

ZADATAK	VLAKIĆ	ULICA	NAJBOLJIH 5	OKRET	ZADAĆA	KOMPIĆI	FUNKCIJA	RASPORED
izvorni kôd	vlakic.pas vlakic.c vlakic.cpp	ulica.pas ulica.c ulica.cpp	najboljih5.pas najboljih5.c najboljih5.cpp	okret.pas okret.c okret.cpp	zadaca.pas zadaca.c zadaca.cpp	kompici.pas kompici.c kompici.cpp	funkcija.pas funkcija.c funkcija.cpp	raspored.pas raspored.c raspored.cpp
ulazni podaci	standardni ulaz							
izlazni podaci	standardni izlaz							
vremensko ograničenje	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda	2 sekunde
memorijsko ograničenje	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB	128 MB	128 MB
broj bodova	20	30	50	80	100	120	150	150
	ukupno 700, maksimalno 600 (natjecatelju se zbrajaju bodovi onih 5 zadataka na kojima je ostvario najviše bodova)							

Vrtićka grupa od N djece treba otići u zabavni park. Za grupu se brinu dvije tete, koje su odlučile da cijelu grupu od N djece podijele u dvije skupine s jednakim brojem djece i da svaka teta povede jednu od njih.

Ako je ukupan broj djece neparan, onda će ih se podijeliti na “skoro jednake“ skupine, tj. jedna će teta povesti jedno dijete više nego druga teta.

Nakon što se djeca podijele u skupine, svaka skupina krenut će svojim putem u obilazak parka. Glavna atrakcija zabavnog parka je popularni **vlakić**. U vlakić stane M djece. Izračunajte najmanji ukupan broj vožnji koji je potreban da bi se sva djeca iz obje skupine provozala u vlakiću. (Djeca iz različitih skupina ne mogu se voziti zajedno jer posjećuju vlakić u različito vrijeme.)

Na primjer, ako je N (ukupan broj djece) jednak 9 i M (broj djece koji stane u vlakić) jednak 3, najprije ćemo podijeliti djecu na skupinu od 5 i skupinu od 4 djece. Prva skupina od 5 djece rasporedit će se na dvije vožnje, tj. u dva vlakića: u jedan će se smjestiti troje, a drugi vlakić dvoje djece. Druga skupina od 4 djece rasporedit će se također na dvije vožnje, tj. u dva vlakića: u jedan će se smjestiti troje, a drugi vlakić jedno dijete. Ukupno će dakle biti potrebne četiri vožnje.

ULAZNI PODACI

U jedinom retku ulaza nalaze se prirodni brojevi N i M ($2 \leq N, M \leq 100$) iz teksta zadatka, odvojeni razmakom.

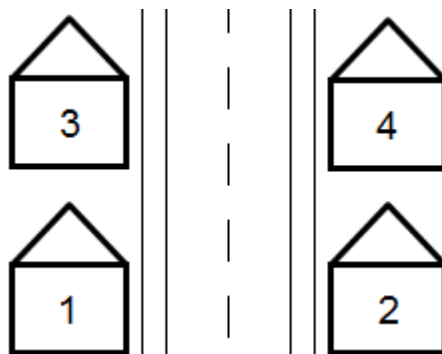
IZLAZNI PODACI

U jedini redak izlaza ispišite traženi broj vožnji.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
9 3	12 3	11 5
izlaz	izlaz	izlaz
4	4	3

Mirko i Slavko žive u istoj ulici. Kuće s neparnim brojevima su na lijevoj strani ulice, a one s parnim na desnoj. Ulica je jako dugačka, a na slici je prikazan njen početak:



Mirko raznosi novine kućama u ulici. **Kreće od svoje kuće**, a svaka sljedeća kuća kojoj će dostaviti novine **mora** biti kuća susjedna trenutačnoj, na istoj strani ceste, ili kuća točno preko puta trenutačne, na drugoj strani ceste. Istoj kući smije više puta dostaviti novine.

Budući da je ulica jako dugačka, odlučio je da će dostaviti novine samo nekim kućama. Koliko najmanje novina mora dostaviti tako da završi dostavljanje u Slavkovoj kući?

Na primjer, ako je Mirkov kućni broj 1, a Slavkov 4, Mirko može odmah prijeći cestu, dostaviti novine kući broj 2 pa dostaviti novine Slavku u kući broj 4. Jedini drugi način da dostavi dvoje novine je da prvo dostavi novine kući broj 3 te prijeđe cestu i dostavi novine Slavku u kući broj 4.

