Nr. 15 / 04.01.2023

Vizat,
Inspector Școlar General Adjunct,
Prof. Răzvan Delcea VASILE

SUBIECTELE PROBEI PRACTICE PENTRU EXAMENUL DE ATESTAT PROFESIONAL LA INFORMATICĂ, ANUL ȘCOLAR 2022 – 2023

Matematică informatică - intensiv informatică

Subiectul 1

În fişierul ***Numere.txt*** pe prima linie este memorat un număr natural **n** (**n<10000**), iar pe linia următoare un şir de **n** numere naturale distincte cu maximum **4** cifre fiecare, separate prin câte un spaţiu. Se cere:

a) Afişaţi pe prima linie a fişierului de ieşire ***Rezultat.out*** poziţia pe care s-ar găsi primul element din şirul aflat pe linia a doua a fişierului, daca şirul ar fi ordonat crescător. Numerotarea poziţiilor elementelor în cadrul şirului este de la **1** la **n**. Alegeţi un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate şi al timpului de executare.
b) Verificaţi dacă primul număr de pe a doua linie din fişier este număr **prim** şi afişaţi un mesaj corespunzător, pe a doua linie a fişierului de ieşire (**DA** sau **NU**). Veţi folosi un subprogram care primeşte ca parametru un număr natural **x,** de cel mult **4** cifre si returnează **TRUE** daca numărul **x** este prim sau **FALSE** in caz contrar.

Exemplu: Dacă fişierul ***Numere.txt*** are următorul conţinut:
6
267 13 45 628 7 79

Fişierul ***Rezultat.out*** va avea următorul conţinut:
5
NU

deoarece numărul **267** ar ocupa poziţia a cincea în şirul ordonat crescător (**7 13 45 79 267 628**) şi 267 nu este număr prim.

Subiectul 2

Fişierul text ***Numere.txt*** conţine pe prima linie numărul natural **n**, (**1≤n≤30000)**, pe următoarele **n** linii un şir de **n** numere naturale, iar pe ultima linie două numere naturale **a** şi **b** (**a≤b**) separate de un spaţiu. Fiecare dintre cele **n** numere, precum şi valorile **a** şi **b**, au cel mult două cifre. Se cere:

a) **A**fişaţi pe prima linie in fişierul ***Rezultat.out*** cel mai mic număr întreg din intervalul închis **[a,b]** care se găseşte în şirul dat. Dacă nu există un astfel de număr, programul afişează textul **NU EXISTA**. Alegeţi un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate şi al timpului de executare.
b) Pe a doua linie din fişierul de ieşire afişaţi **c.m.m.m.c** dintre **a** şi **b**. Pentru calculul celui mai mic multiplu comun a doua numere se va utiliza un subprogram care primeşte ca parametrii două numere naturale **x** şi **y** şi returnează **c.m.m.d.c**. al lor. Reamintim ca **c.m.m.m.c(x,y)**= $\frac{x * y}{c.m.m.d.c.(x, y)}$

Exemplu: Dacă fişierul ***Numere.txt*** are următorul conţinut:

4
7
20
11
35
9 21
Fişierul ***Rezultat.out*** va avea următorul conţinut :
11
63

Subiectul 3

Fişierul text ***Numere.in*** conţine pe prima linie un număr natural nenul **n,** (**2≤n≤100**) şi pe următoarea linie **n** numere reale pozitive, în ordine **strict crescătoare**, separate prin câte un spaţiu.

a) **U**tilizând un algoritm eficient din punct de vedere al execuţiei şi al memoriei utilizate, determinaţi şi afişaţi pe prima linie în fişierul ***Numere.out*** cel mai mare număr natural **x,** cu proprietatea că în orice interval **deschis** având drept capete două numere de pe poziţii alăturate dintre cele **n** numere aflate pe linia a doua în fişierul ***Numere.in*** se găsesc cel puţin **x** numere întregi.

b) Afişaţi numărul de cifre distincte din numărul **x** utilizând un subprogram care primeşte prin intermediul parametrului **y** un număr natural si returnează prin cel de-al doilea parametru **z** numărul cifrelor distincte ale numărului **y.**

Exemplu: Dacă fişierul ***Numere.in*** are următorul conţinut:

5
3.5 323.1 549 925.3 1312.7

Fişierul ***Numere.out*** va avea următorul conţinut :
225
2
Explicaţie: In intervalul **(3.5, 323.1)** exista **320** de numere întregi, in **(323.1 , 549)** exista **225** de numere întregi, in **(549, 925.3)** exista **376** de numere întregi, in **(925.3, 1312.7)** există **387** numere întregi, deci in oricare dintre intervale exista cel puţin **225** de numere întregi. Numărul **225** este format din două cifre distincte.

