

ZADATAK	IGRA	MNOŽENJE	FIRMA
izvorni kôd	igra.pas igra.c igra.cpp igra.cxx	mnozenje.pas mnozenje.c mnozenje.cpp mnozenje.cxx	firma.pas firma.c firma.cpp firma.cxx
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		3 sekunde
memorijsko ograničenje	256 MB		
broj bodova	50	60	90
	200		



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

Goran nije stigao smisliti zadatak za natjecanje jer je zadnjih par dana proveo zabavljajući se najboljom igrom ikada. Nadam se da nemate planove za sljedećih par sati jer će je upravo podijeliti s vama!

U igri se koriste dvije matrice dimenzija četiri puta četiri (matrica je kvadratna mreža koja se sastoji od određenog broja redaka i stupaca). U obje matrice elementi su nenegativni cijeli brojevi.

Matrica P je *matrica prioriteta* i ona je fiksna - ne mijenja se tijekom igre. Matrica prioriteta sadrži sve prirodne brojeve između 1 i 16 i to svaki od njih **točno jednom**.

Matrica M je *matrica stanja* i ona se mijenja tijekom igre prema pravilima koja ćemo opisati. Na početku igre vrijednost svakog polja matrice M je jednaka 0. Za svako polje matrice M njegov *prioritet* definiramo kao vrijednost polja na istoj poziciji (u istom retku i stupcu) u matrici P .

Igra se sastoji od niza poteza od kojih svaki donosi određeni broj bodova. Broj bodova na početku igre iznosi 0.

U svakom potezu se mijenja matrica stanja i dobivaju bodovi prema sljedećim pravilima:

- Na početku poteza, od svih polja matrice stanja koje imaju vrijednost 0, ono **najvećeg prioriteta** se mijenja te postavlja na vrijednost 1. Ako nema niti jedno polje vrijednosti 0, matrica stanja ostaje nepromijenjena.
- Zatim, igrač unošenjem znakova na tipkovnici odabire točno jedan od četiri moguća tipa poteza:
 - Tipkom 'a' odabire se potez prema lijevo. Potez se odvija u tri koraka:
 - *Poravnanje retka ulijevo*: u svakom retku matrice M sve vrijednosti veće od 0 zapisuju se istim redoslijedom počevši od najlijevijeg polja u istom retku na desno. Ostala polja u retku poprimaju vrijednost 0. Drugim riječima, redak se preuredi tako da na lijevo dolaze brojevi veći od nule, na desno dolaze nule, a relativni poredak brojeva većih od nule ostane isti. Tako na primjer redak [0 2 0 1] postaje [2 1 0 0].
 - *Spajanje susjednih*: s lijeva na desno promatramo po dva susjedna polja u retku (najprije prvi i drugi, pa drugi i treći pa treći i četvrti). Ako neka dva polja **veća od 0** u trenutku promatranja imaju jednaku vrijednost, vrijednost lijevog polja odmah povećavamo za 1, a vrijednost desnog odmah postaje 0. **Ukupan broj bodova povećava se za novu vrijednost lijevog polja**.
 - Još jednom se ponovi prvi korak, tj. ponovo se redak poravna ulijevo.
 - Tipkom 's' odabire se analogan potez na stupcima prema dolje.
 - Tipkom 'd' odabire se analogan potez na retcima prema desno.
 - Tipkom 'w' odabire se analogan potez na stupcima prema gore.

Na primjer, pravilo za tipku 's' dobijemo tako da uzmemo pravila za potez prema lijevo te zamijenimo pojam 'redak' sa pojmom 'stupac', pojam 'lijevo' sa pojmom 'dolje', pojam 'desno' sa pojmom 'gore', pojam 'najlijeviji' sa pojmom 'najdonji' itd.

Ilustrirat ćemo pravila tako da prođemo kroz nekoliko poteza. U ovom objašnjenju promatramo samo jedan redak matrice te **ilustriramo samo drugi dio poteza** kada igrač odabere slovo na tipkovnici.

Primjer poteza udesno tipkom 'd':				
Početno stanje retka	1	0	1	0
Stanje nakon 1. koraka - "poravnanje retka udesno" ¹	0	0	1	1
Stanje nakon 2. koraka - "spajanja susjednih"	0	0	0	2
Stanje nakon 3. koraka - "poravnanje retka udesno"	0	0	0	2

¹ U originalnom tekstu zadatka je greškom pisalo 'ulijevo'.

