

AVIZAT,
Inspector școlar pentru informatică,
prof. Merlan Doina Narcisa



EXAMEN DE ATESTARE A COMPETENȚELOR PROFESIONALE A ABSOLVENȚILOR
CLASELOR DE MATEMATICĂ - INFORMATICĂ ȘI
MATEMATICĂ - INFORMATICĂ, INTENSIV INFORMATICĂ

SUBIECTE PROBA PRACTICĂ – 2021

○ PROGRAMARE

SUBIECTUL 1

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural n , nenul, $n \leq 10000$, și pe a doua linie un șir de n numere întregi, de cel mult 9 cifre fiecare, despărțite prin câte un spațiu,

a) Să se afișeze, în fișierul **atestat.out**, pe prima linie, valoarea minimă și poziția pe care se regăsește aceasta, pe a doua linie valoarea maximă din șir, precum și poziția pe care se regăsește aceasta. Dacă minimul (sau maximul) apare de mai multe ori în șir, se va afișa prima sa apariție. Valorile de pe aceeași linie vor fi despărțite prin câte un spațiu.

b) Să se afișeze, în fișierul **atestat.out**, pe a treia linie, cele mai mari două valori distincte din șir, în ordine crescătoare, separate prin câte un spațiu.

EXEMPLU:

atestat.in	atestat.out
10	1 7
4 2 9 100 56 32 1 8 61 39	100 4
	61 100

SUBIECTUL 2

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural n , nenul, $0 < n < 100$, și pe a doua linie un șir de n numere naturale a_1, a_2, \dots, a_n , de cel mult 9 cifre fiecare, despărțite prin câte un spațiu,

a) Să se afișeze, în fișierul **atestat.out**, pe prima linie, numărul de numere prime din șir.

b) Să se afișeze, în fișierul **atestat.out**, pe a doua linie, numărul cifrelor 0 în care se termină numărul natural $p = a_1 a_2 \dots a_n$.

EXEMPLU:

atestat.in	atestat.out
5	3
4 5 13 2 10	2

SUBIECTUL 3

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural n , $n \leq 1000000$, nenul, iar pe a doua linie un șir de n numere naturale, despărțite prin câte un spațiu.

a) Să se afișeze, în fișierul **atestat.out**, pe prima linie, cifra care apare de cele mai multe ori în scrierea numerelor citite. Dacă sunt mai multe astfel de cifre, se va afișa cea mai mare dintre ele.

b) Să se afișeze, în fișierul **atestat.out**, pe a doua linie, despărțite prin câte un spațiu, toate cifrele care nu apar în scrierea niciunui număr din șir.

EXEMPLU:

atestat.in	atestat.out
8	1
12 123 45 456 7 34 23 111	0 8 9

SUBIECTUL 4

Se citesc din fișierul **atestat.in** două numere naturale nenule n și k , $0 < n \leq 100000$, $0 < k \leq 100$, scrise pe două linii separate. Să se afișeze, în fișierul **atestat.out**,

a) pe prima linie, toate puterile numărului k mai mici sau egale cu n . Numerele vor fi afișate pe o linie, cu spațiu între ele.

b) în cazul în care k este număr prim, se va afișa, pe linia următoare din fișier, puterea la care apare k , în descompunerea lui n în factori primi. În cazul în care k nu este prim, se va afișa puterea la care apare 2 în descompunerea în factori primi a lui k .

EXEMPLU:

atestat.in	atestat.out
30	1 3 9 27
3	1

SUBIECTUL 5

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural n , nenul ($n \leq 1000$) și pe a doua linie un șir de n numere naturale de maxim 9 cifre fiecare, despărțite prin câte un spațiu. Să se afișeze, în fișierul **atestat.out**:

a) acele numere din șir care se divid cu suma cifrelor lor, afișate pe o singură linie, cu spațiu între ele.

b) pe linia următoare, cel mai mare divizor comun al numerelor din șir.