Subiectul 4

Se consideră fişierul text ***Date.in*** ce conţine pe prima linie două numere naturale nenule, **n** şi **s (n≤9, s<20)**, iar pe a doua linie **n** numere întregi, separate prin câte un spaţiu, fiecare număr având maximum **9** cifre nenule.

a) Afişaţi în fişierul de ieşire ***Date.out*** , despărţite prin câte un spaţiu, numerele situate pe a doua linie a fişierului ***Date.in***, a căror sumă a cifrelor este mai mică decât **s, ordonate strict crescător**. În cazul în care nu există nici un astfel de număr în fişierul de ieşire se va afişa valoarea **0**. Veţi folosi un subprogram numit ***cifre***, care primeşte prin intermediul primului parametru, **a**, un număr întreg cu maximum **9** cifre nenule şi returnează, prin intermediul celui de-al doilea parametrul **b**, suma cifrelor lui **a.**

b) Pe o doua linie a fişierului ***Date.out*** afişaţi cel mai mic număr natural format din cifrele distincte ale numerelor afişate pe prima linie in fişierul ***Date.out***.

Exemplu: Dacă fişierul ***Date.in*** are următorul conţinut:

6 18
321 175 999 242477 16 -269

Fişierul ***Date.out*** va avea următorul conţinut :
-269 16 175 321
1235679

Subiectul 5

Fişierul text ***Date.in*** conţine cel puţin două şi cel mult **10000** de numere naturale distincte, dintre care cel puţin două sunt pare. Numerele sunt separate prin câte un spaţiu şi fiecare dintre ele are cel puţin **3** si cel mult **9** cifre.

a) Determinaţi cele mai mari două numere pare din fişier, utilizând un algoritm eficient din punct de vedere al timpului de executare şi al spaţiului de memorie utilizat. Cele două numere vor fi afişate în pe prima linie a fişierului de ieşire ***Rezultate.out***, în ordine descrescătoare, separate printr-un spaţiu.

b) Construiţi in memorie o matrice pătratică cu **n** linii si **n** coloane, unde **n** este numărul de cifre al primului număr scris în fişierul de ieşire. Matricea va fi construită astfel: pe diagonala principală va conţine **0**, deasupra diagonalei principale cea mai mare cifră, iar sub diagonala principală cea mai mică cifră a acestui număr. Matricea va fi afişată pe următoarele **n** linii în fişierul ***Rezultate.out***, elementele fiecărei linii fiind separate de câte un singur spaţiu. Veţi utiliza un subprogram care primeşte ca parametru un număr natural **a** de cel mult **9** cifre şi returnează prin intermediul parametrilor **n, max, min** numărul de cifre, cifra maximă, respectiv cifra minimă a numărului **a**.

Exemplu: dacă fişierul ***Date.in*** are următorul conţinut:
5123 610 301 122 824

Fişierul ***Rezultate.out*** va avea următorul conţinut:

824 610
0 8 8
2 0 8
2 2 0

Subiectul 6

Fişierul text ***Numere.txt*** conţine pe prima linie un număr natural **n** (**0<n<100000**), iar pe a doua linie **n** numere naturale, formate din cel mult **4** cifre, separate prin câte un spaţiu.

a) Determinaţi în mod eficient, din punct de vedere al memoriei şi a timpului de executare, cifrele ce apar în scrierea numerelor situate pe a doua linie a fişierului. Programul va afişa în fişierul de ieşire ***Rezultate.out*** cifrele (în ordine crescătoare) precum şi numărul de apariţii al acestora (pe fiecare linie cifra şi numărul ei de apariţii separate de un singur spaţiu).

b) Verificaţi dacă suma cifrelor impare este un număr palindrom, şi afişaţi pe ecran un mesaj corespunzător (**DA** sau **NU**). Veţi utiliza un subprogram care primeşte ca parametru un număr natural **x** de cel mult **4** cifre şi returnează inversul numărului **x.**

