



A 2013/2014 tanévi Országos Középiskolai Tanulmányi Verseny második fordulójának feladatai

II. (programozás) kategória

1. feladat: Nagy Fal (38 pont)

A Kínai Nagy Falon N őrhelyet létesítettek. Közülük azonban csak M helyen van őrség. Két szomszédos őrhely közötti fal őrzött, ha legalább az egyik végén van őrség; védett, ha mindkét végén van őrség. Őrzött szakasznak nevezzük egymást követő őrzött falak nem bővíthető sorozatát. Hasonlóan, védett szakasznak nevezzük egymást követő védett falak nem bővíthető sorozatát. Ha egy helyen több őrség is van, akkor a feleslegesek elküldhetők más őrhelyre (egynek azonban ott kell maradnia).

Készíts programot (`nagyfal.pas`, ...), amely megadja a védett és az őrzött szakaszok számát, valamint azt, hogy a felesleges őrségek jó helyre küldésével hány további fal tehető védetté, illetve őrzötté! (Az utóbbi két feladathoz lehetséges, hogy a felesleges őrségeket más-más helyre kellene küldeni.)

A `nagyfal.be` szöveges állomány első sorában az őrhelyek száma ($1 \leq N \leq 100$) és az őrségek száma ($1 \leq M \leq 100$) van, egy szóközzel elválasztva. A következő M sor az őrségek leírását tartalmazza, közülük az i -edik annak az őrhelynek a sorszáma, ahol az i -edik őrség van.

A `nagyfal.ki` szöveges állományba négy sort kell írni! Az első sorba a védett szakaszok száma, a másodikba az őrzött szakaszok száma kerüljön! A harmadik sorba a védetté, a negyedik sorba az őrzötté tehető újabb falak számát kell írni!

Példa:

`nagyfal.be`

12 9

6

3

12

11

4

5

8

6

6

`nagyfal.ki`

2 [a 3-6 és a 11-12 között]

2 [a 2-9 és 10-12 közötti]

3 [pl. a 6-7,7-8 és 2-3 közötti]

2 [az 1-2 és a 9-10 közötti]



2. feladat: Város (26 pont)

Egy modern nagyváros úthálózata egy négyzetrácsal írható le, ahol N jelöli a négyzetrács sorainak számát (azaz a kelet-nyugati irányú utak számát, az ilyen utakat alulról felfelé sorszámozzuk), M pedig az oszlopokét (azaz az észak-déli utakét, az ilyen utakat balról jobbra sorszámozzuk). El szeretnénk jutni a város egyik kereszteződéséből egy másik kereszteződésbe. Az egyes kereszteződések (csomópontok) előtt a haladási irányt befolyásoló jelzőtáblák lehetnek, melyeket a következő kódokkal látunk el:

balra fordulni tilos	BT
jobbra fordulni tilos	JT
balra haladni kötelező	BK
jobbra haladni kötelező	JK

Visszafordulni semelyik csomópontban sem lehet, azaz ha nincs jelzőtábla, akkor is csak maximum háromfelé haladhatunk. Utközben a várost nem hagyhatjuk el (bár erről szóló jelzőtáblák nincsenek). Az indulási helyről bámerre indulhatunk, azt nem befolyásolja tábla.

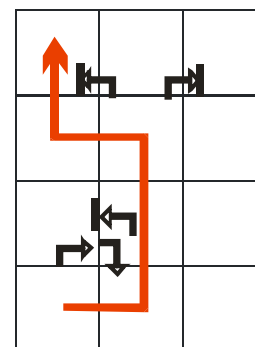
Írj programot (`varos.pas`, ...), amely a táblák figyelembevételével megadja a legrövidebb útvonalat, amelyeken áthaladva eljuthatunk az indulási helyről a célba!