ULAZNI PODACI

U prvom retku ulaza nalaze se Mirkov i Slavkov kućni broj, tim redoslijedom. Oba kućna broja manja su od 100.

IZLAZNI PODACI

U jedinom retku izlaza ispišite traženi broj novina.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
3 1	5 4	1 4
izlaz	izlaz	izlaz
1	2	2

U propozicijama ovog natjecanja piše: “Za svako kolo natjecanja bit će pripremljeno 8 zadataka. Svaki natjecatelj može rješavati bilo koji zadatak po svom izboru, ali ukupni broj osvojenih bodova bit će jednak zbroju bodova 5 zadataka koji tom natjecatelju ukupno donose najviše bodova.”

Kako su organizatori bili zauzeti smišljenjem zadataka za ovo kolo, jednostavno su zaboravili riješiti problem određivanja ukupnog broja bodova za svakog natjecatelja. Učini to ti umjesto njih! Hvala!

Napiši program koji za zadan broj bodova koji je natjecatelj dobio na svakom zadatku ispisuje ukupan broj bodova tog natjecatelja na ovom kolu te redom oznake onih 5 zadataka koji su u zbroju dali taj ukupan broj bodova. Natjecatelj nikad neće na neka dva zadatka dobiti isti broj bodova.

ULAZNI PODACI

U osam redaka ulaza se nalazi po jedan prirodan broj X ($0 \leq X \leq 150$). Pri tome je vrijednost u i -tom retku broj bodova koje je natjecatelj dobio za svoje rješenje i -tog zadatka. Brojevi su međusobno različiti.

IZLAZNI PODACI

U prvi redak izlaza treba ispisati ukupan broj bodova dotičnog natjecatelja na ovom kolu. U drugi redak treba ispisati oznake traženih pet zadataka, poredane od manje oznake prema većoj i odvojene razmakom. Oznake zadataka su prirodni brojevi od 1 do 8.

BODOVANJE

Ako je točan samo ukupan broj bodova (bez obzira jesu li ispisane oznake traženih 5 zadataka), dobit će se **40% vrijednosti** test primjera.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
20	20	20
30	0	30
50	50	50
48	80	80
33	77	110
66	110	11
0	56	0
64	48	85
izlaz	izlaz	izlaz
261	373	355
3 4 5 6 8	3 4 5 6 7	2 3 4 5 8

Mirko je krenuo u autoškolu i još uvijek **ne zna okrenuti auto u uskoj ulici**. Stoga je odlučio otići u grad u kojem nigdje nije dopušteno takvo okretanje. Ova zabrana može se predočiti prometnim znakom:



Mirko je brzo zaključio kako u njegovom gradu ne smije biti slijepih ulica, jer se iz njih ne može izaći osim opisanim okretanjem. Napišite program za Mirka koji prima izgled grada i određuje je li taj grad Mirku povoljan za njega ili nije – tj. ima li u njemu slijepih ulica.

Grad je prikazan tablicom od $R \times S$ polja, od kojih je svako polje zgrada (označena slovom X) ili polje slobodno za promet (označeno točkom). Iz svakog slobodnog polja može se prijeći u neko od njemu susjedna četiri polja (gore, dolje, lijevo ili desno) pod uvjetom da je slobodno.

Formalno, reći ćemo da Mirkov grad nema slijepih ulica ako se, krenemo li iz bilo kojeg slobodnog polja u bilo kojem dozvoljenom smjeru, uvijek možemo vratiti u polazno polje bez da se okrećemo za 180 stupnjeva – tj. bez da u nekom trenutku promijenimo smjer kretanja u suprotan.

ULAZNI PODACI

U prvom retku ulaza nalaze se prirodni brojevi R i S ($3 \leq R, S \leq 10$), dimenzije grada.

U sljedećih R redaka nalazi se po S znakova, od kojih je svaki „X“ (iks) ili „.“ (točka). Ovih $R \times S$ znakova predstavlja mapu grada kako je opisano u tekstu zadatka. Barem dva polja bit će slobodna, a sva slobodna polja bit će povezana.

