parametri, care primește prin parametrul **n** un număr natural cu maximum 8 cifre, și prin parametrul **c** o cifră zecimală. Subprogramul va returna numărul de apariții ale cifrei **c** în scrierea numărului **n**.
Exemplu: dacă $n=15356$, iar $c=5$, subprogramul va returna valoarea 2.
 - a) Scrieți doar antetul subprogramului **nr**. (3p.)
 - b) Scrieți programul C/C++ care citește de la tastatură un număr natural **n**, cu cel mult 8 cifre, și afișează pe ecran numărul de cifre distincte ale numărului **n**. Se vor folosi apeluri utile ale funcției **nr**. (7p.)**Exemplu:** pentru $n=15356$ se va afișa valoarea 4 deoarece numărul conține 4 cifre distincte și anume 1, 3, 5 și 6.
4. 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. (4p.)
 - 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, separate prin câte un spațiu. Se vor folosi apeluri utile ale subprogramului **P**. Dacă un număr din fișierul **BAC.IN** nu conține nicio cifră pară nenulă, 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.
4. 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.)

4. 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 C/C++ 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.)
3. 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.)
4. Pe prima linie a fișierului **numere.txt** sunt memorate cel mult 90 de numere naturale, mai mici decât 1000 fiecare, separate prin câte un spațiu. Se cere să se determine, apelând subprogramul **interval** definit la punctul anterior, numărul de elemente din fișier care sunt mai mici sau egale cu ultimul număr din fișier.
Exemplu: dacă fișierul **numere.txt** conține, în această ordine, numerele 6 267 13 45 628 7 79 101 se va afișa 6.
- a) Descrieți succint, în limbaj natural, strategia de rezolvare. (4p.)
- b) Scrieți programul C/C++ corespunzător algoritmului ales. (6p.)
3. 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.)
4. În fișierul text **nr1.txt** sunt memorate pe prima linie două valori, **n** și **m**, de cel mult 3 cifre fiecare, separate prin spațiu, iar pe fiecare dintre următoarele **n** linii sunt memorate câte **m** numere reale, separate prin câte un spațiu. Se cere scrierea în fișierul text **nr2.txt** a numerelor de pe fiecare dintre cele **n** linii ale fișierului **nr1.txt**, numerele de pe fiecare linie fiind aranjate astfel încât valorile pozitive să se afle la început și cele negative la sfârșit. Ordinea în cadrul secvenței de valori pozitive, respectiv în cadrul secvenței de valori negative, poate fi oricare. Cele **m** numere de pe o linie a fișierului **nr1.txt** se vor afișa pe câte o linie a fișierului **nr2.txt**, separate prin câte un spațiu. Se vor folosi apeluri utile ale subprogramului **aranjare**, definit la punctul anterior.
Exemplu: pentru fișierul: **nr1.txt** se poate obține fișierul **nr2.txt**
- | | |
|--------------|--------------|
| 3 5 | |
| 3 -6 -1 9 2 | 3 9 2 -6 -1 |
| 5 -3 -8 -4 7 | 5 7 -3 -8 -4 |
| -7 6 9 3 -2 | 6 9 3 -7 -2 |
- a) Descrieți succint, în limbaj natural, strategia de rezolvare. (4p.)
- b) Scrieți programul C/C++ corespunzător algoritmului ales. (6p.)
3. 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.)

3. 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 $n=4$ și $x=(1,2,3,4)$, 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 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). (6p.)
3. 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 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. (4p.)
3. 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.)
4. a) Scrieți definiția completă a unui subprogram **sterge**, cu trei parametri, care:
 - primește prin parametrii:
 - **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**.
 Tabloul modificat este furnizat tot prin parametrul **v**. (10p.)
- 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. (6p.)
3. 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 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.)