Primjer poteza ulijevo tipkom 'a':				
Početno stanje retka	2	2	2	2
Stanje nakon 1. koraka - "poravnanje retka ulijevo"	2	2	2	2
Stanje nakon 2. koraka - "spajanje susjednih"	3	0	3	0
Stanje nakon 3. koraka - "poravnanje retka ulijevo"	3	3	0	0

Primjer poteza udesno tipkom 'd':				
Početno stanje retka	2	0	2	2
Stanje nakon 1. koraka - "poravnanje retka udesno"	0	2	2	2
Stanje nakon 2. koraka - "spajanje susjednih"	0	2	0	3
Stanje nakon 3. koraka - "poravnanje retka udesno"	0	0	2	3

Napišite program koji će za zadanu matricu prioriteta i niz poteza ispisati trenutno stanje matrice i bodove nakon svakog poteza.

ULAZNI PODACI

U prva četiri retka reda zadana je matrica P i to tako da se u svakom redu nalaze četiri prirodna broja u rasponu od 1 do 16 (uključivo) odvojena razmakom. Dodatno, svaki broj u matrici P je jedinstven.

U sljedećem redu ulaza nalazi se prirodan broj N ($1 \leq N \leq 1\,000$), broj poteza.

U svakom od sljedećih N redova ulaza nalazi se jedno od malih slova 'a', 's', 'd' ili 'w' - tipka kojom je odigran potez.

IZLAZNI PODACI

Izlaz se treba sastojati od N blokova gdje i -ti blok sadrži trenutni broj bodova i trenutnu matricu stanja nakon i -tog poteza.

Svaki blok se sastoji od pet redova. Prvi red bloka sadrži jedan cijeli broj - trenutni broj bodova. Zatim slijede četiri reda koja opisuju matricu stanja M u tom trenutku. Svaki red sadrži točno četiri prirodna broja odvojena razmakom - odgovarajući redak matrice M u tom trenutku.

PRIMJERI TEST PODATAKA

ulaz 5 4 15 9 2 12 13 10 3 11 8 14 6 16 1 7 4 a a w d izlaz 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 2 2 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 2 1 0 0 0 0 0 0 0 0 0 0 0 1	ulaz 10 2 3 14 5 16 7 8 9 1 12 11 13 4 15 6 5 s a d w w izlaz 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 0 0 2 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 1 2 0 1 0 2 0 0 0 1 0 0 0 0 0 0 0 0 4 0 2 0 2 0 0 0 1 0 0 0 0 0 0 0 0	ulaz 15 14 4 8 6 7 5 11 16 10 13 3 2 12 9 1 6 d w s d s d izlaz 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 2 0 0 1 2 0 0 0 0 0 0 0 0 0 0 0 1 0 0 2 1 4 0 0 0 0 0 0 0 0 0 0 0 0 1 0 2 2 7 0 0 0 0 0 0 0 0 0 0 0 1 0 0 1 3
---	---	--

Mirko uči tablicu množenja. Kako bi provjerio svoje znanje, zamislio je tablicu množenja veličine N puta N gdje se na i -tom retku j -tog stupca nalazi broj $i*j$ za sve prirodne brojeve i i j između 1 i N . Potom Mirko poreda svih N^2 elemenata tablice u niz od najmanjeg prema najvećem te naglas izgovori K -ti broj iz tog niza.

Napišite program pomoću kojeg Mirko može provjeriti je li uspješno izračunao taj broj.

Napomena: Savjetujemo da koristite 64-bitne cjelobrojne tipove podataka (long long u C/C++-u, int64 u Pascalu).

ULAZNI PODACI

U prvom i jedinom retku ulaza nalaze se prirodni brojevi N i K ($2 \leq N \leq 100\,000$, $1 \leq K \leq N^2$), kao iz teksta zadatka.

IZLAZNI PODACI

Ispišite K -ti najmanji broj koji se nalazi na zamišljenoj tablici množenja veličine N puta N .

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
4 13	4 16	51412 2643193744
izlaz	izlaz	izlaz
9	16	2643193744

Pojašnjenje prvog test primjera: Donja ilustracija prikazuje tablicu koju je Mirko zamislio. Kada elemente te tablice poredamo od najmanjeg prema najvećeg, dobivamo niz [1, 2, 2, 3, 3, 4, 4, 4, 6, 6, 8, 8, 9, 12, 12, 16]. 13. element tog niza je broj 9.

