

SUBIECTE PROPUSE PENTRU
EXAMENUL DE ATESTAT PROFESIONAL LA INFORMATICĂ - 2014
Programare Pascal/C/C++
Matematică-informatică

1. Se citesc din fișierul "pol.in", de pe prima linie, gradele a două polinoame n, m ($0 \leq n, m \leq 50$). De pe următoarele două linii se citesc două șiruri de numere reale care reprezintă coeficienții celor două polinoame, în ordinea crescătoare a gradelor. Să se scrie un program care să construiască în memorie și să afișeze în fișierul "pol.out" suma polinoamelor. Coeficienții sumei se vor afișa cu două zecimale separați printr-un singur spațiu.

Exemplu:

| | |
|--|----------------------------------|
| pol.in 2 3 7 -1 1 5 3 -4 1 | pol.out 3 12 2 -3 1 |
|--|----------------------------------|

2. Se citesc din fișierul "pol.in", de pe prima linie, gradele a două polinoame n, m ($0 \leq n, m \leq 50$). De pe următoarele două linii se citesc două șiruri de numere reale care reprezintă coeficienții celor două polinoame, în ordinea crescătoare a gradelor. Să se scrie un program care să construiască în memorie și să afișeze în fișierul "pol.out" produsul polinoamelor. Coeficienții produsului se vor afișa cu două zecimale separați printr-un singur spațiu .

Exemplu:

| | |
|---|-----------------------------------|
| pol.in 3 1 1 5 -2 1 2 1 | pol.out 4 2 11 1 0 1 |
|---|-----------------------------------|

3. Se citesc din fișierul "sort.in", de pe prima linie, 2 numere n și m ($1 \leq n, m \leq 50$). De pe următoarele 2 linii se citesc elementele a două șiruri de lungime n , respectiv m , , numere întregi, de maxim 9 cifre, care sunt ordonate crescător. Să se scrie un program care să construiască în memorie un șir care conține rezultatul interclasării celor două șiruri. Șirul rezultat se va afișa în fișierul "sort.out".

Exemplu:

| | |
|---|--------------------------------------|
| sort.in 5 4 1 3 4 7 8 1 2 3 6 | sort.out 1 1 2 3 3 4 6 7 8 |
|---|--------------------------------------|

4. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere n și m ($1 \leq n, m \leq 50$). De pe următoarele două linii se citesc elementele a două șiruri de lungime n , respectiv m , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Scrieți un program, care să construiască în

memorie un șir ce va conține intersecția celor două șiruri. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

| mult.in | mult.out |
|----------------|-----------------|
| 5 4 | 1 3 |
| 1 3 4 7 8 | |
| 1 2 3 6 | |

5. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere n și m ($1 \leq n, m \leq 50$). De pe următoarele două linii se citesc elementele a două șiruri de lungime n , respectiv m , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Să se scrie un program care să construiască în memorie un șir ce va conține diferența dintre primul șir și cel de-al doilea șir. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

| mult.in | mult.out |
|----------------|-----------------|
| 5 4 | 4 7 8 |
| 1 3 4 7 8 | |
| 1 2 3 6 | |

6. Se citesc din fișierul "mult.in", de pe prima linie, 2 numere n și m ($1 \leq n, m \leq 50$). De pe următoarele două linii se citesc elementele a două șiruri de lungime n , respectiv m , numere întregi, de maxim 9 cifre. Elementele șirurilor sunt distincte două câte două. Să se scrie un program, care să construiască în memorie un șir ce va conține reuniunea celor două șiruri. Rezultatul se va afișa în fișierul "mult.out".

Exemplu:

| mult.in | mult.out |
|----------------|-----------------|
| 5 4 | 1 2 3 4 6 7 8 |
| 1 3 4 7 8 | |
| 1 2 3 6 | |

7. Se citește din fișierul "mult.in", de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc elementele unui șir de lungime n , numere întregi, de maxim 9 cifre. Să se scrie un program care să verifice dacă șirul citit conține numai elemente distincte.