Exemplu: dacă fişierul ***Numere.txt*** are următorul conţinut:
7

243 32 545 74 12 1344 90
Fişierul ***Rezultate.out*** va avea următorul conţinut:
0 1
1 2
2 3
3 3
4 5
5 2
7 1
9 1
Pe ecran se va afişa **NU** (deoarece suma cifrelor impare **2*1+3*3+2*5+1*7+1*9 = 37** care nu este un număr palindrom)

Subiectul 7

În fişierul ***Numere.txt*** pe prima linie sunt memorate două numere naturale **n** şi **m** (1<n,m<51), iar pe următoarele n linii câte m numere naturale distincte cu maxim **4** cifre fiecare, separate prin câte un spaţiu. Se cere:

c) Afişaţi pe prima linie a fişierului de ieşire ***Rezultat.out*** toate numerele din fişierul de intrare care au cifra de control pară. Numerele se vor afişa separate prin câte un spaţiu, în ordinea apariţiei lor în fişierul de intrare. **Cifra de control** a unui număr se obţine calculând suma cifrelor numărului, apoi suma cifrelor sumei şi tot asa până la obţinerea unei singure cifre. În cazul în care nici un număr nu are cifra de control pară în fişierul de ieşire se va afişa mesajul **NU EXISTA**. Veţi folosi un subprogram care primeşte ca parametru un număr natural **x**, cu cel mult 4 cifre şi returnează cifra de control a valorii reţinute de **x**.

d) În cazul în care pe prima linie a fişierului de ieşire au fost afişate valori numerice, atunci să afişeze pe a doua linie a fişierului de ieşire cel mai mare număr natural format din cifrele distincte ale numerelor de pe prima linie a fişierului **Rezultat.out**. Alegeţi un algoritm de rezolvare eficient din punct de vedere al memoriei utilizate şi al timpului de executare. Dacă pe prima linia a fişierului de ieşire s-a afişat NU EXISTA, atunci pe a doua linie a fişierului de ieşire nu se va afişa nimic.

Exemplu: Dacă fişierul ***Numere.txt*** are următorul conţinut:
2 3
123 12 3
524 171 11
Fişierul ***Rezultat.out*** va avea următorul conţinut:
123 524 11
54321

Subiectul 8

Fişierul text ***Matrice.txt*** conţine pe prima linie un număr natural **n (0<n<=50),** şi un număr natural **k (0<k<=9)** iar pe fiecare dintre următoarele **n** linii câte **n** numere naturale de cel mult 9 cifre, separate prin câte un spaţiu, numere care reprezintă elementele unei matrice pătratice de dimensiune **n.**

a) Construiţi un vector cu **n** elemente, fiecare element $\mathbf{v_i}$ (i=1,2,...,n) al vectorului conţine **minimul** dintre elementele liniei **i,** care conţine exact **k** cifre de **1** în reprezentarea sa binară, **sau** valoarea **zero** dacă nu există un astfel de element pe linia **i**. Afişaţi elementele vectorului astfel construit, separate de câte un spaţiu pe prima linie în fişierul ***Rezultate.out.***

Se va utiliza un subprogram ***cifunu*** care primeşte ca parametru un număr natural **x,** şi returnează numărul de cifre **1** din reprezentarea binară a numărului **x.**

b) Pe a doua linie a fişierului de ieşire ***Rezultate.out*** afişaţi cel mai mare număr obţinut din cifrele elementului maxim din vectorul format

Exemplu: dacă fişierul ***Matrice.txt*** are următorul conţinut:

4 3
11 12 3 15
8 2 17 3
5 0 7 10
13 11 155 16

Fişierul ***Rezultate.out*** va avea următorul conţinut:

11 0 7 11
11

Subiectul 9

Fişierul ***Cuvant.in*** conţine cuvinte formate doar din litere mici, separate prin unul sau mai multe spaţii (numărul cuvintelor poate să difere de la un rând la altul; o linie din fişier conţine cel mult 255 de caractere).

Definim **randamentul** unui cuvânt ca fiind **numărul de caractere distincte din cuvânt / lungimea cuvântului**. De exemplu şirul '**caractere**' are randamentul **5/9=0.56.**

a) Afişaţi în fişierul ***Rezultate.out***, pentru fiecare linie citită din fişierul de intrare cuvintele în ordinea apariţiei lor, precum şi randamentul fiecărui cuvânt. Veţi folosi un subprogram care primeşte ca parametru un şir de caractere **s** returnează printr-un al doilea parametru numărul de caractere distincte ale lui **s.**

b) Pe ultimul rând în fişierul de ieşire afişaţi cuvântul cu randamentul cel mai mare. Dacă în fişier există mai multe astfel de cuvinte afişaţi-l pe ultimul.