EXEMPLU:

atestat.in	atestat.out
6	111 24 100
111 24 15 26 100 99	1

SUBIECTUL 6

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural nenul n ($n \leq 1000$) și n perechi de numere naturale nenule de forma $(a \ b)$, scrise pe câte o linie, cu spațiu între elementele perechii. Fiecare număr din pereche are maxim 9 cifre. Să se afișeze, în fișierul **atestat.out**,

a) pe primele n linii, numărătorul și numitorul fracției ireductibile la care poate fi adusă fiecare fracție a/b .

b) pe linia $n+1$, numărul de fracții care erau deja ireductibile.

EXEMPLU:

atestat.in	atestat.out
6	
12 45	4 15
13 31	13 31
15 25	3 5
14 21	2 3
2 3	2 3
99 45	11 5
	2

SUBIECTUL 7

Se citește din fișierul **atestat.in**, un număr natural n , $0 \leq n \leq 50$. Să se afișeze, în fișierul **atestat.out**:

a) pe prima linie, al n -lea termen din sirul lui Fibonacci, folosind o funcție care returnează al n -lea element din acest șir.

b) pe a doua linie, valoarea termenului găsit la punctul a, în baza 2.

EXEMPLU:

atestat.in	atestat.out
6	8
	1000

SUBIECTUL 8

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural n , nenul, $n \leq 18$, și pe a doua linie un șir de n cifre, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:

a) pe prima linie, cel mai mare număr ce poate fi obținut cu cifrele numărului inițial.

b) pe a doua linie, separate prin câte un spațiu, toate numerele prime ce pot fi scrise folosind cifre din fișier. Fiecare cifră din fișier, se va folosi în scrierea numerelor prime, de un număr de ori mai mic sau egal cu numărul de apariții a sa în fișierul de intrare.

EXEMPLU:

atestat.in	atestat.out
4	5322
2 3 2 5	2 3 5 23 53 223 523

SUBIECTUL 9

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural $n \leq 10000$, nenul și pe a doua linie un șir de n numere întregi, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:

a) pe prima linie, toate numerele impare din fișier în ordine descrescătoare, separate prin câte un spațiu.

b) pe a doua linie din fișier, toate numerele din șir în ordine crescătoare, cele pare rămânând pe loc.

EXEMPLU:

atestat.in	atestat.out
6	35 23 15
12 23 -12 15 35	12 15 -12 23 35

SUBIECTUL 10

În fișierul **atestat.in** se găsesc, pe prima linie, un număr natural $n \leq 10000$, nenul și pe a doua linie un șir de n numere întregi, despărțite de câte un spațiu. Să se afișeze, în fișierul **atestat.out**:

a) pe prima linie, ordonate crescător toate elementele vectorului aflate între poziția maximului și poziția minimului, inclusiv acestea (Dacă valoarea maximă apare de mai multe ori se va considera prima apariție, iar dacă valoarea minimă va apărea de mai multe ori, se va considera ultima apariție a acesteia).

b) pe a doua linie, elementele vectorului după ștergerea tuturor aparițiilor primului element în vector.

EXEMPLU:

atestat.in	atestat.out
10	-12 -1 1 2 3 5 12 25
12 23 -12 12 5 3 2 1 -1 25	23 -12 5 3 2 1 -1 25

SUBIECTUL 11

Se dă o matrice pătratică cu n ($4 \leq n \leq 10$) elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul n , iar de pe următoarele n linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe prima, respectiv a doua linie, următoarele valori:

a) Media aritmetică a elementelor strict pozitive de deasupra diagonalei principale;

b) Numărul elementelor negative de sub diagonala principală.

EXEMPLU:

matrice.in	matrice.out
4	15
3 2 -2 5	3
4 -1 2 3	
2 1 -3 -1	
1 -3 2 4	

SUBIECTUL 12

Se dă o matrice pătratică cu n ($4 \leq n \leq 10$) elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul n , iar de pe următoarele n linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe primele n linii, respectiv pe următoarele n linii, următoarele valori:

a) Matricea obținută în urma inversării liniei **L1** cu linia **L2** din matricea dată (**L1** și **L2** citite de la tastatură)

b) Matricea obținută în urma inversării coloanei **C1** cu coloana **C2** din matricea inițială (**C1** și **C2** citite de la tastatură).

