

Kedves Versenyző! A megoldások értékelésénél csak a **programok futási eredményeit** vesszük tekintetbe. Ezért igen fontos a **specifikáció pontos betartása**. A programokat csak a feladatkiírásban leírt szabályoknak megfelelő adatokkal próbáljuk ki, emiatt nem kell ellenőrizni, hogy a bemenő adatok helyesek-e, illetve a szükséges állományok léteznek-e. Ha a programnak valamilyen állományra van szüksége, akkor azt mindig az aktuális könyvtárba kell rakni. Az állomány neve minden esetben rögzített.

1. feladat: Strázsa

Egy iskola több napig tartó rendezvényén a tanulók folyamatos ügyeletet tartanak. A rendezvény kezdete az 1 ., vége az N -edik időegység. Minden tanuló megad egy ügyeleti idő-intervallumot (egy $a\ b$ számpárt), amikor vállalná az ügyeletet. Az ügyeleti beosztást úgy kell elkészíteni, hogy az egymást váltó tanulók át tudják adni egymásnak a szolgáltatást, azaz ha az A tanuló ($a1\ a2$ intervallum) a B tanuló ($b1\ b2$ intervallum) követi a szolgálatban, akkor $b1 \leq a2$ teljesüljön.

Írj programot (**strazsa.cbp (main.cpp)**), amely kiszámítja, hogy minimálisan hány tanulóval lehet a kapuügyeletet megoldani a rendezvény teljes idejére a fent leírt feltételnek megfelelően!

A **strazsa.be** állomány első sora a rendezvény időtartamát (N időegység) tartalmazza ($1 \leq N \leq 10000000$). Az állomány ezt követő minden sorában egy $a\ b$ számpár van, egy szóközzel elválasztva ($1 \leq a \leq b \leq N$): egy tanuló által vállalt ügyeleti idő-intervallum. A bemeneti állomány utolsó sora a $0\ 0$ számpárt tartalmazza.

A **strazsa.ki** állományba egyetlen számot kell írni: a lehető legkevesebb tanulót igénylő beosztásban alkalmazott tanulók számát. Ha a feladatnak nincs megoldása (azaz nem lehet a rendezvény teljes időtartamára a jelentkezésekből megfelelő beosztást kialakítani), akkor az állományba 0 -t kell írni.

Példa:

strazsa.be	strazsa.ki
10	3
1 4	
2 3	
3 7	
2 5	
4 9	
5 10	
0 0	

2. feladat: Mérő

Van két kannánk, az első űrtartalma A liter, a másodiké B liter. Ki kell mérnünk N liter vizet a második kannába. Kezdetben mindkét kanna üres. A kannákkal a következő műveleteket végezhetjük:

- TA az első kannát teletöltjük a csapról,
- TB a második kannát teletöltjük a csapról,
- UA az első kannát kiürítjük,
- UB az második kannát kiürítjük,
- AB az első kannából áttöltjük a benne lévő vizet a második kannába úgy, hogy ha mind belefér, akkor mind áttöltjük, egyébként annyit töltünk át, hogy a második kanna tele legyen,
- BA a második kannából áttöltjük a benne lévő vizet az első kannába úgy, hogy ha mind belefér, akkor mind áttöltjük, egyébként annyit töltünk át, hogy az első kanna tele legyen,

Írj programot (**mero.cbp (main.cpp)**), amely a két kanna űrtartalma és az előállítandó N mennyiség ismeretében meghatároz egy olyan műveletsort, amelyet sorban végrehajtva a második kannában N liter víz lesz!

A **mero.be** állomány első sorában az előállítandó N vízmennyiség ($1 \leq N \leq 100$), második sorában pedig az első és a második kanna űrtartalma, egy A B egész számpár van, egyetlen szóközzel elválasztva ($1 \leq A, B \leq 100$).

A **mero.ki** állományba a megoldás műveletsorát (soronként egy-egy műveletet) kell írni. Ha nincs megoldás, akkor az állomány első és egyetlen sora a NINCS szöveg legyen.

Példa:

mero.be	mero.ki
2	TA
3 7	AB
	TA
	AB
	TA
	AB
	UB
	AB

3. feladat: Tájékozódási verseny

A tájékozódási versenyen egy adott helyről kiindulva és ugyanide megérkezve megadott idő alatt minél több állomást fel kell keresni. Az állomásokat utak kötik össze, és ismerjük, hogy melyik hány perc alatt járható be.