IZLAZNI PODACI

U jedini redak izlaza ispišite 0 ako Mirkov grad nema slijepih ulica. Inače, ispišite 1.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
4 3	5 5	3 9
XXX	XX.XX	...XXX...
X.X	X...X	.X.....X.
X.XXXX...
XXX	X...X	
izlaz	izlaz	izlaz
1	1	0

Mirko za zadaću iz matematike treba izračunati **najveći zajednički djelitelj** prirodnih brojeva **A** i **B**. Kako su brojevi bili predugački, učiteljica mu je dala manjih **N** prirodnih brojeva čiji je umnožak upravo broj **A**, kao i **M** prirodnih brojeva čiji je umnožak **B**.

Mirko bi želio provjeriti svoj rezultat, pa vas moli da napišete program koji rješava njegov zadatak.

Ukoliko rezultat ima više od 9 znamenaka, ispišite samo **posljednjih 9**.

ULAZNI PODACI

U prvom retku ulaza nalazi se prirodan broj **N** ($1 \leq N \leq 1000$).

U sljedećem retku nalazi se **N** prirodnih brojeva manjih od 1 000 000 000, čiji je umnožak **A**.

U trećem retku ulaza nalazi se prirodan broj **M** ($1 \leq M \leq 1000$).

U sljedećem retku nalazi se **M** prirodnih brojeva manjih od 1 000 000 000, čiji je umnožak **B**.

IZLAZNI PODACI

U jedini redak izlaza ispišite najveći zajednički djelitelj brojeva **A** i **B**. Ako on ima više od 9 znamenaka, ispišite samo posljednjih 9.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
3	4	3
2 3 5	6 2 3 4	358572 83391967 82
2	1	3
4 5	1	50229961 1091444 8863
izlaz	izlaz	izlaz
10	1	000012028

Pojašnjenje prvog primjera: najveći zajednički djelitelj brojeva $A = 30$ i $B = 20$ iznosi 10.

Nakon što je uspješno riješio zadaću iz prethodnog zadatka, Mirku je na satu matematike postalo dosadno, pa je na papir napisao N velikih brojeva. Potom je primijetio da mu se neki parovi od tih brojeva sviđaju, a neki drugi parovi ne sviđaju.

Za parove brojeva koji mi se sviđaju Mirko je skovao i naziv. Za **dva broja** reći ćemo da su **kompici** ako imaju **barem jednu zajedničku znamenku** (ne nužno na istoj poziciji).

Pomozite Mirku i recite mu koliko na njegovom papiru ima parova brojeva koji su kompici.

ULAZNI PODACI

U prvom retku ulaza nalazi se prirodan broj N ($1 \leq N \leq 1\,000\,000$).

U sljedećih N redaka nalazi se po jedan prirodan broj iz intervala $[1, 10^{18}]$. To su brojevi koje je Mirko napisao na papir; među njima neće biti jednakih.

IZLAZNI PODACI

U jedini redak izlaza ispišite traženi broj parova.

PRIMJERI TEST PODATAKA

ulaz	ulaz
3	4
4	32
20	51
44	123
	282
izlaz	izlaz
1	4

Mirko je napisao sljedeću funkciju:

```
int fun() {  
    int ret = 0;  
    for (int a = X1; a <= Y1; ++a)  
        for (int b = X2; b <= Y2; ++b)  
            ...  
            for (int <N-to> = XN; <N-to> <= YN; ++<N-to>)  
                ret = (ret + 1) % 1000000007;  
    return ret;  
}
```

```
function fun: longint;  
var  
    ret: longint;  
    a, b, ... , y, z: longint;  
begin  
    ret := 0;  
    for a := X1 to Y1 do  
        for b := X2 to Y2 do  
            ...  
            for <N-to> := XN to YN do  
                ret := (ret + 1) mod 1000000007;  
    fun := ret;  
end;
```

<N-to> označava N-to malo slovo engleske abecede. Svaki X_i i Y_i označava prirodan broj manji ili jednak 100 000 ili naziv varijable po kojoj iterira neka viša petlja. Na primjer, X_3 može biti isključivo a, b ili neki broj. X_i i Y_i nisu oba nazivi varijabli ni za koji i.

Odredite vrijednost koju funkcija vraća.

ULAZNI PODACI

U prvom retku ulaza nalazi se prirodan broj N ($1 \leq N \leq 26$).

U i -tom od sljedećih N redaka ulaza nalaze se X_i i Y_i . Ako su X_i i Y_i oba brojevi, $X_i \leq Y_i$.

