

zadaci

zadatak	kasino	polja
izvorni kôd	kasino.pas kasino.c kasino.cpp	polja.pas polja.c polja.cpp
ulazni podaci	biblioteka	standardni ulaz
izlazni podaci	biblioteka	standardni izlaz
vremensko ograničenje (intel celeron 2.4 ghz)	1 sekunda	2 sekunde
memorijsko ograničenje (heap)	64 MB	64 MB
memorijsko ograničenje (stack)	16 MB	16 MB
broj bodova	100	100
	200	

Stroj za miješanje karata u jednom kasinu radi na sljedeći način:

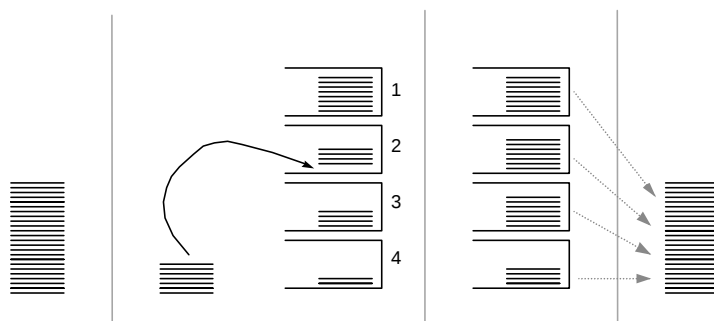
Špil se sastoji od N različitih karata označenih brojevima od 1 do N ($1 \leq N \leq 1000$), dok se stroj sastoji od K ($1 \leq K \leq 5$) pretinaca označenih brojevima od 1 do K .

Na početku su karte sortirane uzlazno tj. na vrhu špila se nalazi karta označena brojem 1, a na dnu špila karta označena brojem N .

U svakom koraku miješanja karata stroj odabere dva slučajna broja A i B , $1 \leq A \leq K$, $1 \leq B \leq 2$, te kartu s vrha špila stavlja u pretinac A . Ako je B jednako 1, onda se karta stavlja na vrh špila u tom pretincu, a ako je B jednako 2 onda se ona stavlja na dno.

Nakon što se na taj način prebace sve karte iz špila, pretinci se spajaju na način da karte iz prvog pretinca idu na vrh, karte iz drugog pretinca ispod njih, itd...

Primjer miješanja karata:



Sada u kasino ulaze Mirko i Slavko, uzimaju tako promiješani špil karata i igraju jednu zanimljivu igru.

Mirko uzima špil karata, a Slavko igra igru pogađanja karata koja se sastoji od N koraka.

U svakom koraku Slavko pokušava pogoditi koja se karta nalazi na nekoj poziciji u špil. Preciznije, Slavko kaže neka dva broja P i R , $1 \leq P, R \leq N$, a Mirko mu kaže koja se karta nalazi na poziciji P (karta na vrhu se nalazi na poziciji 1, a karta na dnu na poziciji N). Ako se na toj poziciji nalazi karta označena brojem R , Slavko **dobiva pet kuna**, inače **gubi jednu kunu**. Za svaku poziciju tj. broj od 1 do N , Slavko mora pitati **točno jednom**.

Na početku igre Slavko ima **nula kuna**, a dozvoljeno je da tijekom igre Slavkov iznos kuna ode u minus.

Napišite program koji će pomoću zadanih funkcija iz biblioteke igrati ovu igru tako da je na kraju igre Slavko sigurno u dobitku tj. da **ima pozitivan iznos kuna**.

Možete pretpostaviti da biblioteka miješa karte točno kako je gore opisano tj. u svakom koraku se A i B biraju slučajno, sve mogućnosti s istom vjerojatnošću.

biblioteka

Za rješavanje zadatka na raspolaganju je biblioteka funkcija **kasinolib** koja sadrži dvije funkcije:

Init - ovu funkciju je potrebno pozvati točno jednom i to na početku vašeg programa. Ona vraća vrijednosti N i K.

```
procedure Init(var n,k : integer);  
void Init(int *n,int *k);
```

Igraj - ova funkcija se poziva s dva argumenta - brojevima P i R koje Slavko kaže Mirku u igri.

```
procedure Igraj(p,r : integer);  
int Igraj(int p,int r);
```

Upute za natjecatelje u Pascal-u

Na početku vašeg programa mora se nalaziti naredba 'uses kasinolib'.