Exemplu:

| mult.in | |
|----------------|----|
| 5 | da |
| 1 3 4 7 8 | |

8. Se citesc din fișierul "sort.in", de pe prima linie, un număr n . De pe următoarea linie se citesc elementele unui șir de lungime n . Să se scrie un program care sortează crescător elementele prime ale șirului, valorile neprime pastrandu-si pozitia initiala . Rezultatul se va afișa în fișierul "sort.out".

Exemplu:

| | |
|-------------------------------------|-------------------------------------|
| sort.in 6 8 13 4 5 2 1 | sort.out 8 2 4 5 13 1 |
|-------------------------------------|-------------------------------------|

9. Se citește din fișierul “sort.in”, de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc elementele unui șir de lungime n , numere întregi, de maxim 9 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda selecției. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

| | |
|--------------------------------------|----------------------------------|
| sort.in 7 3 8 4 1 9 2 4 | sort.out 1 2 3 4 4 8 9 |
|--------------------------------------|----------------------------------|

10. Se citește din fișierul “sort.in”, de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc elementele unui șir de lungime n , numere întregi, de maxim 9 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda inserției. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

| | |
|--------------------------------------|----------------------------------|
| sort.in 7 3 8 4 1 9 2 4 | sort.out 1 2 3 4 4 8 9 |
|--------------------------------------|----------------------------------|

11. Se citește din fișierul “sort.in”, de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc elementele unui șir de lungime n , numere întregi, de maxim 9 cifre. Să se scrie un program care să realizeze sortarea crescătoare a elementelor șirului folosind metoda numărării. Se va ține cont de faptul că în șir poate să apară un element și de mai multe ori. Rezultatul se va afișa în fișierul “sort.out”.

Exemplu:

| | |
|--------------------------------------|----------------------------------|
| sort.in 7 3 8 4 1 9 2 4 | sort.out 1 2 3 4 4 8 9 |
|--------------------------------------|----------------------------------|

12. Se citește din fișierul “cmmdc.in”, de pe prima linie, un număr natural n ($1 \leq n \leq 50$). De pe următoarea linie se citesc n numere naturale de maxim 9 cifre.. Să se scrie un program care să calculeze cel mai mare divizor comun al celor n numere de pe linia a doua a fișierului. Rezultatul se va afișa în fișierul “cmmdc.out”.

Exemplu:

| | |
|--------------------------------------|-----------------------|
| cmmdc.in 5 14 8 22 4 32 | cmmdc.out 2 |
|--------------------------------------|-----------------------|

13. Se citește din fișierul “exista.in”, de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc n numere naturale de maxim 9 cifre reprezentând elementele unui șir. Să se scrie un program care să verifice dacă există în șir trei elemente alăturate care au același număr de divizori.

Exemplu:

| | |
|--|----|
| exista.in 8 9 4 15 6 22 12 3 14 | Da |
|--|----|

14. Se citește din fișierul “progresie.in”, de pe prima linie, un număr n ($1 \leq n \leq 50$). De pe următoarea linie se citesc elementele unui șir de lungime n , numere întregi, de maxim 9 cifre.. Să se scrie un program care să verifice dacă elementele șirului formează o progresie aritmetică sau o progresie geometrică. În caz afirmativ să se afișeze tipul progresiei și rația, iar în caz negativ să se scrie un mesaj corespunzător în fișierul ”progresie.out”.

Exemplu:

| | |
|---|---|
| progresie.in 5 2 6 18 54 162 | progresie.out elementele șirului formează o progresie geometrică cu rația 3 |
|---|---|

15. Se citește un text din fișierul ”text.in”. Textul are mai multe linii și mai multe cuvinte. Pe fiecare linie cuvintele sunt despărțite prin câte un singur spațiu sau prin semne de punctuație. Să se afișeze în fișierul ”text.out”, pe linii consecutive, toate literele care apar în text, împreună cu frecvențele lor de apariție.

