Problema 1-Energia 100puncte

Recent, în Oradea s-a deschis un centru de închiriat biciclete și s-au construit noi piste pentru bicicliști. Noile piste de biciclete leagă principalele puncte ale orașului, iar bicicliștii se pot deplasa în ambele direcţii pe aceste piste. Între unele puncte din oraș pista parcurge o suprafață plană, iar între alte puncte piste poate urca sau poate coborî în funcție de direcția de deplasare. Prin urmare, consumul de energie la deplasarea între două puncte nu este același. Astfel, dacă un biciclist se deplasează pe o suprafață plană între două puncte el va consuma **1** unitate de energie, dacă urcă consumul va fi de **2** unități, iar la coborâre consumul va fi evident de 0 unități. De exemplu, dacă între punctele **x** și **y** pista urcă la deplasarea între punctele **x** și **y** biciclistul va consuma **2** unități de energie, dar dacă se deplasează între punctele **y** și **x** consumul de energie va fi **0**, deoarece între aceste puncte pista coboară.

Cerinţe

Cunoscând **n**, numărul de puncte din oraş între care există piste de biciclete, **m** numărul pistelor de biciclete între care suprafața este plană, punctul de start și punctul final al celor **m** piste între care suprafața este plană, **k** numărul de piste care coboară, punctul de start și punctul final al celor **k** piste între există coborâre, scrieți un program care determină:

a) Care este punctul în care se intersectează cele mai multe piste plane;
b) Care este consumul minim de energie între două puncte **x** și **y**.

Date de intrare

Fișierul **`energia.in`** conține pe prima linie un număr natural **p** (**1 ≤ p ≤ 2**), semnificând varianta cerinței de rezolvare. Pe a doua linie se găsesc trei numere naturale despărțite printr-un spațiu **n**, **m** și **k**, cu semnificația din enunț. Pe următoarele **m** linii se găsesc câte două numere naturale pe fiecare linie, separate printr-un spațiu, reprezentând punctul de start și punctul final al pistelor plane. Pe următoarele **k** linii se găsesc câte două numere naturale pe fiecare linie, separate printr-un spațiu, reprezentând punctul de start și punctul final al pistelor care coboară. Pe ultima linie se află două numere naturale **x** și **y**.

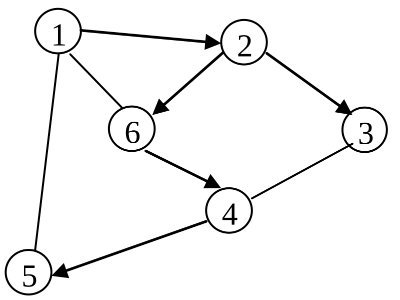
Date de ieşire

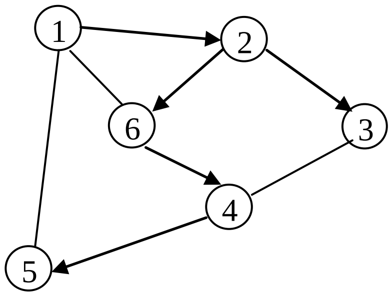
Dacă **`p = 1`** se va rezolva **numai punctul a) al cerinței**. În acest caz, fișierul **`energia.out`** conține o valoare naturală reprezentând numărul de ordine al punctului în care se intersectează cele mai multe piste plane. Dacă există mai multe soluții se va afișa cea mai mică.
Dacă **`p = 2`** se va rezolva **numai punctul b) al cerinței**. În acest caz, fișierul **`energia.out`** conține o valoare naturală reprezentând consumul minim de energie între punctele **x** și **y**.

Restricţii şi precizări

- punctele din oraș între care există piste sunt numere naturale numerotate de la 1 la **n**;
- 1 ≤ **n** ≤ 200, 1 ≤ **m** + **k** ≤ 19900;
- dacă între două puncte **a** și **b** pe traseul de la **a** la **b** pista urcă, pe traseul de la **b** la **a** pista coboară;
- între oricare două puncte legate direct printr-o pistă traseul poate fi numai plat, în urcare sau în coborâre.
- oricare două puncte din oraș sunt legate prin piste (direct sau indirect).
- `30%` din teste vor avea pe prima linie valoarea `1`, iar restul de `70%` din teste vor avea pe prima linie valoarea `2`.

Exemplu

energia.in	**energia.out**	**Explicație**
1 6 3 5 1 5 1 6 3 4 1 2 2 6 2 3 4 5 6 4 1 5	1	 Se va rezolva **numai punctul a)** În punctul cu numărul de ordine 1 se intersectează cele mai multe piste plane.

2 6 3 5 1 5 1 6 3 4 1 2 2 6 2 3 4 5 6 4 1 5	0		Se va rezolva numai punctul b) Pe traseul 1→2→6→4→5consumul de energie este minim și are valoarea 0.

Timp maxim de execuţie/test: 1s

Memorie totală: 8KB

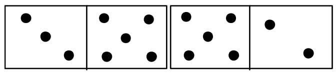
Dimensiune maximă a sursei: 4 KB.

Problema 2 – Domino 100p

Se consideră **n** dominouri, din care se pot construi şiruri respectând următoarele reguli:

- Prima piesă de domino face parte în mod obligatoriu din şir.
- Următoarele piese se iau în considerare în ordinea în care sunt date (se decide pentru fiecare dacă se va pune în şir sau nu).
- Fiecare piesă se pune la un capăt al şirului deja format în poziţia în care a fost dată sau învârtită cu 180°, sau se pune la o parte şi nu se mai revine la ea.
- O piesă poate fi pusă la un capăt al şirului, dacă numărul de pe dominoul din capăt (mai precis de pe jumătatea nealipită de şirul construit deja) şi numărul de pe dominoul care se alipeşte şirului la pasul curent (partea care se alipeşte) sunt egale.

Exemplu



Cerinţă

Să se determine cel mai lung şir de dominouri care se poate construi din dominourile date.

Date de intrare

Pe prima linie a fişierului **`DOMINO.IN`** se află un numărnatural *n*, reprezentând numărul de dominouri. Pe fiecare din următoarele **n** linii se află două numere ***x*** şi ***y***, care reprezintă numerele de pe piesa de domino respectivă.

Date de ieşire

În fişierul **`DOMINO.OUT`** se va scrie un singur număr natural, reprezentând lungimea celui mai lung şir de piese care se poate construi respectând regulile de mai sus.

Restricţii şi precizări

- 1 ≤**n**≤ 100000;
- 0 ≤**x, y**≤ 9;
- piesele pot fi rotite înainte să fie aşezate la un capăt al şirului.

Exemplu

DOMINO.IN	DOMINO.OUT
6 1 2 1 6 2 3 1 4 2 3 4 3	5

Timp maxim de execuţie/test: 1s
Memorie totală:10KB
Dimensiune maximă a sursei:5 KB.