EXEMPLU:

matrice.in

```
4
3 2 -2 5
4 -1 2 3
2 1 -3 -1
1 -3 2 4
```

L1=1, L2=3
C1=2, C2=3

matrice.out

```
2 1 -3 -1
4 -1 2 3
3 2 -2 5
1 -3 2 4
3 -2 2 5
4 2 -1 3
2 -3 1 -1
1 2 -3 4
```

SUBIECTUL 13

Se dă o matrice pătratică cu n ($4 \leq n \leq 10$) elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul n , iar de pe următoarele n linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa, pe prima, respectiv a doua linie, valorile **DA** sau **NU**, în funcție de rezultatul următoarelor verificări:

a) Dacă matricea este simetrică față de diagonala principală, se va afișa **DA**, altfel **NU**;

b) Dacă matricea este simetrică față de diagonala secundară, se va afișa **DA**, altfel **NU**.

EXEMPLU:

matrice.in

```
4
3 2 2 1
2 -1 2 3
2 2 -3 -1
1 3 -1 4
```

matrice.out

```
DA
NU
```

SUBIECTUL 14

Se dă o matrice pătratică cu n ($4 \leq n \leq 10$) elemente numere întregi. Matricea se citește din fișierul **matrice.in**, de pe prima linie numărul n , iar de pe următoarele n linii, valorile de pe fiecare linie a matricei. În fișierul **matrice.out** se vor afișa următoarele valori:

a) Pe prima linie se va afișa produsul elementelor aflate pe bordura matricei;

b) Pe următoarele $n+2$ linii se va afișa matricea dată, bordată cu o linie la început și una la sfârșit și cu o coloană la început și una la sfârșit având toate elementele egale cu **1**.

EXEMPLU:

matrice.in

```
4
1 1 -2 1
1 -1 2 1
1 1 -3 -1
1 -1 1 2
```

matrice.out

```
-4
1 1 1 1 1 1
1 1 1 -2 1 1
1 1 -1 2 1 1
1 1 1 -3 -1 1
1 1 -1 1 2 1
1 1 1 1 1 1
```

SUBIECTUL 15

Să se scrie un program care citește de la tastatură un șir de cel mult 100 de caractere litere mici și afișează în fișierul **date.out**:

a) Pe prima linie șirul inițial modificat prin eliminarea vocalelor folosind o funcție recursivă.

b) Pe linia a doua a fișierului șirul inițial modificat prin adăugarea la începutul și sfârșitul său a literei „A”.

SUBIECTUL 16

Se citește de la tastatură un vector cu n elemente numere naturale. Să se calculeze maximul dintre elementele vectorului.

a) Se va folosi o funcție recursivă pentru citirea vectorului.

b) Se va folosi o funcție recursivă pentru determinarea elementului maxim.

SUBIECTUL 17

Se citește de la tastatură un număr natural n . Scrieți un program care să folosească o funcție recursivă ce primește ca parametru numărul natural n și afișează:

```
1 2 3 4 ... n
...
1 2 3
1 2
1
```

SUBIECTUL 18

Se citește de la tastatură un vector cu n elemente numere naturale. Scrieți un program care să afișeze elementele vectorului care au proprietatea că sunt numere factoriale.

Se vor folosi funcții recursive pentru prelucrările necesare.

EXEMPLU:

Pentru $n=6$ și vectorul 6 7 24 100 120 46
se vor afișa numerele 6 24 120

SUBIECTUL 19

Se citește de la tastatură două șiruri de maxim 100 de caractere.

a) Să se verifice dacă cele două șiruri sunt anagrame (conțin aceleași caractere în altă ordine) prin afișarea pe ecran a unui mesaj corespunzător, folosind o funcție recursivă care realizează această verificare.

b) Să se ștergă din ambele șiruri toate vocalele și să se afișeze șirurile modificate.

SUBIECTUL 20

Se citește din fișierul text **date.in** de pe prima linie dimensiunea n a unui vector de numere reale, iar de pe linia următoare elementele tabloului. În fișierul **date.out** se vor afișa

a) Pe prima linie elementul maxim din vector folosind metoda Divide & Impera.

b) Pe a doua linie elementul minim din vector folosind metoda Divide & Impera.