4. 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 text **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 (dacă există), 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
 100 14 21 8 77 35 atunci se afișează pe ecran 14 21 77 35 (10p.)
3. 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 $1 \leq i \leq j \leq n$.
 Subprogramul returnează suma elementelor $v_1, \dots, v_{i-1}, v_{j+1}, \dots, v_n$ din tabloul **v**. (10p.)
3. Scrieți în limbajul C/C++ definiția completă a subprogramului **ordonare** care primește ca parametru un tablou unidimensional **x** cu cel mult 100 de elemente, numere reale, și un număr natural **n** ($n \geq m$), ce reprezintă numărul efectiv de elemente ale tabloului **x**. Subprogramul va afișa tabloul obținut în urma schimbării poziției doar a elementelor impare din tablou astfel încât acestea să apară în ordinea crescătoare a valorilor lor. (10p.)
Exemplu: pentru $n=6$ și $x = (7, 11, 2, -8, -3, 10)$
 subprogramul va afișa $-3, 7, 2, -8, 11, 10$
3. Scrieți definiția completă a subprogramului **prime** care are ca parametri două numere naturale **x** și **y** (având cel mult 4 cifre fiecare) și afișează pe ecran toate numerele prime din intervalul închis, având unul din capete în **x** și celălalt în **y**. Numerele afișate vor fi separate prin câte un spațiu.
Exemplu: pentru $x=32$ și $y=18$ valorile afișate vor fi 19 23 29 31. (10p.)
3. Scrieți definiția completă a subprogramului **impar** care primește prin parametrul **x** un tablou unidimensional cu cel mult 100 de elemente numere naturale, fiecare având cel mult 9 cifre, iar prin parametrul **n** o valoare naturală reprezentând numărul efectiv de elemente ale tabloului ($1 \leq n \leq 100$) și afișează mesajul **DA** în cazul în care printre elementele tabloului **x** se află cel puțin un număr impar, sau afișează mesajul **NU** în caz contrar. (10p.)
3. Subprogramul **cifre_impere** primește prin parametrul **n** un număr natural având cel mult 9 cifre și returnează valoarea 1 dacă numărul **n** are toate cifrele impare și 0 în caz contrar.
- a) Scrieți numai antetul subprogramului **cifre_impere**. (4p.)
- b) În fișierul **numere.txt** se află memorat pe prima linie un număr natural **n**, iar pe următoarele linii, **n** numere naturale. Toate numerele din fișier au cel mult 9 cifre. Scrieți programul C/C++ care, folosind apeluri utile ale subprogramului **cifre_impere**, afișează pe ecran acele numere din fișier care sunt mai mari decât 100 și au toate cifrele impare. Numerele afișate sunt separate prin câte un spațiu.
- Exemplu:** dacă fișierul **numere.txt** are conținutul alăturat, se vor afișa pe ecran:
- | | |
|------------------|-----------|
| 10 | |
| 1 333 709 23 258 | |
| 34 713 2678 | |
| 333 713 11573 | 11573 778 |
- (10p.)
3. a) Scrieți definiția completă a unui subprogram recursiv **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 ($7=2+5$). (4p.)
- 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, suma divizorilor săi proprii și afișează pe ecran sumele determinate, în ordinea crescătoare a valorilor lor, separate prin câte un spațiu. (6p.)
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.

4. 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**. (4p.)
- 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 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 la subprogramul **cifra**, cel mai mare număr care se poate forma preluând ultima cifră pară a fiecărui element, dacă o astfel de cifră există. În cazul în care toate valorile din fișier conțin doar cifre impare, atunci se va afișa valoarea -1.
- Exemplu:** dacă fișierul **bac.in** are conținutul
- | | |
|---------------------------------------|------------------------------|
| 7 | |
| alăturat, pe ecran se va afișa: 64220 | (6p.) 369 113 2 0 33 1354 42 |
4. 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**. (4p.)
- 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**, pentru fiecare valoare din șir numărul de divizori primi. Numerele afișate vor fi separate prin câte un spațiu. (6p.)
- Exemplu:** dacă fișierul **bac.in** are conținutul alăturat, pe
- | | |
|--------------------------------|--------------------|
| 6 | |
| ecran se va afișa: 3 3 3 2 2 1 | 30 105 140 10 20 5 |
3. Scrieți în limbajul C/C++ definiția completă a subprogramului **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 $y=6$. (10p.)
4. 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++ numai antetul subprogramului **inter**. (4p.)
- 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**. (6p.)
4. 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**. (4p.)
- b) Considerăm un număr natural nenul $n > 99$ cu cel mult 9 cifre. Să se realizeze un program C/C++ care citește numărul n și care, folosind apeluri utile ale subprogramului **pr**, afișează pe ecran, separate prin câte un spațiu, doar valorile prime din șirul numerelor obținute din n , prin eliminarea succesivă a ultimei cifre, apoi a ultimelor două cifre, apoi a ultimelor trei cifre etc., până se obține un număr de două cifre, ca în exemplu.
- 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. (6p.)
4. Considerăm definite subprogramele:
- **pr**, care primește prin intermediul parametrului **x** un număr natural cu cel mult 4 cifre și returnează 1 dacă numărul este prim și 0 în caz contrar;
 - **sdiv**, care primește prin intermediul parametrului **y** un număr natural cu cel mult 4 cifre și returnează suma tuturor divizorilor numărului **y**.
- a) Scrieți numai antetul subprogramelor **pr** și **sdiv**. (4p.)
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural n ($n < 1000$) și care afișează pe ecran toate numerele mai mici decât n cu proprietatea că suma divizorilor lor este număr prim. Se vor utiliza apeluri utile ale subprogramelor **pr** și **sdiv**.
- Exemplu:** dacă $n=20$, atunci programul va afișa: 2 4 9 16. (6p.)