Exemplu:

| | |
|--|--|
| text.in Mama are un mar. Ce mare e marul! | text.out A – 6 C – 1 E – 4 M – 5 N – 1 R – 4 U - 2 |
|--|--|

16. Să se afișeze în fișierul “prietene.out” toate perechile de numere prime prietene mai mici decât un număr natural n citit de la tastatură ($1 \leq n \leq 30000$). Două numere prime se numesc prietene dacă diferența lor în modul este mai mică sau egală cu 2.

Exemplu:

| | |
|----|--|
| 20 | prietene.out (2 3), (3 5), (5 7), (11 13), (17 19) |
|----|--|

17. Se citește un număr natural n de la tastatură ($1 \leq n \leq 30000$). Să se verifice dacă n este număr puternic. Un număr este puternic dacă are mai mulți divizori decât orice număr mai mic decât el.

Exemplu:

| | |
|----|----|
| 6 | DA |
| 10 | NU |

18. Se citește de la tastatură un număr natural k ($1 \leq k \leq 50$). Să se determine cel mai mic număr natural care are exact k divizori.

Exemplu:

| | |
|---|----|
| 5 | 16 |
|---|----|

19. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural n ($1 \leq n \leq 50$) și de pe următoarele n linii o matrice pătratică de numere întregi, de maxim 9 cifre. Să se verifice dacă matricea este pătrat magic (sumele elementelor pe linii, coloane și de pe cele două diagonale, este aceeași).

Exemplu

| | |
|---|----|
| mat.in 3 4 3 8 9 5 1 2 7 6 | da |
|---|----|

20. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural n ($1 \leq n \leq 50$), și de pe următoarele n linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se numere elementele cu număr par de cifre aflate în triunghiul de sub diagonala principală a matricei și să se afișeze rezultatul pe ecran.

Exemplu

| | |
|--|---|
| mat.in 4 1 4 12 3 13 3 22 5 14 22 2 34 222 11 3 31 | 4 |
|--|---|

21. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural n ($1 \leq n \leq 50$), și de pe următoarele n linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se calculeze suma elementelor prime aflate în triunghiul de deasupra diagonalei principale a matricei și să se afișeze rezultatul pe ecran.

Exemplu

| | |
|--|---|
| mat.in 4 2 4 12 3 15 3 22 5 16 22 2 34 222 11 3 31 | 8 |
|--|---|

22. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural n ($1 \leq n \leq 50$), și de pe următoarele n linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se verifice dacă există elemente cu suma cifrelor pară în triunghiul de sub diagonală secundară a matricei și să se afișeze rezultatul pe ecran.

Exemplu

| | |
|--|----|
| mat.in 4 7 11 5 8 9 5 1 12 3 4 31 15 3 2 23 14 | Da |
|--|----|

23. Se citește din fișierul “mat.in”, de pe prima linie, un număr natural n , ($1 \leq n \leq 50$) și de pe următoarele n linii o matrice pătratică ce conține numere naturale de maxim 4 cifre. Să se verifice dacă toate elementele aflate în triunghiul de deasupra diagonalei secundare a matricei au aceeași paritate și să se afișeze rezultatul pe ecran.

Exemplu

| | |
|--|----|
| mat.in 4 7 11 5 8 9 5 1 12 3 4 31 15 3 2 23 14 | Da |
|--|----|

24. Se citesc din fișierul “mat1.in” de pe prima linie, două numere naturale $n1$ ($1 \leq n1 \leq 50$) și $m1$ ($1 \leq m1 \leq 50$) și de pe următoarele $n1$ linii o matrice cu $n1$ linii și $m1$ coloane. Analog, se citesc din fișierul “mat2.in”, de pe prima linie, două numere naturale $n2$ ($1 \leq n2 \leq 50$) și $m2$ ($1 \leq m2 \leq 50$) și de pe următoarele $n2$ linii o matrice cu $n2$ linii și $m2$ coloane. Elementele celor 2 matrice sunt numere reale. Să se scrie un program în care se calculează produsul celor două matrice. Dacă produsul se poate calcula atunci se va afișa matricea rezultată în fișierul “mat.out”, iar în caz contrar se va afișa un mesaj corespunzător.