SUBIECTUL 21

Se citește un vector de numere reale din fișierul **date.in**. Scrieți un program care afișează în fișierul **date.out** vectorul sortat crescător folosind metoda interclasării.

SUBIECTUL 22

Se citește un număr real $x \geq 1$. Se cere să se scrie un program care calculează $\ln(x)$ cu 3 zecimale exacte folosind o funcție recursivă. Nu se va utiliza funcția logaritmică a limbajului.

SUBIECTUL 23

Se citește din fișierul **date.in** un vector cu n elemente numere naturale. Să se afișeze în fișierul **date.out**:

a) Cmmdc dintre primul și ultimul element al tabloului pe prima linie a fișierului.

b) Cmmdc dintre cele n elemente ale tabloului folosind metoda Divide & Impera, pe linia a doua a fișierului.

SUBIECTUL 24

La inaugurarea unui restaurant sunt prezente mai multe persoane. Clienții își aleg din meniul pus la dispoziție câte o specialitate. Dar deocamdată restaurantul a angajat un singur bucătar care pregătește mâncărurile una după alta, deci clienții nu pot fi serviți decât pe rând. Presupunând că bucătarul se apucă de lucru după ce s-au strâns toate comenzile, stabiliți în ce ordine trebuie să pregătească specialitățile, astfel încât timpul mediu de așteptare al clienților să fie minim.

• **Date de intrare:** Fișierul de intrare **menu.in** conține:

- N ($N \leq 50$) – numărul clienților (pe prima linie);

- N numere reprezentând timpul (exprimat în minute) necesar pregătirii mâncărurilor comandate (pe a doua linie).

- **Date de ieșire:** Fișierul de ieșire **menu.out** va conține:

- Timpul mediu de așteptare (pe prima linie);
- Numerele de ordine ale persoanelor din restaurant în ordinea în care trebuie servite (pe a doua linie).

EXEMPLU:

menu.in	menu.out
5	86.00
30 40 20 25 60	3 4 1 2 5

SUBIECTUL 25

O asociație caritabilă asigură consultații medicale gratuite pentru cei fără posibilități materiale. Există un singur cabinet dotat cu aparatură medicală. Din acest motiv la un moment dat un singur medic poate face consultații. Asociația apelează la n medici de diverse specialități, care își oferă benevol serviciile. Fiecare prezintă un singur interval de-a lungul aceleiași zile, în care este disponibil. Ajutați asociația să realizeze o planificare a consultațiilor în cabinet, astfel încât numărul de medici planificați să fie maxim.

- **Date de intrare:** Fișierul de intrare **medici.in** conține:

- N ($N \leq 1000$) – numărul de medici (pe prima linie);
- N linii conținând câte 2 valori întregi mai mici sau egale cu 24: ora de început și ora de sfârșit a unei consultații.

- **Date de ieșire:** Fișierul de ieșire **medici.out** va conține:

- Numărul de medici planificați (pe prima linie);
- Numerele de ordine ale medicilor planificați (pe a doua linie).

EXEMPLU:

medici.in	medici.out
4	3
2 6 20 4	1 4 3
3 8 23 7	

SUBIECTUL 26

Se consideră un rucsac cu care se poate transporta o greutate maximă G_{max} și mai multe obiecte de greutate g_1, g_2, \dots, g_n , la transportul cărora se obțin câștigurile c_1, c_2, \dots, c_n . Se cere să se încarce rucsacul astfel încât să se obțină un câștig maxim, știind că obiectele pot fi „tăiate”. Rezultatul va fi afișat printr-un șir de n numere reale cuprinse între 0 și 1, având semnificația procentului transportat din fiecare obiect.

- **Date de intrare:** Fișierul de intrare **rucsac.in** conține:

- G_{max} n (pe prima linie);
- Doua linii cu n elemente: prima conține greutățile celor n obiecte iar a doua linie conține câștigurile obținute.