3. Să se scrie 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. Iar altfel subprogramul returnează valoarea 0. (10p.)

4. 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**. (4p.)
b) Scrieți în limbajul C/C++ definiția completă a unui subprogram cu un singur parametru, **x**, prin intermediul căruia primește un număr natural cu cel mult 9 cifre și care returnează valoarea 1 dacă în scrierea în baza 10 a lui **x** se găsește cel puțin o cifră care să fie media aritmetică a celorlalte cifre din componența lui **x** și 0 în caz contrar, folosind apeluri utile ale subprogramului **cifre**, de la punctul a.
Exemplu: pentru **n=27989** subprogramul va returna valoarea 1 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 returnează 0. (6p.)

4. 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.)

3. Se consideră subprogramul **cmdiv** 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 **cmdiv**. (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 **cmdiv**. 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.)

4. 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.)

3. Se consideră subprogramul **divxy** 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 **divxy**. (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 **divxy**. 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.)

3. 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 **date.in** și afișează pe ecran suma obținută adunând, pentru fiecare dintre cele **n** numere citite de pe a doua linie din fișier, cel mai mic divizor prim. Se vor utiliza apeluri utile ale subprogramului **divizor**.
- Exemplu:** dacă fișierul **date.in** are conținutul alăturat, se
- | | |
|----|--------------|
| 6 | |
| 16 | 25 6 77 10 4 |
- va afișa 20 ($20 = 2 + 5 + 2 + 7 + 2 + 2$). (6p.)
3. 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
- | | |
|----|---------------------------|
| 6 | |
| 16 | 1775 333 242477 123 55566 |
- alăturat, atunci se vor afișa numerele: (6p.)
3. 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
- | | |
|----|----------------|
| 7 | |
| 16 | 25 15 12 7 4 5 |
- afișa 4 pentru că pechile (16,25), (12,7), (7,4), (4, 5) sunt formate din numere prime între ele. (6p.)
3. Se consideră subprogramul **pal**, care primește prin intermediul primului parametru, **a**, un număr natural, cu minimum 2 cifre și maximum 8 cifre, și furnizează prin intermediul celui de-al doilea parametru, **b**, cel mai apropiat număr de valoarea lui **a**, care este palindrom. În cazul în care există 2 astfel de numere, subprogramul va returna numărul mai mic. Un număr natural **x** este palindrom dacă este egal cu numărul obținut prin scrierea cifrelor lui **x** în ordine inversă.
- Exemplu:** dacă **a=16**, atunci după apel **b=11**; dacă **a=126**, atunci după apel **b=121**; dacă **a=33**, atunci după apel **b=33**.
- a) Scrieți definiția completă a subprogramului **pal**. (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 nenule, separate prin câte un spațiu, fiecare număr având minimum 2 cifre și maximum 8 cifre. Scrieți un program **C/C++** care citește toate numerele din fișierul text **date.in** și afișează pe ecran, despărțite prin câte un spațiu, pentru fiecare dintre cele **n** numere, cel mai apropiat număr palindrom, folosind apeluri utile ale subprogramului **pal**.
- Exemplu:** dacă fișierul **date.in** are conținutul alăturat, atunci
- | | |
|----|--------------|
| 4 | |
| 16 | 1775 333 190 |
- se vor afișa numerele: 11 1771 333 191 (6p.)

3. 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 parametru **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 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 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 | |
4. 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 ale 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.)
4. 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, ($x \leq y$), 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**. (6p.)
4. 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.)
4. 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.)

