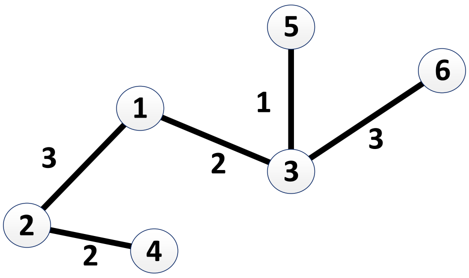
Subiectul 1 (3 puncte)

Se dă un graf neorientat ponderat G cu n>3 vârfuri, m muchii și un vârf s.
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 un vârf sursă s.

Pentru un lanț P în G definim ponderea lanțului P ca fiind produsul ponderilor muchiilor care îl compun.
Dacă G este arbore, să se afișeze pentru fiecare vârf v ponderea unicului lanț elementar de la s la v (sub forma v: pondere lanț de la s la v), altfel să se afișeze un arbore parțial al componentei care conține s. **Complexitate O(n+m)**

graf.in	***Iesire pe ecran***
6 5 1 2 3 1 3 2 2 4 2 3 5 1 3 6 3 1	Este arbore 1: 0 2: 3 3: 2 4: 6 5: 2 6: 6



Subiectul 2 (3 puncte)

Se citesc informaţii despre un graf **orientat** ponderat **fără circuite** G din fişierul `graf.in`. Fişierul are următoarea structură:

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

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

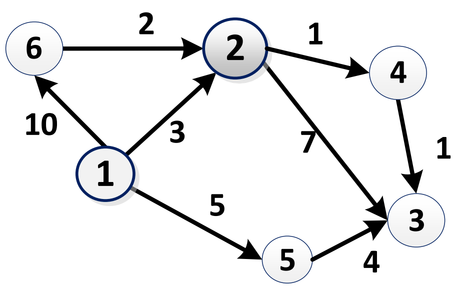
a) Să se verifice dacă graful dat este fără circuite şi să se afişeze un mesaj corespunzător.

b) Să se determine un drum de cost maxim care începe cu un vârf din S şi se termină cu un vârf din T. Dacă nu există un astfel de drum se va afişa un mesaj corespunzător: niciun vârf destinaţie 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.

Pentru n proiecte, numerotate 1,..., n s-au înscris m studenți numerotați 1,..,m, fiecare student depunând o listă de opțiuni cu proiectele la care vrea să participe.

a) Dat un număr k de la tastatură, să de determine o listă de k asocieri proiect – student prin care k studenți diferiți sunt asociați la k proiecte diferite **Complexitate O(km)**

b) Să se determine, dacă există, o modalitatea de a asocia toți studenții la proiecte astfel încât un student să fie asociat la exact 2 proiecte, iar la un proiect să fie asociați exact 2 studenți și să se afișeze o astfel de modalitate sub forma prezentată în exemplul de mai jos. Altfel se va afișa mesajul “nu este posibil”. **Complexitate O(nm)**

Datele despre proiecte și studenți se vor citi dintr-un fișier cu următoarea structură:

- pe prima linie sunt numerele naturale n și m
- pe următoarele linii sunt perechi de numere naturale i j cu $i \in \{1, \dots, n\}$ și $j \in \{1, \dots, m\}$ cu semnificația: studentul j s-a înscris la proiectul i.

graf.in	Iesire pe ecran (solutia nu este unica)
4 4 1 1 1 2 1 3 2 1 2 2 3 1 3 3 3 4 4 3 4 4 (primul este indicele proiectului, al doilea al studentului)	a) pentru k=2 asocieri proiect - student 1 1 2 2 b) asocieri proiect-student 1 1 1 2 2 1 2 2 3 3 3 4 4 3 4 4