Exemplu: dacă fişierul ***Cuvânt .in*** are următorul conţinut:

caractere litere mici
problemele pentru concurs sunt grele
flori galbene

Fişierul ***Rezultate.out*** va avea următorul conţinut:
caractere 0.56 litere 0.83 mici 0.75
problemele 0.60 pentru 1.00 concurs 0.86 sunt 1.00 grele 0.80
flori 1.00 galbene 0.86
flori

Subiectul 10

Se citesc de la tastatură un număr natural nenul **n, (n<10)** şi un număr natural **k (k<=9).**

a) Construiţi un tablou pătratic a, unde $\mathbf{a_{i\ j}}$ este al **i+j-1** lea număr prim (dacă numerotăm indicii matricei 1..n) sau al **i+j+1** lea număr prim (dacă numerotăm indicii matricei 0..n-1). Afişaţi tabloul astfel construit în fişierul ***Matrice.out,*** pe primele n linii ale fişierului, elementele unei linii fiind separate de un singur spaţiu. Utilizaţi un subprogram ***prim***, care primeşte ca şi parametru un număr natural **x**, şi returnează valoarea **1** dacă numărul transmis ca parametru este prim, şi **0** în caz contrar.

b) Afişaţi pe următoarea linie în fişierul de ieşire, în ordine crescătoare toate elementele tabloului astfel construit care au exact **k** cifre distincte. Dacă nu există astfel de numere în tabloul construit afişaţi în fişier mesajul **NU EXISTĂ.**

Exemplu: dacă **n=4** şi **k=2**
Fişierul ***Matrice.out*** va avea următorul conţinut:

2 3 5 7
3 5 7 11
5 7 11 13
7 11 13 17
13 17
OBS. Pentru **n=4 şi k=4** pe ultima linie din fişier se va afişa **NU EXISTĂ**
Considerăm că numărul 2 este primul număr prim.

Subiectul 11

Fişierul ***Date.in*** conţine cel mult **10000** numere naturale cu cel mult **2** cifre fiecare, printre care cel puţin un număr par şi cel puţin un număr impar, separate prin câte un spaţiu. Numărul de numere din fişier este variabil de la o linie la alta.

a) Scrieţi un program care citeşte numerele din fişierul ***Date.in*** şi scrie în fişierul text ***Date.out*** valorile distincte citite, separate prin câte un spaţiu, respectându-se regula: pe prima linie vor fi scrise numerele **impare distincte** în **ordine crescătoare**, iar pe linia a doua numerele **pare distincte**, în ordine **descrescătoare**. Alegeţi o metodă eficientă din punctul de vedere al timpului de executare.
b) Calculaţi suma numerelor pare distincte din fişierul de intrare şi eliminaţi din numărul obţinut cifrele impare. Veţi utiliza un subprogram care primeşte ca parametru un număr natural **x** de cel mult **8** cifre şi returnează prin intermediul aceluiaşi parametru numărul obţinut după eliminarea cifrelor impare. Rezultatul se va tipări pe a treia linie a fişierului de ieşire.

Exemplu: Dacă fişierul ***Date.in*** are următorul conţinut:
75 12 3
3 18
75 1 3 92
atunci fişierul ***Date.out*** va conţine:
Fişierul ***Date.out*** va avea următorul conţinut:
1 3 75
92 18 12
22
deoarece suma numerelor pare este **122**, iar după eliminarea cifrei **1** care e impară se obţine **22.**