- **Date de ieșire:** Fișierul de ieșire **rucsac.out** va conține:

- O linie cu n valori reale cu semnificația cerută în enunț și pe linia următoare câștigul obținut.

EXEMPLU:

rucsac.in	rucsac.out
3	0 1 0.66
2 1 3	8
2 4 6	

SUBIECTUL 27

Se dau mulțimile $A=\{a_1, a_2, a_3, \dots, a_m\}$ și $B=\{b_1, b_2, b_3, \dots, b_n\}$ având elemente numere întregi nenule. Se știe că m este mai mic sau egal cu n . Se cere să se determine o submulțime a mulțimii B ($\{x_1, x_2, x_3, \dots, x_m\}$) astfel încât valoarea expresiei $E=a_1 \cdot x_1 + a_2 \cdot x_2 + a_3 \cdot x_3 + \dots + a_m \cdot x_m$ să fie minimă și să se calculeze valoarea expresiei E .

- **Date de intrare:** Fișierul de intrare **expresie.in** conține:

- pe prima linie m ;
- pe a doua linie valorile $a_1, a_2, a_3, \dots, a_m$;
- pe a treia linie n ;
- pe a patra linie valorile $b_1, b_2, b_3, \dots, b_n$.

- **Date de ieșire:** Fișierul de ieșire **expresie.out** va conține:

- pe prima linie valorile alese din mulțimea B ;
- pe a doua linie valoarea expresiei E .

EXEMPLU:

expresie.in	expresie.out
3	1 2 4
-3 -1 2	3
4	
1 2 3 4	

SUBIECTUL 28

Pe artera principală a unui oraș sunt amplasate n stații de autobuz. Se dorește selectarea unui număr maxim de stații în care să se facă oprirea unui autobuz cu condiția ca între două stații alăturate să fie cel puțin x metri.

- **Date de intrare:** Fișierul de intrare **statii.in** conține:

- pe prima linie n (numărul stațiilor);
- pe a doua linie $n-1$ numere naturale reprezentând distanțele între stații alăturate;
- pe a treia linie valoarea x .

- **Date de ieșire:** Fișierul de ieșire **statii.out** va conține:

- pe prima linie o valoare reprezentând numărul maxim de stații selectate;
- pe a doua linie un șir de n valori 0 și 1 cu următoarea semnificație: 1 dacă stația i a fost selectată și 0 dacă nu a fost selectată (i ia valori de la 1 la n).

EXEMPLU:

statii.in	statii.out
10	5
100 50 25 25 50 10 10 80 20	1 1 0 1 0 1 0 0 1
60	

SUBIECTUL 29

Într-un zonă turistică se găsesc n obiective turistice. Un turist dorește să plece de la unul dintre obiective, să viziteze toate obiectivele și să revină în punctul de plecare. Cunoscând legăturile directe existente între obiective să se determine toate variantele de traseu pe care le are la dispoziție turistul respectiv.

- **Date de intrare:** Fișierul de intrare **turist.in** conține:
 - pe prima linie n (numărul obiectivelor), x (obiectivul de plecare);
 - pe următoarele n linii, pentru fiecare obiectiv $i \in \{1, 2, 3, \dots, n\}$, obiectivele cu care are legătură directă.
- **Date de ieșire:** Fișierul de intrare **turist.out** va conține:
 - pe fiecare linie numerele de ordine ale obiectivelor turistice în ordinea în care apar în traseu separate prin spațiu.

EXEMPLU:

turist.in	turist.out
6 1	1 2 3 4 5 6 1
2 6	1 2 5 4 3 6 1
1 3 5 6	-----
2 4 5 6	
3 5	
2 3 4 6	
1 2 3 5	

SUBIECTUL 30

Să se genereze toate modalitățile de plată a unei sume S de bani, având la dispoziție n tipuri de monede. Din fiecare tip sunt disponibile oricâte monede.

- **Date de intrare:** Fișierul de intrare **plata.in** conține:
 - pe prima linie n (numărul tipurilor de monede), S (suma de bani);
 - pe următoarea linie n valori naturale reprezentând valorile monedelor.
- **Date de ieșire:** Fișierul de intrare **plata.out** va conține:
 - pe fiecare linie perechi de forma (tipul monedei, numărul monedelor de acel tip).

