Subiectul 1 (3 puncte)

Se dă un graf neorientat cu $n>3$ vârfuri și m muchii și un vârf s.
a) Adăugați la G un număr minim de muchii astfel încât să devină conex. Construiți în memorie și afișați pe ecran listele de adiacență ale grafului astfel obținut. **Complexitate O(n+m)**
b) Determinați excentricitatea ecc(s) a vârfului s în noul graf G_1 obținut la a):

$$ecc(s) = \max(d(s,v)| \text{ v vârf în } G_1\}$$

unde d(s,v) este distanța de la s la v.
Complexitate O(n+m)
Informațiile despre graf se citesc din fișierul ***graf.in*** cu structura:

- pe prima linie sunt n și m
- pe următoarele m linii sunt câte 2 numere naturale reprezentând extremitățile unei muchii
- pe ultima linie este vârful s

graf.in	***Iesire pe ecran (solutia nu este unica)***
6 4 1 3 1 5 3 5 2 4 6	a) 1 2 2 6 b) 3 Explicații: d(6,1)=2, d(6,2)=1, d(6,3)=3, d(6,4)=2, d(6,5)=3, deci ecc(6)=3

Subiectul 2 (3 puncte)

Se citesc informații despre un graf **orientat** ponderat **fără circuite** G din fișierul `graf.in`. Fișierul are următoarea structură:

- pe prima linie sunt două numere reprezentând numărul de vârfuri n (n>4) și numărul de arce m ale grafului, **m>n**
- pe următoarele m linii sunt câte 3 numere pozitive reprezentând extremitatea inițială, extremitatea finală și costul unui arc din graf
- pe următoarea linie (a (m+2)-a linie) din fișier sunt un număr natural k (0<k<n) și un șir de k vârfuri reprezentând vârfurile sursă ale grafului $s_1,\ldots,s_k$
- pe ultima linie a fișierului sunt două vârfuri t_1 și t_2, reprezentând vârfurile destinație ale grafului (distincte de vârfurile sursă de pe linia anterioară).

Notăm cu $S=\{s_1,\ldots,s_k\}$ mulțimea vârfurilor sursă din G și cu $T=\{t_1, t_2\}$ mulțimea vârfurilor destinație din G. Spunem că un vârf y este accesibil din x în G dacă există un drum de la x la y.

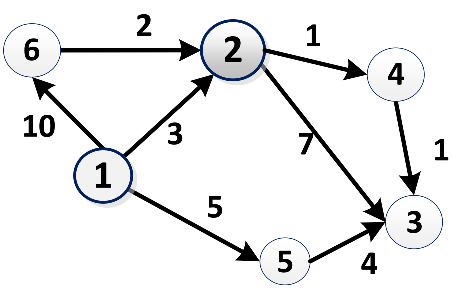
a) Să se verifice dacă graful dat este fără circuite și să se afișeze un mesaj corespunzător.

b) Să se determine un drum de cost maxim care începe cu un vârf din S și se termină cu un vârf din T. Dacă nu există un astfel de drum se va afișa un mesaj corespunzător: niciun vârf destinație nu este accesibil dintr-un vârf sursă.

Complexitate O(m)

Exemplu

graf.in	**Iesire pe ecran**
6 8 1 2 3 1 6 10 6 2 2 2 4 1 4 3 1 5 3 4 1 5 5 2 3 7 2 1 3 2 5	drum maxim 1 6 2



S = {1, 3}
T = {2, 5}

Subiectul 3 (3 puncte)

Propuneți un algoritm **bazat pe algoritmul Ford-Fulkerson / Edmonds Karp** pentru rezolvarea următoarei probleme.

Într-un restaurant sunt n mese numerotate 1,...,n sunt și m ospătari numerotați 1,..., m (m≥n).

Proprietarul restaurantului urmează să aibă un eveniment în restaurant și dorește să repartizeze fiecărui ospătar mesele de care trebuie să se ocupe. El întreabă pe fiecare ospătar la câte mese ar vrea să servească maxim și notează cu $o_1,..,o_m$ răspunsurile acestora.

Proprietarul ar vrea ca la fiecare masa i să **fie exact k_i ospătari** și ar vrea ca numărul de mese la care repartizează un ospătar i **să nu depășească** opțiunea acestuia o_i.

Scrieți un program care, dacă este posibilă o distribuție a ospătarilor la mese care să respecte dorințele proprietarului, să afișeze o astfel de distribuție sub forma prezentată în exemplul de mai jos. Altfel se va afișa mesajul "imposibil"

Datele despre restaurant și opțiunile ospătarilor se vor citi dintr-un fișier cu următoarea structură:

- pe prima linie sunt numerele naturale n, m
- pe următoarea linie n numere naturale $k_1, \ldots, k_n$ reprezentând câți ospătari trebuie să fie la fiecare masă
- pe următoarea linie m numere naturale $o_1 \ldots o_m$ reprezentând opțiunile ospătarilor.

Complexitate $O(n^2m^2)$

restaurant.in	Iesire pe ecran (solutia nu este unica)
3 3 1 2 1 1 2 2	Masa 1: ospătari 1 Masa 2: ospătari 2 3 Masa 3: ospătari 3
restaurant.in	Iesire pe ecran
4 4 1 1 4 4 1 1 4 4	Imposibil