Subiectul 12

În fişierul ***Cuvant.in*** pe prima linie este memorat un număr **n** (1<n<51) şi pe următoarele n linii cuvinte separate prin unul sau mai multe spaţii (numărul cuvintelor poate să difere de la un rând la altul; o linie din fişier conţine cel mult 255 caractere). Se cere:

a) Afişaţi în fişierul de ieşire ***Prefix.out*** pe fiecare linie toate prefixele celui mai lung cuvânt de pe linia corespunzătoare din fişierul ***Cuvant.in***. Dacă sunt mai multe cuvinte de aceeaşi lungime se vor scrie prefixele ultimului cuvânt dintre cele care îndeplinesc condiţia. ***Prefixele*** unui cuvânt **x** sunt reprezentate de toate subşirurile şirului **x** (care încep cu primul caracter din x), de la cel de lungime 1 până la cel de lungimea şirului x – 1. Veţi folosi un subprogram care primeşte ca parametru un şir de caractere şi determină afişarea în fişierul de ieşire a tuturor prefixelor acelui şir.
b) Pe ultima linie a fişierului de ieşire se vor afişa toate cuvintele din fişierul de intrare care încep cu litera a sau A. În cazul în care în fişierul de intrare nu există astfel de cuvinte, în fişierul de ieşire se va afişa mesajul **NU EXISTA**.

Exemplu: Dacă fişierul ***Cuvant.in*** are următorul conţinut:
3
Ana cascada lupta
feriga parada alama
rusine ard jucarie

Fişierul ***Prefix.out*** va avea următorul conţinut:
c ca cas casc casca cascad
p pa par para parad
j ju juc juca jucar jucari
Ana alama ard

Subiectul 13

Fişierul text ***Numere.in*** conţine cel mult **1000** de numere naturale cu cel mult patru cifre fiecare, despărţite prin câte un spaţiu.

a) Scrieţi programul care citeşte numerele din fişierul ***Numere.in*** şi afişează în fişierul de ieşire ***Rezultate.out,*** în ordine **crescătoare**, acele numerele din fişierul de intrare care au toate cifrele egale. Pentru a verifica dacă toate cifrele unui număr sunt egale veţi folosi un subprogram care primeşte ca parametru un număr natural **x** de cel mult patru cifre şi returnează numărul de cifre distincte ale lui **x**. Dacă fişierul nu conţine nici un astfel de număr, atunci se va scrie în fişierul de ieşire mesajul **NU EXISTA**.
b) Daca există astfel de numere afişaţi factorii primi împreună cu exponenţii lor, din descompunerea în factori primi ai celui mai mare număr care are toate cifrele egale. Afişarea se va face in fişierul ***Rezultate.out***, începând cu cea dea doua linie, fiecare factor şi exponentul său pe câte o linie în fişier separaţi de câte un spaţiu.

Exemplu: dacă fişierul ***Numere.in*** are următorul conţinut:
30 11 444 7 25 5

Fişierul ***Rezultate.out*** va avea următorul conţinut:

5 7 11 444
2 2
3 1
37 1

Cel mai mare număr care are toate cifrele egale este **444** iar **444= 2^2 + 3^1 + 37^1**

Subiectul 14

O matrice pătratică **A** de dimensiune **n** cu **p** elemente nenule este memorată economic în fişierul de intrare ***Matrice.txt*** sub următoarea formă.: pe prima linie a fişierului se găsesc două numere **n** şi **p**, dimensiunea matricei respectiv numărul de elemente nenule, iar pe următoarele **p** linii triplete de numere naturale **(v, l, c**) care reprezintă valoarea, linia respectiv coloana pe care se găsesc elementele nenule.

a) Scrieţi un program care citeşte informaţiile din fişierul de intrare, reface şi scrie în fişierul de ieşire ***Matrice.out*** matricea **A.**
b) Verificaţi dacă suma elementelor nenule ale matricei este un număr **perfect** (un număr este considerat perfect dacă este egal cu suma tuturor divizorilor săi cu excepţia lui însuşi ex: **6 =1+2+3 ; 28=1+2+4+7+14**) şi afişaţi pe ecran un mesaj corespunzător (**DA** sau **NU**). Pentru a verifica daca un număr este perfect veţi utiliza un subprogram care primeşte prin intermediul unui parametru un număr natural **x** şi returnează suma divizorilor săi, excepţie numărul.