EXEMPLU:

plata.in	plata.out
3 30	(1, 15)
2 3 5	(2, 10)
	(3, 6)
	(1, 5) (3, 4)

SUBIECTUL 31

Să se genereze toate submulțimile unei mulțimi cu n elemente numere întregi.

- **Date de intrare:** Fișierul de intrare **submult.in** conține:
 - pe prima linie n (numărul elementelor din mulțime);
 - pe următoarea linie n valori întregi reprezentând elementele mulțimii.
- **Date de ieșire:** Fișierul de intrare **submult.out** va conține:
 - pe fiecare linie elementele unei submulțimi separate prin punct și virgulă și încadrate între acolade.

EXEMPLU:

submult.in	submult.out
4	{3}
3 -9 5 -2 1	{3; -9}
	{3; -9; 5}

SUBIECTUL 32

Să se genereze toate modalitățile de descompunere a unui număr natural dat ca sumă de numere naturale nenule.

- **Date de intrare:** Fișierul de intrare **desc.in** conține:
 - pe prima linie n (numărul natural nenul care trebuie descompus).
- **Date de ieșire:** Fișierul de intrare **desc.out** va conține:
 - pe fiecare linie numerele naturale nenule în care este descompus n ; numerele vor fi separate prin spațiu.

EXEMPLU:

desc.in	desc.out
5	1 1 1 1 1
	1 1 1 2
	1 1 2 1

SUBIECTUL 33

O fotografie alb-negru este prezentată sub forma unei matrice binare. Ea înfățișează una sau mai multe obiecte. Porțiunile corespunzătoare obiectelor în matrice au valoarea 1. Se cere să se determine dacă fotografia reprezintă unul sau mai multe obiecte.

- **Date de intrare:** Fișierul de intrare **fotografie.in** conține:
 - pe prima linie m și n (numărul liniilor și coloanelor din matricea care codifică fotografia);
 - pe următoarele m linii câte n valori 0 și 1 (codificarea fotografiei).

- **Date de ieșire:** Fișierul de intrare **fotografie.in** va conține:
 - pe prima linie mesajul „Fotografia conține un sigur obiect” sau mesajul „Fotografia conține mai multe obiecte” urmat de două puncte și de valoarea numerică ce reprezintă numărul obiectelor din fotografie.

EXEMPLU:
fotografie.in

```
4 5
0 1 1 0 1
0 0 0 0 1
1 1 1 0 1
0 1 0 0 1
```

fotografie.out

```
Fotografia conține mai multe obiecte: 3
```

SUBIECTUL 34

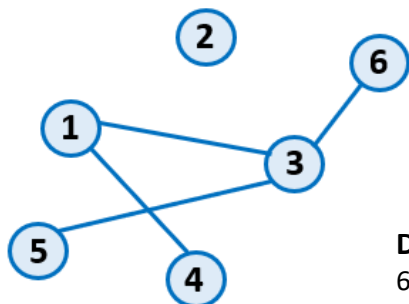
Se dă un graf neorientat cu n vârfuri și m muchii, citit prin vectorul muchiilor, ($0 < n \leq 30$; $0 \leq m \leq 100$).

a) Să se afișeze în fișierul **graf.out**, pe prima linie, numărul total de componente conexe ale grafului citit.

b) Să se afișeze în fișierul **graf.out**, pe următoarele linii, separate, componentele sale conexe.

EXEMPLU:

Pentru $n=6$ vârfuri, $m=4$ muchii și următorul graf neorientat:


 $M = \{[1,3]; [1,4]; [3,5]; [3,6]\}$
Date de intrare:

```
6 4
1 3
1 4
3 5
3 6
```

graf.out

```
2
2
1 3 4 5 6
```

SUBIECTUL 35

Se citesc numărul n de noduri ale unui graf neorientat și adiacentele dintre ele ($n \leq 10$), sub formă de perechi de noduri, din fișierul **graf.in**. Pe prima linie se găsește n , iar pe următoarele, câte o pereche de noduri adiacente.