1	2	3	4
2	4	6	8
3	6	9	12
4	8	12	16

Tabela 1: Tablica množenja četiri puta četiri

U firmi za brzu i kvalitetnu prehranu 'Gangsta Sushi d.o.o' radi N radnika označenih redom brojevima od 1 do N . Svaki radni dan je podijeljen na sitne vremenske jedinice koje nazivamo trenutci. Trenutaka u radnom vremenu jednog dana ima točno 10^9 te su redom označeni cijelim brojevima 0, 1, 2, ..., 10^9-1 .

Svaki radnik dođe svaki dan na posao u nekom trenutku, ostane na poslu određeni broj trenutaka te nakon toga ode sa posla i više se ne vraća do sljedećeg dana. Marljivost radnika K je prirodni broj koji označavamo s M_K te označava razliku između zadnjeg i prvog trenutka u kojima je radnik bio na poslu. Dakle, ako radnik K dođe neki dan na posao u trenutku T onda je toga dana na poslu u točno trenutcima $T, T+1, T+2, \dots, T+M_K$ (uključivo). Marljivost svakog pojedinog radnika je ista svaki dan, ali različiti radnici mogu imati različite marljivosti.

Radnici firme se dijele u dvije grupe: kasnioci i ranioci.

- *Kasnioci* svaki dan **dolaze i odlaze** s posla jedan **trenutak kasnije** nego prethodni dan. Dakle, ako je kasnioc u jednom danu stigao na posao u trenutku T , sljedeći dan će stići u trenutku $T+1$.
- *Ranioci* svaki dan **dolaze i odlaze** s posla jedan **trenutak ranije** nego prethodni dan. Slično, ako je ranioc u jednom danu stigao na posao u trenutku T , sljedeći dan će stići u trenutku $T-1$.

Za svakog radnika poznata je njegova marljivost te je li kasnioc ili ranioc. Također, poznato je točno u kojem trenutku je koji radnik došao na posao prvog dana 2014. godine. Zanima nas koliko se najviše radnika nalazilo na poslu u istom trenutku u istom danu tijekom razdoblja od točno D dana koji počinje s prvim danom 2014. godine (i uključuje taj dan).

U ulazu se nalazi P različitih scenarija, napišite program koji će za svaki od njih odrediti taj najveći broj radnika kako je opisano gore.

ULAZNI PODACI

Prvi red ulaza sadrži prirodni broj P ($1 \leq P \leq 5$) - broj scenarija koje je potrebno riješiti. U nastavku slijedi P blokova gdje svaki blok sadrži opis jednog scenarija.

Prvi red svakog bloka sadrži dva prirodna broja N i D ($1 \leq N \leq 200\,000$, $1 \leq D \leq 1\,000\,000\,000$) - broj radnika i broj dana u razdoblju koji nas zanima. U svakom od sljedećih N redova bloka nalaze se podaci o jednom zaposleniku - najprije znak Z_K koji označava tip zaposlenika (veliko slovo 'K' za kasnioce i veliko slovo 'R' za raniocce) te dva cijela broja T_K i M_K ($0 \leq T_K \leq 1\,000\,000\,000$, $1 \leq M_K \leq 1\,000\,000\,000$), trenutak dolaska na posao prvi dan² razdoblja te marljivost zaposlenika.

Napomena: Ulazni podaci bit će takvi da niti jedan zaposlenik neće doći na posao prije trenutka 0 ili ostati na poslu nakon trenutka 10^9-1 u niti jednom od D dana.

IZLAZNI PODACI

Potrebno je ispisati P redova - rješenja za pojedine scenarije onim redoslijedom kojim dolaze u ulazu.

² U originalnom tekstu zadatka je greškom pisalo 'Drugi dan'.

PRIMJERI TEST PODATAKA

ulaz 3 2 1 K 1 1 K 2 1 2 1 K 1 1 R 4 1 2 2 K 1 1 R 4 1 izlaz 2 1 2	ulaz 1 6 6 K 5 9 R 6 13 R 6 8 R 14 14 K 1 4 R 8 15 izlaz 5	ulaz 3 4 1000 K 1000 1 K 1001 1 R 1005 1 R 1006 1 3 3 R 9 11 K 3 9 R 3 6 3 3 R 6 9 K 5 10 R 8 10 izlaz 3 3 3
--	--	--