Exemplu: dacă fişierul ***Matrice.txt*** are următorul conţinut:
5 8
1 1 1
5 1 2
1 2 3
3 2 5

4 3 2
4 4 1
2 4 3
8 5 4

Fişierul ***Matrice.out*** va avea următorul conţinut:

1 5 0 0 0
0 0 1 0 3
0 4 0 0 0
4 0 2 0 0
0 0 0 8 0

Pe ecran se va afişa **DA** (deoarece suma este **1+5+1+3+4+4+2+8=28** care este un număr perfect)

Subiectul 15

Fişierul text ***Numere.txt*** conţine pe prima linie un număr natural **k (0<k<15)** si pe următoarele rânduri cel mult **50.000** de numere naturale din intervalul închis **[0, 99]**, numerele de pe acelaşi rând fiind separate prin câte un spaţiu.

a) Scrieţi un program care afişează in fişierul de ieşire ***Numere.out,*** în ordine **crescătoare**, separate de cate un spaţiu, acele numere din fişierul ***Numere.txt*** care au cel puţin **k** divizori proprii. Utilizaţi un algoritm eficient din punct de vedere al timpului de executare. Dacă un număr care corespunde cerinţei apare de mai multe ori, se va afişa o singură dată.

b) Verificaţi dacă cifrele ultimului număr scris în fişierul de ieşire ***Numere.out*** au aceeaşi paritate (toate pare sau toate impare), şi afişaţi pe ecran un mesaj corespunzător (**DA** sau **NU**), Veţi folosi un subprogram care primeşte prin intermediul unui parametru un număr natural **n** din intervalul **[0, 99]** şi returnează atât numărul de cifre pare cât şi numărul de cifre impare ale numărului **n.**

Exemplu: dacă fişierul ***Numere.txt*** are următorul conţinut:
4
15 36 33
36 1 12 1 24
2
fişierul ***Numere.out*** va conţine :
Fişierul ***Numere.out*** va avea următorul conţinut:

12 24 36

Pe ecran se va afişa **NU** (deoarece **3** şi **6** nu sunt de aceeaşi paritate)

Subiectul 16

În fişierul ***Cuvant.txt*** pe prima linie sunt memorate două numere **p** (1<p<51) şi **n** (1<n<101), pe a doua linie un cuvânt **cuv** şi pe următoarele **n** linii cuvinte separate prin unul sau mai multe spaţii (numărul cuvintelor poate să difere de la un rând la altul; o linie din fişier conţine cel mult 255 caractere). Se cere:

a) Afişaţi în fişierul de ieşire ***Rime.txt*** pe fiecare linie toate cuvintele de pe linia corespunzătoare din fişierul de intrare care sunt rime cu cuvântul **cuv** (două cuvinte sunt rime dacă ultimele **p** caractere coincid). Cuvintele de pe aceeaşi linie se vor afişa separate prin câte un spaţiu. Dacă pe o linie nu există nici un cuvânt care să îndeplinească condiţia atunci pe linia corespunzătoare în fişierul ***Rime.txt*** se va afişa mesajul **NU EXISTA**. Veţi folosi un subprogram care primeşte ca parametrii două şiruri de caractere şi un număr natural **p**. Subprogramul va returna valoarea **1** dacă cele două şiruri de caractere sunt rime (au ultimele **p** caractere identice) sau valoarea **0** în caz contrar.

b) Pe ultima linie a fişierului de ieşire se va afişa numărul cuvintelor din fişierul de intrare care încep cu majusculă (nu se va număra cuvântul de pe a doua linie a fişierului de intrare).

Exemplu: Dacă fişierul ***Cuvant.txt*** are următorul conţinut:
2 4
ferit
Sarit cascada indraznit
feriga fugit alama
rusine Ranit Jucarie
Albastru

Fişierul ***Rime.txt*** va avea următorul conţinut:
Sarit indraznit
fugit
Ranit
NU EXISTA
4

Subiectul 17

Fişierul ***atestat.in*** conţine un text format din cel mult 250 de caractere, scris pe o singură linie. Cuvintele din text sunt separate prin câte un spaţiu, iar fiecare cuvânt este format din cel mult 20 caractere literele mici ale alfabetului englez.

a) Să se scrie în fişierul **atestat.out** textul în care fiecare cuvânt care conţine cel puţin o consoană să fie înlocuit de inversul său. Se va utiliza un subprogram s1 care primeşte prin intermediul singurului său parametru *s* un cuvânt şi returnează numărul de consoane din cuvântul *s*.
b) Pe a doua linie în fişierul atestat.out să se scrie cuvântul de lungime maximă şi lungimea lui. Dacă sunt mai multe astfel de cuvinte, se va afişa primul găsit.