IZLAZNI PODACI

U prvom i jedinom retku izlaza ispišite vrijednost koju funkcija vraća.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
2	3	3
1 2	2 3	1 2
a 3	1 2	a 3
izlaz	1 a	1 b
5	izlaz	izlaz
	10	11

Mirkova pizzerija najbolja je u gradu. Toliko je dobra da svi stanovnici svaki dan ručaju pizzu. Dostavljači koji rade za Mirka toliko su brzi da je vrijeme dostave zanemarivo. Problem nastaje kod pečenja pizze, jer svaki stanovnik ima svoj odabir sastojaka, pa tako pečenje pizze za dva stanovnika ne mora trajati jednako dugo. Budući da ima malu peć u koju stane samo jedna pizza, Mirku je vrlo bitan raspored pečenja koji mora biti određen **prije početka dana**.

Za svakog od N stanovnika (označenih od 1 do N) poznato je koliko vremena traje pečenje njegove pizze (V_i) kao i trenutak u danu kada planira ručak (R_i). Ako neki stanovnik dobije pizzu K trenutaka prije planiranog ručka, Mirko dobiva napojnicu od K kuna. Ako pak pizza stigne K trenutaka nakon planiranog ručka, Mirko njemu plaća K kuna (kao odštetu). U slučaju da pizza stigne točno na vrijeme, Mirko ne plaća niti dobiva napojnicu.

Mirka zanima **maksimalna suma svih napojnica** (uključujući i odštete kao negativne napojnice) koju može dobiti u jednom danu ako pizze peče optimalnim redoslijedom. Primijetite da Mirko može biti i u minusu (ako plati više nego što dobije).

Budući da stanovnici ponekad mijenjaju sastojke koje žele u pizzi, kao i trenutak u kojem će ručati, Mirkov se raspored mora adekvatno mijenjati kako bi zaradio što veću ukupnu napojnicu. Napišite program koji računa maksimalnu ukupnu Mirkovu napojnicu za početne zahtjeve stanovnika, kao i nakon svake promjene.

Napomena: U ovom gradu dan počinje u trenutku $t = 0$ i traje puno dulje od vremena potrebnog za pečenje pizze za sve stanovnike. Raspored kao i sve zadane promjene rade se prije početka dana.

ULAZNI PODACI

Prvi redak ulaza sadrži dva prirodna broja: N i P , broj stanovnika i broj promjena u zahtjevima (redom).

Sljedećih N redaka sadrži po dva prirodna broja: R_i (trenutak u kojem i -ti stanovnik planira ručak) i V_i (vrijeme potrebno za pečenje pizze za i -tog stanovnika).

Sljedećih P redaka sadrži po tri prirodna broja: S (indeks stanovnika), R (promijenjeni trenutak u kojem stanovnik S odlučuje ručati) i V (novo vrijeme potrebno za pečenje pizze za S -tog stanovnika).

Ograničenja:

$$1 \leq N, P \leq 200\,000,$$

$$0 \leq R_i, R \leq 100\,000,$$

$$1 \leq V_i, V \leq 100\,000,$$

$$1 \leq S \leq N.$$

IZLAZNI PODACI

Prvi redak izlaza mora sadržavati maksimalnu ukupnu napojnicu s obzirom na početne zahtjeve stanovnika.

Nakon svake od P promjena potrebno je ispisati novo izračunatu maksimalnu ukupnu napojnicu (svaku u zasebni redak).

BODOVANJE

U test podacima ukupno vrijednima barem 50% bodova vrijedi $1 \leq V_i, V \leq 1000$.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
3 2	4 2	6 7
10 2	3 2	17 5
6 5	0 3	26 4
4 3	4 3	5 5
1 6 1	4 1	12 4
3 0 10	3 0 4	8 1
izlaz	1 4 5	18 2
3	izlaz	3 31 3
2	-8	4 11 5
-11	-13	4 19 3
	-18	5 23 2
		6 15 1
		5 19 1
		3 10 4
		izlaz
		27
		59
		56
		69
		78
		81
		82
		58

Pojašnjenje prvog primjera: Optimalno je pizze ispeći redoslijedom (1, 3, 2). Tako će prva biti ispečena u trenutku $t = 2$, treća u trenutku $t = 5$, druga u $t = 10$. Prva je će uraniti osam trenutaka pa Mirko dobiva 8 kuna, druga kasni jedan trenutak (-1 kuna), treća kasni četiri trenutka (-4 kune), ukupno 3 kune. Nakon što prvi stanovnik promijeni zahtjeve, redoslijed ostaje isti, a napojnice postaju redom 5, 0 i -3. Nakon druge promjene optimalni redoslijed je (1, 2, 3) a napojnice redom 5, 0 i -16.