A `varos.be` állomány első sorában a sorok és oszlopok száma ($1 \leq N, M \leq 100$), valamint a táblák száma ($1 \leq T \leq 10\,000$) van egy-egy szóközzel elválasztva. A következő T sorban soronként egy-egy tábla leírása található. A táblaleírás formája: `irány sor oszlop kód`, ahol a sor és az oszlop a csomópont koordinátáit adja meg ($1 \leq \text{sor} \leq N$, $1 \leq \text{oszlop} \leq M$), az irány pedig a csomópontba beérkező útszakasz irányát (E, K, D, N), azaz a beérkező út északi, keleti, déli vagy nyugati irányban halad-e. Az utolsó sorban a két kereszteződés sor és oszlopindexe van, egy-egy szóközzel elválasztva, az első az induló hely, a második a cél.

A `varos.ki` állomány első sorába a két pont közötti legrövidebb út hosszát kell írni! A második sorba a legrövidebb út leírását kell írni, ahol minden lépést a haladás iránya, azaz az E, K, D vagy N betű azonosít. Feltehető, hogy ilyen út mindig van!

Példa:

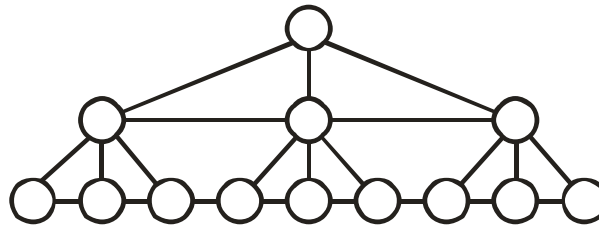
<code>varos.be</code>	<code>varos.ki</code>
4 3 5	5
E 2 1 JK	KEENE
K 2 2 JK	
E 2 2 BT	
E 4 2 BT	
E 4 2 JT	
1 1 4 1	



E 2 1 JK magyarázata: Ha északi irányba haladva a (2,1) pontba érünk, akkor ott kötelező jobbra fordulni, azaz innen csak keleti irányba a (2,2) pontba mehetünk.

3. feladat: Játéktábla (26 pont)

Egy játéktábla 101 sorból áll, minden sorában pontosan háromszor annyi elem van, mint a fölötte levő sorban. A tábla a következő szerkezetű:



A tábla felső pontjából indulunk. Az egyes lépéseket a következők írják le:

- 0 balra lefelé lépünk egyet,
- 1 középen lefelé lépünk egyet,
- 2 jobbra lefelé lépünk egyet,
- 3 felfelé lépünk egyet,
- 4 balra lépünk egyet,
- 5 jobbra lépünk egyet.

Készíts programot (`lepes.pas, ...`), amely beolvas egy lépéssorozatot, amely elvezet a tábla valamely eleméhez, majd megad egy olyan lépéssorozatot, amely a legrövidebb úton vezet ugyanide!

A `lepes.be` állomány első sorában a lépések K száma van ($1 \leq K \leq 100$), a következő sorban pedig az egyes lépéseket leíró K darab szám, egy-egy szóközzel elválasztva. *A lépéssorozat biztosan helyes, azaz nem hagyjuk el vele a játéktáblát.*

A `lepes.ki` állomány első sorába a legrövidebb lépéssorozat L hosszát kell írni, amely a bemenetben kapott lépéssorozattal azonos helyre vezet! A második sorba pedig egy ilyen legrövidebb lépéssorozat kerüljön, azaz L szám, egy-egy szóközzel elválasztva!

Példa:

<code>lepes.be</code>	<code>lepes.ki</code>
6	2
0 2 5 3 1 5	1 2

4. feladat: Gépek (25 pont)

Egy vállalkozó a következő N napra megrendeléseket fogad. Minden munkát egy nap alatt tud elvégezni. M megrendelés érkezett, minden megrendelésben szerepel, hogy az igényelt munkát milyen határidőig kell elvégezni. A vállalkozónak ki kell számítani, hogy legkevesebb hány gépre van szükség, hogy minden igényelt munkát határidőre el tudjon végezni.