Exemplu: Dacă fişierul ***atestat.in*** are următorul conţinut:
fetita ia o ie si o da elenei

Fişierul ***atestat.out*** va avea următorul conţinut:
atitef ia o ie is o ad ienele
fetita 6

Subiectul 18

Se citesc de la tastatură două numere naturale **n (1≤n≤10)** şi **k (0<k<=10).**

a) Construiţi în memorie o matrice cu **n** linii şi **n** coloane, astfel încât parcurgând liniile matricei de sus în jos şi de la stânga la dreapta se obţin, în prima linie primele **n** numere ale şirului Fibonacci în ordine **crescătoare**, în linia a doua următoarele **n** numere ale şirului Fibonacci în ordine **descrescătoare**, în linia a treia următoarele **n** numere ale acestui şir în ordine **crescătoare**, şi aşa mai departe, ca în exemplu. Elementele şirului Fibonacci se obţin astfel: primul element este **0**, al doilea este **1**, iar elementele următoare se obţin însumând cele două elemente care preced elementul curent. Astfel, primele **16** elemente ale acestui şir sunt: **0**, **1**, **1**, **2**, **3**, **5**, **8**, **13**, **21**, **34**, **55**, **89**, **144**, **233**, **377**, **610**. Programul afişează in fişierul ***Matrice.out*** matricea obţinută, câte o linie a matricei pe câte o linie a fişierului, elementele fiecărei linii fiind separate prin câte un spaţiu.
b) Pe următoarea linie în fişier afişaţi produsul indicilor coloanelor pe care există cel puţin **k** elemente pare.

Exemplu: dacă se citesc de la tastatură n=4 şi k=2

Fişierul ***Matrice.out*** va avea următorul conţinut:
0 1 1 2

13 8 5 3
21 34 55 89
610 377 233 144
8
Pe ultima linie in fişier se va afişa **8** (deoarece coloanele **1, 2 şi 4** conţin cel puţin două elemente pare).

Subiectul 19

Un şir de caractere **s** se numeşte "**şablon**" pentru un alt şir de caractere **x**, dacă este format din caractere din mulţimea **{*, ?,#}**, are aceeaşi lungime cu **x** şi pe fiecare poziţie din **s** în care apare ***** în **x** se găseşte o vocală, pe fiecare poziţie din **s** în care apare **#** în **x** se găseşte o consoană şi pe fiecare poziţie din **s** în care apare **?** putem avea orice caracter în **x**. Se consideră vocală orice literă din mulţimea **{a,e,i,o,u}**.

a) Se citesc din fişierul ***Cuvinte.in*** două şiruri de caractere, de aceeaşi lungime, formate din cel mult **200** de litere mici ale alfabetului englez. Şirurile se găsesc în fişer fiecare pe câte o linie. Afişaţi pe ecran, un şablon **comun** celor două şiruri citite, care conţine un număr minim de caractere **?**.
b) Verificaţi dacă unul din cuvintele citite este anagrama celuilalt şi afişaţi pe ecran mesajul **DA** în caz afirmativ sau **NU** în caz contrar.

Exemplu: dacă fişierul ***Cuvinte.in*** are următorul conţinut:
diamant
pierdut
Pe ecran se va afişa
##??#**
NU

Subiectul 20

Se consideră un arbore cu rădăcină şi *n* vârfuri (n din N, 2 < n < 100), dat prin vectorul de taţi. Vârfurile sunt etichetate cu numere consecutive şi distincte: 1, 2, 3, .. , n. Fişierul **atestat.in** conţine 2 linii. Pe prima linie a fişierului este scris numărul n, reprezentând numărul de vârfuri, iar pe a doua linie sunt scrise n numere, separate prin spaţiu, reprezentând valorile consecutive ale vectorului de "taţi".

a) Să se scrie pe prima linie în fişierul **atestat.out** etichetele tuturor frunzelor arborelui.
b) Pe următoarele linii ale fişierului **atestat.out** să se scrie fiecare nod neterminal împreună cu numărul descendenţilor direcţi (câte un nod cu numărul fiilor săi pe câte un rând). Se va utiliza un subprogram **fii** cu trei parametrii n (numărul de vârfuri), t (vectorul de taţi) şi k (eticheta unui vârf). Subprogramul returnează numărul tuturor descendenţilor direcţi ai vârfului cu eticheta k.

Exemplu: Dacă fişierul ***atestat.in*** are următorul conţinut:
10
0 1 1 2 3 2 2 6 6 5

Fişierul ***atestat.out*** va avea următorul conţinut:
4 7 8 9 10
1 2
2 3
3 1
5 1
6 2