4. 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.)
3. 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**. (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 le afișează pe ecran, separate prin câte un spațiu, numai pe acelea care au primele trei cifre impare. Se vor utiliza apeluri utile ale subprogramului **verif**. Dacă nu există niciun număr cu această proprietate, se va afișa pe ecran mesajul **nu**. Alegeți o metodă eficientă din punctul de vedere al memoriei utilizate.
Exemplu: dacă fișierul **date.txt** conține numerele alăturate, pe ecran se afișează:
- | | | |
|--------|--------|--------|
| 133579 | 973314 | |
| | | (6p.) |
| | | 3 |
| | | 133579 |
| | | 345796 |
| | | 973314 |
- c) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)
3. 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 a_i este divizor al lui a_j , sau a_j este divizor al 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.)
3. 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: dacă **x**=24973 se va returna valoarea 1.
- a) Scrieți definiția completă a subprogramului **sfx**. (10p.)
- b) Fișierul text **date.in** conține cel mult 10000 de numere naturale de exact 6 cifre fiecare, separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier, determină și afișează pe ecran câte dintre aceste numere au toate cifrele în ordine strict descrescătoare. Programul va folosi apeluri utile ale subprogramului **sfx**. Se va utiliza un algoritm eficient din punctul de vedere al memoriei utilizate. (6p.)
Exemplu: dacă fișierul **date.in** conține
- | | |
|--------|-----------------------|
| 236543 | pe ecran se afișează: |
| 865210 | 2 |
| 976532 | |
- c) Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei (3 – 4 rânduri). (4p.)
3. 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 cel mai mare divizor comun al elementelor tabloului **a**. (10p.)
Exemplu: în urma apelului, pentru **n**=5 și tabloul unidimensional (12,36,48,6,60), se va returna 6.
3. 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: pentru **n**=5 și tabloul unidimensional (12,37,43,6,71) în urma apelului se va returna 3.

3. 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` ($1 \leq n \leq 10000$ și $1 \leq k \leq 5$), 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 ale lui `n`, atunci funcția returnează valoarea -1. (10p.)
Exemplu: în urma apelului `calcul(9243, 3)` se va returna 2.
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.
3. 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 nicio cifră pară `x` primește valoarea 0. (10p.)
Exemplu: în urma apelului `calcul(9278, x)`, `x` primește valoarea 82.
3. 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.)
4. Fișierul text `numere.in` conține mai multe numere naturale de cel mult 9 cifre fiecare, numerele fiind despărțite prin câte un spațiu.
a) Scrieți un program C/C++ care citește numerele din fișierul `numere.in` și, folosind apeluri utile ale funcției `Palindrom`, definită la punctul 3, determină în mod eficient care este cel mai mare număr palindrom citit. Programul scrie în fișierul text `numere.out` numărul astfel determinat.
Exemplu: dacă `numere.in` conține numerele:
 23 565 78687 7887 7865 78687 7887 23 78687 98798
 atunci `numere.out` va conține:
 78687 (6p.)
b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită (3 – 4 rânduri). (4p.)
3. 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. (10p.)
Exemplu: dacă `a = 123446` și `b = 248766` atunci subprogramul va returna 3, iar dacă `a = 1244` și `b = 4456` subprogramul va returna 1.
4. Fișierul text `numere.in` conține pe prima linie a sa două numere naturale, `x`, cu cel mult nouă cifre, și `p`, cu exact o cifră, iar pe a doua linie cel mult 1000 de numere naturale de cel mult nouă cifre fiecare.
a) Scrieți programul C/C++ care citește numerele din fișierul `numere.in` și, apelând funcția `Cifre`, definită la punctul 3, scrie în fișierul text `numere.out` acele numere de pe a doua linie a fișierului `numere.in` care au exact `p` cifre comune cu numărul `x`. Numerele vor fi scrise pe aceeași linie, despărțite prin câte un spațiu. (6p.)
b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită (3 – 4 rânduri) (4p.)
Exemplu:
- | | |
|----------------------------------|-------------------------|
| <code>numere.in</code> | <code>numere.out</code> |
| 345638 3 | 45239 900583 635 |
| 123 45239 96583 900583 635 78900 | |

3. Scrieți definiția completă a subprogramului **Insereaza**, cu patru parametri, **n**, **x**, **k** și **v**, care primește prin intermediul parametrului **n** un număr natural ($0 < n < 1000$), prin intermediul parametrului **x** un tablou unidimensional cu **n** elemente, numere reale, numerotate de la 1 la **n**, prin parametrul **k** un număr natural reprezentând o poziție din tablou ($1 \leq k \leq n$), iar prin parametrul **v** un număr real. Subprogramul va insera valoarea **v** pe poziția **k** a tabloului **x** și furnizează tabloul modificat, precum și numărul elementelor acestuia, actualizate. **(10p.)**