Írj programot (**tajek.cbp (main.cpp)**), amely megadja azt a rajttól a célig vezető útvonalat, amelyik az út teljesítésére szánt időtartam alatt a lehető legtöbb különböző állomást érint! Ha több ilyen is van, akkor közülük azt kell megadni, ami a legkevesebb idő alatt teljesíthető.

A **tajek.be** állomány első sorában az állomások száma ($1 \leq N \leq 100$), a másodikban a rajt/cél állomás sorszáma, a harmadikban a verseny teljes időtartama ($0 \leq T \leq 1000$) van. Ezt követően soronként egy-egy út adatai vannak: a két végén található állomás sorszáma és a kalkulált idő (pozitív egész szám) egy-egy szóközzel elválasztva.

A **tajek.ki** állományba a leghosszabb megtehető útvonalat kell írni: minden egyes sora egy állomás sorszámát és elérésének idejét tartalmazza, az elérés ideje szerint növekvő sorrendben.

Példa:

tajek.be	tajek.ki
5	1 0
1	2 20
100	3 40
1 2 20	2 60
1 4 30	5 80
1 5 20	1 100
2 3 20	
2 4 40	
2 5 20	
3 4 40	
4 5 30	

4. feladat: Sziget

Egy, a tengeren levő sziget sorsát nagy mértékben befolyásolja a tengerszint változása. A tengerszint emelkedése esetén a sziget területe csökken, a tengerszint csökkené-

se esetén pedig növekszik. A vízszint emelkedésekor az alacsonyan fekvő területeket elönti a tenger, csökkenésekor pedig a magasabban fekvő területek szárazra kerülhetnek, ha a víz le tud folyni róluk. (A víz mozgásakor minden négyzet 4 oldalszomszédját kell figyelembe venni.)

A szigetet tartalmazó területet kis négyzetekre osztjuk, megadva minden négyzeten a talaj eredeti tengerszint feletti magasságát (ez a szám negatív ott, ahol jelenleg tenger van – ilyenkor a tenger mélységét jelenti). Feltehetjük, hogy a szélső négyzetek a legnagyobb tengerszint csökkenés esetén is vízzel fedettek maradnak.

Készíts programot (**sziget.cbp (main.cpp)**), amely a terület magasságadatai és a tengerszint változása alapján kiszámítja a sziget területét (a vízzel nem fedett négyzetek számát)!

A **sziget.be** állomány első sorában a terület mérete (N sor, M oszlop, $1 \leq N \leq 100$, $1 \leq M \leq 100$) és a tengerszint változások száma ($1 \leq K \leq 100$) van, egy szóközzel elválasztva. A következő N sor mindegyikében M egész szám van, az egyes területek eredeti tengerszint feletti magassága, szintén egy-egy szóközzel elválasztva. Az utolsó K sor egy-egy pozitív vagy negatív egész számot tartalmaz, amely a tengerszint növekedését vagy csökkenését írja le.

A **sziget.ki** állomány $K+1$ sorába egy-egy egész számot kell írni. Az első sor a sziget területét (azaz a vízzel nem fedett négyzetek számát) tartalmazza a kiinduló állapotban. A további K sorban az egyes tengerszintváltozások utáni sziget terület értéke legyen!

Példa:

```
sziget.be
8 10 3
```

```
-5 -5 -5 -5 -5 -5 -5 -5 -5 -5
-5 -5 +1 +2 +1 +1 +1 +1 -5 -5
-5 +1 +1 +9 +9 +9 +1 +1 +1 -5
-5 +1 +1 +9 +4 +4 +9 +1 +1 -5
-5 +1 +1 +9 +4 +4 +9 +1 +1 -5
-5 -5 +1 +4 +9 +9 +4 +4 +1 -5
-5 +0 +0 +1 +1 +1 +1 +1 +1 -5
-5 -5 -5 -5 -5 -5 -5 -5 -5 -5
```

Egy lapos partú sziget, amelyen egy magas, gyűrű alakú hegyerinc zár körbe egy közepes magasságú völgyet. (A +0 magasságú pont kezdetben még szárazföld.)

```
5
```

```
8
```

```
-5
```

A tengerszint háromszor változik, kétszer nő, majd egyszer csökken.

```
sziget.ki
```

```
45
```

```
13
```

```
0
```

```
9
```

Magyarázat

5 méter növekedés elönti a partot, de a hegyekkel körülrárt terület szárazon marad.

Újabb 8 méter növekedés után a teljes sziget elmerül.

5 méter csökkenés után a hegyek már kilátszanak, de a belső völgyből nem tud lefolyni a víz.