29.

Se consideră două tablouri unidimensionale A şi B cu elemente numere naturale din intervalul [1,10000]. Spunem că tabloul A “se poate reduce” la tabloul B dacă există o împărţire a tabloului A în secvenţe disjuncte de elemente aflate pe poziţii consecutive în tabloul A astfel încât prin înlocuirea secvenţelor cu suma elementelor din secvenţă să se obţină, în ordine, elementele tabloului B.

De exemplu tabloul A

**A: 7 3 4 1 6 4 6 9 7 1 8 7**

se poate reduce la tabloul

**B: 14 7 26 16**

Fişierul text **NUMERE.IN**conţine pe prima linie două numere naturale nenule n şi m (1≤m≤n≤100), pe linia a doua m numere naturale din intervalul [1;10000] şi pe linia a treia alte m numere naturale din intervalul [1;10000]. Pe fiecare linie numerele sunt separate prin câte un spaţiu.

Scrieţi un program C/C++ care citeşte toate numerele din fişierul **NUMERE.IN** şi verifică, utilizând un algoritm eficient din punctul de vedere al timpului de executare, dacă tabloul construit cu cele n numere aflate pe linia a doua în fişier se poate reduce la tabloul construit cu cele m numere aflate pe linia a treia în fişier. Programul afişează pe ecran mesajul DA în caz afirmativ şi mesajul NU în caz negativ.

28.

a) Scrieţi definiţia completă a unui subprogram primul, care - primeşte prin singurul său parametru, a, o valoare naturală din intervalul [2,10000]- returnează o valoare naturală reprezentând **cel mai mic divizor al numărului a mai mare strict decât 1**.

b) Fişierul text NUMERE.IN conţine pe prima linie un număr natural nenul n (1≤n≤100) şi pe următoarea linie n numere naturale din intervalul [2,10000] separate prin câte un spaţiu.

Un număr natural n se numeşte „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 şi 7.

Scrieţi un program C/C++ care determină, folosind apeluri utile ale suprogramului primul, cel mai mare număr „ aproape prim” de pe linia a doua a fişierului NUMERE.IN.

În cazul în care există un astfel de număr se afişează pe ecran mesajul DA,

urmat de numărul determinat, iar în caz contrar mesajul NU.

Exemplu:

Dacă fişierul NUMERE.IN are conţinutul:

**6**

**100 14 21 8 77 35**

atunci se afişează pe ecran

**DA 77**

pentru că numărul 77este cel cel mai mare dintre numerele„aproape prime” din fişier(14=7\*2,21=7\*3,77=7\*11, 35=7\*5)

30

Fişierul text NUMERE.IN conţine pe prima linie un număr natural nenul n (1≤n≤100)şi pe următoarea linie n numere reale pozitive ordonate crescător, separate prin câte un spaţiu.

a)Scrieţi un program C/C++care citeşte din fişierul

NUMERE.IN numărul natural n, şi determină, utilizând un algoritm eficient din punct de vedere al timpului de executare şi al memoriei utilizate, numărul minim de intervale închise de forma [x,x+1], cu x număr natural, a căror reuniune include toate numerele reale din fişier.

Exemplu:

Dacă fişierul NUMERE.IN are conţinutul:

6

2.3 2.3 2.8 5.7 5.7 6.3

atunci se afişează

3 (intervalele [2,3], [5,6], [6,7] sunt cele 3 intervale de forma

cerută care conţin numere din şir).

31

În fişierul numere.txt se află memorate, pe prima linie un număr natural **n** (1≤n≤100), iar pe fiecare dintre următoarele n linii, câte două numere întregi x,y (-100≤x≤y≤100), reprezentând capetele câte unui segment [x,y] desenat pe axa Ox de coordonate.

Scrieţi în limbajul C/C++ un program eficient din punct de vedere al timpului de executare şi al spaţiului de memorare, care citeşte din fişier datele existente, determină segmentul rezultat în urma intersecţiei tuturor celor n segmente date şi afişează pe ecran două numere despărţie printr-un spaţiu ce reprezintă capetele segmentului cerut.

Dacă segmentele nu au nici un punct comun se va afişa pe ecran valoarea 0.

Exemplu:dacă fişierul **numere.txt** are conţinutul alăturat,

5

-7 10

3 20

-5 5

0 12

-8 30

se va afişa pe ecran 3 5.

32

În fişierul numere.txt sunt memorate pe mai multe linii, numere întregi (cel mult

100), numerele de pe aceeaşi linie fiind despărţite prin câte un spaţiu, fiecare număr având cel mult 9 cifre. Să se determine cele mai mici două valori având exact două

cifre fiecare, memorate în fişier şi să se afişeze pe ecran aceste valori, despărţite printr-un spaţiu. Dacă în fişier nu se află două astfel de valori, pe ecran se va afişa valoarea 0.

Scrieţi programul C/C++ corespunzător metodei descrise la punctul a.

Exemplu:dacă fişierul numere.txt are conţinutul alăturat,

5 10

3 -77 20

50 5 0 12 18 30

se va afişa pe ecran, nu neapărat în această ordine: -77 10

1. Se citeşte un vector de n caractere, ale cărui elemente sunt literele unui cuvânt. Să se verifice dacă este palindrom.
2. Scrieţi o funcţie care verifică dacă un şir de numere reale formează sau nu o progresie aritmetică.
3. Să se afişeze toate permutările circulare la stânga ale unui şir de n elemente.
4. Se dă un şir cu elemente numere întregi. Să se scrie un program care calculează maximul din suma a câte k elemente alăturate ale şirului.

Ex: pentru k=3 si (12,-3,20,11,4,3,13,-4,3) ⇒ max=25

1. Fie două şiruri inegale de lungime m respectiv n. Să se însumeze elementele celor două şiruri într-un şir nou. Elemetele rămase fără corespondent se copiază la sfârşitul şirului nou format.
2. Se dau două şiruri an şi bn. Să se formeze din ele un nou şir cn după formula:

Se dau două şiruri xn şi yn. Să se formeze



1. Să se ordoneze descrescător într-un şir de numere întregi, doar numerele pozitive.
2. Se consideră un şir de numere întregi, ordonat crescător, care are elemente care se repetă. Să se afle cel mai lung subşir "constant" (format din aceleaşi elemente).

Ex. pentru şirul 3, 4, 4, 4, 5, 6, 6, 7, 7, 7, 9, 11, 11, 11, 11, 11, 15, 15, 17, 19, 19 cel mai lung subşir este format din numărul 11 şi are 5 apariţii.

1. Se citesc două mulţimi A şi B de litere (fiecare într-un vector de valori diferite). Să se afişeze mulţimea C=A∪B (reuniunea mulţimilor).
2. Se citesc două mulţimi A şi B de litere (fiecare într-un vector de valori diferite). Să se afişeze mulţimea C=A∩B (intersecţia mulţimilor).
3. Se citesc douǎ mulţimi A şi B de litere (fiecare într-un vector de valori diferite). Să se afişeze mulţimea C=AXB (produsul cartezian al mulţimilor).