4. Fișierul text **sir.in** conține cel mult 100 numere naturale de maximum patru cifre fiecare, numerele fiind despărțite prin câte un spațiu.

a) Scrieți un program C/C++ care citește numerele din fișierul **sir.in** și, folosind apeluri ale subprogramului **Insereaza**, definit la punctul 3, construiește în memorie un tablou unidimensional care va conține toate numerele din fișierul **sir.in**, iar între fiecare două numere consecutive din fișier inserează media lor aritmetică. Programul scrie în fișierul text **sir.out** elementele tabloului obținut, cu cel mult o zecimală, câte 10 elemente pe fiecare rând (cu excepția ultimului rând, care poate să conțină mai puține), elementele de pe același rând fiind despărțite prin câte un spațiu. **(6p.)**

Exemplu: dacă fișierul **sir.in** conține numerele: 7 5 635 456 0 8 587

atunci, după executarea programului, fișierul **sir.out** va conține:

```
7 6 5 320 635 545.5 456 228 0 4
8 297.5 587
```

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită (3 – 4 rânduri). **(4p.)**

3. 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.)**

4. Fișierul text **NUMERE.IN** conține pe prima linie un număr natural **n** ($0 < n < 1000$), iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale cu cel mult nouă cifre fiecare.

a) Scrieți un program C/C++ care citește de la tastatură o cifră **c**, și, folosind apeluri utile ale subprogramului **Cifra**, definită la punctul 3, determină pentru câte dintre numerele de pe a doua linie a fișierului **NUMERE.IN**, cifra **c** reprezintă cifra maximă. Programul scrie pe primul rând al fișierul text **NUMERE.OUT** mesajul **DA** dacă există astfel de valori și pe al doilea rând numărul acestora, iar dacă nu există va scrie în fișier mesajul **NU**. **(6p.)**

b) Descrieți succint, în limbaj natural, metoda de rezolvare folosită (3 – 4 rânduri). **(4p.)**

Exemplu: dacă cifra citită este 8

NUMERE.IN conține

```
10
725 3695 423 0 7895 0 100 101 870 568
```

NUMERE.OUT va conține

```
DA
2
```

4. 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 **n** citite, utilizând pentru determinarea ambelor 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.)

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

3. Scrieți definiția completă a unui subprogram cu numele **patrat** care primește prin intermediul parametrului **n** ($n < 100$) un număr natural și prin intermediul parametrului **v** un tablou unidimensional 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.)**

4. a) Scrieți definiția completă a unui subprogram cu numele **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.)
- b) Scrieți un program C/C++ care citește de pe prima linie a fișierului text **numere.in** un număr natural **n** ($0 < n < 1000$) și de pe a doua linie a fișierului **n** numere naturale cu cel mult 9 cifre și afișează pe ecran, despărțite prin câte un spațiu, toate numerele naturale conținute de a doua linie a fișierului care au cifrele ordonate strict crescător sau strict descrescător, utilizând apeluri utile ale subprogramului **ordonat**. (6p.)
- Exemplu:** dacă fișierul **numere.in** are următorul conținut:
- ```
5
134 6420 1243 9802 731
```
- pe ecran se va afișa:
- ```
134 6420 731.
```
3. 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 cele **n** elemente, numerotate de la 1 la **n**, ale unui tablou unidimensional. Elementele sunt numere naturale, cu cel mult 4 cifre fiecare. Programul 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.)
3. 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.)
4. 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**. (4p.)
- 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**. (6p.)
- Exemplu:** dacă fișierul
- | | | | |
|--------------------------|-------|---------------------|-----|
| NUMERE.IN are conținutul | 12 14 | atunci fișierul | 84 |
| alăturat: | 11 12 | NUMERE.OUT va avea | 132 |
| | 4 8 | următorul conținut: | 8 |
3. 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** fiind 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.)
3. 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 **n=149448**, **c1=4** și **c2=2**, valoarea furnizată prin **n** va fi 129228. (10p.)

3. 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.)
3. 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 tabloul ordonat tot prin parametrul **x**.
 a) Scrieți numai antetul acestui subprogram. (4p.)
 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. (10p.)
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
3. 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, 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**. (10p.)
3. 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.)
3. 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.)
3. 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.)

Subprograme 2008

3. 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). **(10p.)**