

Zadaci

Zadatak	SPOJ	MO	PIZZA	LIFTOVI
Izvršna datoteka	spoj.exe	mo.exe	pizza.exe	liftovi.exe
Izvorni kôd	spoj.pas spoj.c spoj.cpp	mo.pas mo.c mo.cpp	pizza.pas pizza.c pizza.cpp	liftovi.pas liftovi.c liftovi.cpp
Ulazna datoteka	spoj.in	mo.in	pizza.in	liftovi.in
Izlazna datoteka	spoj.out	mo.out	pizza.out	liftovi.out
Vremensko ograničenje (po test podatku)	5 sekundi	5 sekundi	5 sekundi	5 sekundi
Broj test podataka	10	10	10	10
Broj bodova (po test podatku)	3	4	6	7
Ukupno bodova	30	40	60	70
	200			

SPOJ

Ivica je zabrinut za količinu džeparca koju troši na surfanje internetom.

Jedna minuta surfanja u vremenu **između 7h ujutro i 19h navečer** naplaćuje se **10** lipa, a u vremenu **između 19h navečer i 7h ujutro** **5** lipa (cijena je **ista** za radne i neradne dane te praznike).

Početak i kraj internet veze uvijek padaju na **cijelu** minutu, a jedna veza traje **maksimalno 60 minuta**.

Napišite program koji će, na temelju popisa spajanja na internet, izračunati **ukupnu** količinu novca potrošenog na surfanje.

Ulazni podaci

U prvom retku ulazne datoteke nalazi se cijeli broj N , broj spajanja na internet, $1 \leq N \leq 100$.

U svakom od sljedećih N redaka se nalazi podatak o jednom spajanju, u formatu:

$SS:MM$ TT

pri čemu $SS:MM$ označava sat i minutu početka veze, a TT označava trajanje veze (u minutama, maksimalno 60). Između MM i TT nalazi se jedan razmak.

Ako je sat ili minuta početka veze ili trajanje veze jednoznamenasti broj onda se **ispred tog broja nalazi znamenka nula**.

Vremena su označena od 00:00 do 23:59.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženi iznos (u lipama).

Test primjeri

spoj.in

2
11:02 11
15:30 01

spoj.out

120

spoj.in

3
20:05 12
06:45 30
13:08 15

spoj.out

435

spoj.in

5
00:00 05
06:47 35
11:30 18
18:33 60
23:59 22

spoj.out

1035

MO

Mirko i Slavko igraju MO tj. mini go. MO je igra slična drevnoj kineskoj igri GO, ali je jednostavnija i igra se na (**jednodimenzionalnoj**) ploči koja se sastoji od određenog broja polja poredanih jedno do drugoga.

Mirko ima **bijele** žetone i igra **prvi**, a Slavko ima **crne** žetone i igra **drugi**.

Na početku igre ploča je **prazna**. Igrači igraju **naizmjenično** tako da u svakom potezu igre stavljaju po **jedan** žeton na neko od **slobodnih** polja na ploči.

Ako žeton koji je neki igrač **upravo stavio opkoljava** (s nekim žetonom iste boje koji se već od ranije nalazi na ploči) protivnički žeton ili skupinu žetona (uzastopne žetone iste boje), onda se u tom trenutku prije sljedećeg poteza ti protivnički žetoni **uklanjaju** s ploče.

Polja na ploči označena su redom brojevima od 1 do P, slijeva na desno.

Napišite program koji će izračunati broj **bijelih** i broj **crnih** žetona **na ploči na kraju igre**.

Ulazni podaci

U prvom retku nalaze se dva cijela broja P i N, međusobno odvojena jednim razmakom, broj polja na ploči i ukupni broj poteza oba igrača, $1 \leq P \leq 100$, $1 \leq N \leq 1000$.

U svakom od sljedećih N redaka se nalaze potezi Mirka i Slavka **redom igranja**.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati broj bijelih i broj crnih žetona na ploči na kraju igre. Te brojeve međusobno odvojite jednim razmakom.

Test primjeri

mo.in	mo.in	mo.in
4 4	5 6	6 8
2	1	1
3	4	2
4	5	5
3	2	3
	3	4
mo.out	2	6
		2
2 1	mo.out	3
	3 1	mo.out
		2 2

PIZZA

Naš prijatelj Picko naslijedio je velike novce i u rodnom gradu želi otvoriti lanac restorana s dostavom. Glavno, najfinije i najjeftinije jelo bi, naravno, bila **pizza**.