Exemplu

| | | |
|---|--|---------------------------------|
| mat1.in 2 3 3 1 2 5 2 3 | mat2.in 3 2 5 1 0 2 1 1 | mat.out 13 7 28 12 |
|---|--|---------------------------------|

25. Se citesc din fișierul “mat.in”, de pe prima linie, două numere naturale n și m ($1 \leq n, m \leq 50$) și de pe următoarele n linii o matrice cu n linii și m coloane cu elemente numere întregi, de maxim 9 cifre. Să se scrie un program care afișează în fișierul “mat.out”, câte unul pe linie, numerele din matrice care sunt în același timp maxime pe linia și minime pe coloana pe care se află. Dacă nu există astfel de numere se va afișa un mesaj corespunzător.

Exemplu

| Mat.in | | mat.out |
|---------------|---|----------------|
| 4 | 3 | 3 |
| 1 3 2 2 | | |
| 2 7 8 1 | | |
| 4 8 5 2 | | |

26. Se citesc din fișierul “mat.in” , de pe prima linie, două numere naturale n și m ($1 \leq n, m \leq 50$) și de pe următoarele n linii o matrice cu n linii și m coloane cu elemente numere naturale, de maxim 9 cifre. Să se scrie un program care afișează în fișierul “mat.out”, câte unul pe linie, numerele din matrice care au număr maxim de divizori

Exemplu

| Mat.in | mat.out |
|---------------|----------------|
| 1 4 | 14 |
| 9 3 14 10 | 10 |
| 8 5 6 21 | 8 |
| 11 3 15 17 | 6 |
| | 21 |
| | 15 |

27. Se citesc din fișierul “mat.in” , de pe prima linie, patru numere naturale n , m , l și c ($1 \leq n, m \leq 50$, $1 \leq l \leq n$, $1 \leq c \leq m$) și de pe următoarele n linii o matrice cu n linii și m coloane, cu elemente numere întregi, de maxim 9 cifre. Să se scrie un program care să șteargă din matrice linia l și coloana c și afișează matricea rezultată în fișierul “mat.out”.

Exemplu

| Mat.in | mat.out |
|---------------|----------------|
| 1 4 2 3 | 1 2 4 |
| 1 2 3 4 | 9 10 12 |
| 5 6 7 8 | |
| 9 10 11 12 | |

28. Se citește un număr natural n ($1 \leq n \leq 15$). Să se afișeze în fișierul “Pascal.out” triunghiul lui Pascal care are $n + 1$ linii și $n + 1$ coloane. Regulile după care se construiește triunghiul lui Pascal sunt:
- Pe coloana 0 și pe diagonala principală se pune 1.
 - Restul elementelor din triunghi se obțin adunând elementul aflat pe linia anterioară pe aceeași coloană cu el cu elementul aflat pe linia anterioară și pe coloana anterioară.

Exemplu

| | Pascal.out |
|---|-------------------|
| 3 | 1 |
| | 1 1 |
| | 1 2 1 |
| | 1 3 3 1 |

29. Se citesc de la tastatură numitorii și numărătorii (numere întregi de maxim 9 cifre) a două fracții. Să se scrie un program care simplifică suma celor două fracții și afișează pe ecran numitorul și numărătorul fracției rezultate.

Exemplu

| | |
|---------|-----|
| 1 2 1 3 | 5 6 |
|---------|-----|

30. Se citesc de la tastatură două cuvinte de maxim 20 de litere. Să se scrie un program care verifică dacă unul este anagrama celuilalt. Un cuvânt este anagramă pentru altul dacă cele două cuvinte conțin aceleași litere de același număr de ori, dar nu neapărat în aceeași ordine.

Exemplu

| | |
|--------------|----|
| mare rame | Da |
| mare mere | Nu |