Készíts programot (`gepek.pas, ...`), amely kiszámítja, hogy legkevesebb hány gép kell ahhoz, hogy minden megrendelt munkát határidőre el tudjon végezni a vállalkozó! A program adja is meg, hogy ez egyes megrendelést melyik napon, melyik gépen végezze el a vállalkozó!

A `gepek.be` szöveges állomány első sora két egész számot tartalmaz, a munkanapok N számát ($1 \leq N \leq 10000$), és a megrendelések M számát ($1 \leq M \leq 100000$). A második sor pontosan M egész számot tartalmaz egy-egy szóközzel elválasztva. Az i -edik szám az i -edik megrendelés határideje. Minden h határidőre teljesül, hogy $1 \leq h \leq N$.

A `gepek.ki` szöveges állományba $M+1$ sort kell írni! Az első sor a minimálisan szükséges gépek G számát tartalmazza! A további M sor tartalmazza az igényelt munkák beosztását az igények sorszáma szerinti sorrendben! Minden sor két számot tartalmazzon, egy szóközzel elválasztva, az első szám annak a napnak a sorszáma legyen, amelyik napon a munkát elvégzik, a második szám pedig annak a gépnek a sorszáma legyen, amelyiken a munkát elvégzik! Több megoldás esetén bármelyik megadható.

Példa:

gepek.be	gepek.ki
10 8	2
3 2 3 2 4 5 6 2	2 2
	1 1
	3 1
	2 1
	3 2
	4 1
	4 2
	1 2

5. feladat: Fazekas (35 pont)

Korondi János fazekas mester két kemencében égeti ki a termékeit. A kiégetésre váró tárgyak az elkészülés sorrendjében váraakoznak a kiégetésre. Minden tárgyra rá van írva, hogy legkevesebb hány percig kell égetni a kemencében. Mindkét kemencébe egyidejűleg legfeljebb K darab tárgy rakható kiégetésre. Minden tárgyat az elkészülés sorrendjében betesz valamelyik kemencébe, mindkettőbe legalább egy tárgyat kell rakni egy menetben és a következő menet csak akkor kezdődik, ha mindkét kemence befejezte az égetést. Ha valamelyik kemencében több tárgyat éget egy menetben, akkor az égetési idő az abba a kemencébe tett tárgyak egyedi égetési idejének a maximuma. Mivel a kemencék energia fogyasztása az égetési idő függvénye, ezért a fazekasnak arra kell törekednie, hogy az összesített égetési idő a lehető legkevesebb legyen.

Írj programot (`fazekas.pas,...`), amely kiszámítja, hogy legkevesebb mennyi összesített égetési idő alatt lehet kiégetni az összes tárgyat!

A `fazekas.be` szöveges állomány első sora két egész számot tartalmaz, a kiégetésre váró tárgyak N számát ($2 \leq N \leq 1000$) és a két kemence egyedi K kapacitását ($1 \leq K \leq 20$), azaz egyidejűleg legfeljebb K tárgy rakható mindkét kemencébe. A második sor pontosan N egész számot tartalmaz (egy-egy szóközzel elválasztva), az i -edik szám az i -edik tárgy minimális égetési ideje. Minden égetési idő legfeljebb 20000 .

A `fazekas.ki` szöveges állomány első egy egész számot tartalmazzon, a legkevesebb összesített égetési időt! A következő N sor mindegyike két egész számot tartalmazzon egy szóközzel elválasztva! Az i -edik sorban az első szám annak a menetnek a sorszáma legyen, amelyikben az i -edik tárgyat kemencébe rakjuk, a második szám pedig 1, ha az első, 2, ha a második kemencébe rakjuk égetni! A kiírást a menetek sorrendjében kell megadni! Több megoldás esetén bármelyik megadható.

Példa:

fazekas.be	fazekas.ki
8 2	22
1 7 4 9 2 9 1 2	1 1
	1 2
	1 2
	2 1
	2 2
	2 1
	3 1
	3 2

Elérhető összpontszám: 150 pont + 50 pont az 1. fordulóból