- Să se afișeze pe ecran nodurile de grad maxim.
- Să se afișeze pe ecran mesajul „Graful este hamiltonian”, dacă este cazul, respectiv „Graful nu este hamiltonian”.

EXEMPLU:

graf.in

6
1 2
1 3
1 4
3 5
3 6

graf.out

1 3
Graful nu este hamiltonian

SUBIECTUL 36

Se dă un graf orientat $G=(X, U)$ având n vârfuri ($n \leq 20$). Se citesc din fișierul text "**graf.in**", de pe prima linie numărul n de vârfuri, iar de pe următoarele n linii, matricea de adiacență asociată grafului.

a) Să se determine și să se afișeze pe ecran nodurile cu proprietatea că au numărul maxim de incidente interioare și în același timp numărul minim de incidente exterioare.

b) Pentru un nod X citit să se afișeze pe ecran cu spații între ele toate nodurile adiacente cu acesta.

EXEMPLU:

graf.in

6
0 0 0 0 1 0
1 0 0 0 0 0
0 0 0 0 0 1
1 0 1 0 0 0
0 1 0 0 0 0
0 0 0 1 1 0

X=5

Date de ieșire (pe ecran):

1 5 (nodurile cu număr maxim de incidente interioare)
4 6 (nodurile cu număr minim de incidente exterioare)
1 2 6 (nodurile adiacente cu nodul X)

SUBIECTUL 37

În fișierul text **graf.in** sunt memorate pe prima linie n ($n \leq 20$) și m ($m \leq 30$) reprezentând numărul de noduri, respectiv numărul de muchii ale unui graf neorientat iar pe următoarele m linii perechi de numere ce reprezintă muchiile grafului.

a) Să se afișeze pe ecran nodul de grad maxim;

b) Să se afișeze în fișierul **graf.out** o parcurgere în adâncime a grafului plecând din nodul de grad maxim.

EXEMPLU:

graf.in

5 4
1 3
1 4
3 2
3 5

graf.out

3 1 4 2 5

Date de ieșire (pe ecran):

3

SUBIECTUL 38

Se definește o muchie a unui graf neorientat ca o înregistrare cu trei câmpuri: cele două vârfuri extremități și un cost asociat muchiei. Să se creeze un graf neorientat ca un vector de m muchii, n vârfuri și un cost asociat muchiei ($2 \leq n$, $m \leq 20$).

- Să se afișeze în fișierul **graf.out** matricea de adiacență a grafului definit.
- Să se afișeze pe ecran muchia (muchii) de cost minim.

EXEMPLU:

Date de intrare:	graf.out
$n=6$ $m=4$	001100
1 3 7 (muchia [1,3] și costul 7)	000000
1 4 12 (muchia [1,4] și costul 12)	100011
3 6 3 (muchia [3,6] și costul 3)	100000
3 5 9 (muchia [3,5] și costul 9)	001000
	001000

Date de ieșire (pe ecran): 3 6 (muchia de cost minim)

SUBIECTUL 39

Fie un graf cu n ($n \leq 10$) vârfuri a cărui matrice de adiacență se citește din fișierul **graf.in**.

- Să se determine dacă graful citit este neorientat sau nu.
- Să se verifice dacă graful dat este graf complet.

EXEMPLU:

graf.in	Date de ieșire:
4	Graf neorientat
0 1 0 1	Nu este graf complet
1 0 1 0	
0 1 0 1	
1 0 1 0	

SUBIECTUL 40

Fie fișierul **graf.in** ce conține pe linii diferite numărul n ($n \leq 10$) de noduri ale unui graf orientat, matricea sa de adiacență și o secvență de noduri.

- Să se verifice dacă secvența de noduri citită din fișier reprezintă un drum în graful orientat.
- Să se verifice ce fel de drum este elementar sau neelementar.

EXEMPLU:

graf.in	Date de ieșire:
6	Secvența citită reprezintă drum elementar
0 0 0 0 1 0	
1 0 0 0 0 0	
0 0 0 0 0 1	
1 0 1 0 0 0	
0 1 0 0 0 0	
0 0 0 1 1 0	
4 3 6 5 2	