Na raspolaganju mu je određeni broj pogodnih lokacija za otvaranje restorana, a poznate su mu i lokacije nebodera čiji bi stanovnici sigurno bili česti konzumenti njegovih proizvoda.

Dostava svakog restorana će pokrivati sve nebodere u određenom radijusu. Picko može otvoriti samo ograničeni broj restorana, a on želi da to bude na lokacijama kojima bi obuhvatio maksimalni broj stanovnika u neboderima.

Napišite program koji će izračunati **maksimalni** broj stanovnika koje možemo obuhvatiti dostavom.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja K i R , međusobno odvojena jednim razmakom, broj restorana u lancu i radijus dostave svakog restorana, $1 \leq K \leq 10$, $1 \leq R \leq 500$.

U drugom retku se nalazi cijeli broj M , broj pogodnih lokacija za restorane, $K \leq M \leq 20$.

U svakom od sljedećih M redaka nalaze se po dva cijela broja X i Y , međusobno odvojena jednim razmakom, koordinate svake pojedine lokacije, $-1000 \leq X, Y \leq 1000$.

U sljedećem retku nalazi se cijeli broj N , broj nebodera, $1 \leq N \leq 100$.

U svakom od sljedećih N redaka nalaze se po tri cijela broja X , Y i S , međusobno odvojena s po jednim razmakom, X i Y su koordinate svakog pojedinog nebodera, a S broj stanovnika u tom neboderu, $-1000 \leq X, Y \leq 1000$, $1 \leq S \leq 100$. Smatramo da je neboder u radijusu nekog restorana ako je udaljenost između restorana i nebodera **manja ili jednaka** od R .

Sve lokacije za restorane i svi neboderi nalaze se na **međusobno različitim** koordinatama.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženi maksimalni broj stanovnika koje Picko može obuhvatiti dostavom.

Test primjeri

pizza.in

```
2 2
3
1 0
4 0
7 0
4
0 0 1
3 0 7
5 0 9
8 0 1
```

pizza.out

18

pizza.in

```
2 2
3
-2 0
0 1
3 0
8
-3 1 1
-3 0 1
-3 -1 1
-2 -1 1
0 0 3
0 2 1
2 1 3
4 0 2
```

pizza.out

12

pizza.in

```
3 3
5
0 0
1 6
2 3
6 6
7 2
8
0 1 2
0 5 3
0 6 1
1 0 1
3 2 3
3 6 2
6 2 4
8 6 3
```

pizza.out

17

LIFTOVI

Neboder ima N liftova. Svaki lift vozi između **točno dva kata** i **ne staje na međukatovima**. Svi su **jednako brzi** i treba im **5 sekundi** da pređu **jedan kat**.

Na početku su svi liftovi u svojoj **donjoj** poziciji i kreću prema gore. Nakon što dođu do svojih viših katova kreću prema dolje, pa prema gore, itd., non-stop **bez zadržavanja** na katovima između kojih voze.

Mirko se nalazi na **prvom (najnižem)** katu i želi **što brže** doći na **zadnji (najviši)** kat nebodera. Prelaziti iz lifta u kojem se vozi u drugi lift smije **samo** na katovima u kojima oba lifta staju i ako je drugi lift **u tom trenutku na tom katu** prelazak mu **ne oduzima** ni malo vremena.

Napišite program koji će izračunati **minimalno** vrijeme u kojem Mirko može doći na željeni kat.

Ulazni podaci

U prvom retku ulazne datoteke nalaze se dva cijela broja K i N , međusobno odvojena jednim razmakom, broj katova u neboderu i broj liftova, $2 \leq K \leq 1000$, $1 \leq N \leq 50000$ (pedeset tisuća).

U svakom od sljedećih N redaka nalazi se podatak za jedan lift tj. dva cijela broja A i B , međusobno odvojena jednim razmakom, $1 \leq A < B \leq K$, koja nam kažu da taj lift putuje između katova A i B .

Ne postoje dva lifta koja putuju između **istih** katova.

Napomena: ulazni podaci će biti takvi da će rješenje uvijek postojati.

Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženo minimalno vrijeme (u sekundama) iz teksta zadatka.

Test primjeri

liftovi.in	liftovi.in	liftovi.in
10 4	10 3	20 5
1 5	1 5	1 7
5 10	3 5	7 20
5 7	3 10	4 7
7 10		4 10
liftovi.out	liftovi.out	10 20
45	105	liftovi.out
		150