Također, potrebno je preuzeti biblioteku s evaluacijskog sustava.

Upute za natjecatelje u C/C++

Potrebno je preuzeti datoteke kasinolib.h i kasinolib.o s evaluacijskog sustava.

U RHIDE-u otvorite projekt (Project->Open) i nazovite ga kasino. U njega naredbom Project->AddItem dodajte datoteke kasino.c ili kasino.cpp (vaš program) i kasinolib.o.

Izvršnu datoteku možete kreirati i kompajliranjem pomoću komandne linije (npr. za C++)

```
g++ -o kasino.exe kasino.cpp kasinolib.o.
```

Na početku vašeg programa mora se nalaziti naredba '#include "kasinolib.h"'.

Važno: nemojte koristiti naredbu Compile->BuildAll (jer ona briše datoteku kasinolib.o).

Testiranje

Vaš program možete testirati tako da na standardni ulaz unesete tri broja: N, K i seed za generator slučajnih brojeva. Pokrenete li program više puta s istim seedom, dobit ćete iste rezultate, što je korisno jer možete testirati svoj program mnogo puta na istom test podatku. Ukoliko programu proslijedite -1 kao seed, test podatak će biti slučajan.

Biblioteka prilikom izvršavanja kreira datoteku kasino.log u kojoj možete vidjeti detaljni prikaz poziva koje je napravio vaš program.

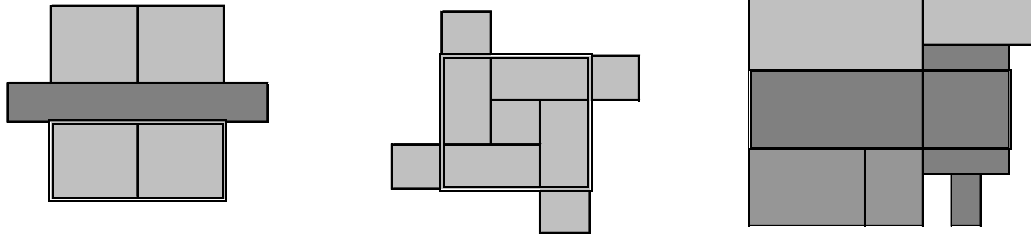
polja

Mirko se nalazi u avionu nad nepreglednom ravnicom na kojoj se nalaze polja pravokutnog oblika i različitih boja, ovisno o vrsti biljke koja se na tom polju uzgaja.

On u nevjerici gleda kroz prozor ta silna prostranstva i silno ga zanima kolika je **površina najvećeg jednobojnog pravokutnog polja** koje vidi.

Nama je poznat raspored polja na tlu. Međutim, među poljima iste boje nema vidljivih granica tj. područje u nekoj boji za koje mi znamo da se sastoji od više polja Mirku izgleda kao samo jedno polje.

Na slici možete vidjeti test primjere s dna stranice (naznačeno je i rješenje svakog primjera):



Napišite program koji će odrediti površinu traženog najvećeg jednobojnog pravokutnog polja.

Niti jedna dva polja se **neće** međusobno preklapati (dodirivanje stranica ne smatramo preklapanjem), a područja na tlu koja nisu obuhvaćena poljima ne uzimamo u gore navedena razmatranja.

ulazni podaci

U prvom retku se nalazi prirodni broj N , $1 \leq N \leq 2500$, broj pravokutnih polja na tlu koje zamišljamo kao koordinatni sustav u ravnini.

U svakom od sljedećih N redaka se nalazi pet cijelih brojeva $X1, Y1, X2, Y2$ ($X1 < X2, Y1 < Y2$) i B koji predstavljaju jedno pravokutno polje (stranica paralelnih s koordinatnim osima) s koordinatama nasuprotnih vrhova $(X1, Y1)$ i $(X2, Y2)$ obojano u boju B , $1 \leq B \leq 100$. Sve koordinate će biti cijeli brojevi između 0 i 1,000,000,000 uključivo.

izlazni podaci

U prvi i jedini redak treba ispisati traženu površinu iz teksta zadatka.

Napomena: rješenje će stati u 64-bitni cjelobrojni tip podataka (int64 u Pascalu, long long u C/C++).

test primjeri

ulaz

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

izlaz

8

ulaz

```
5
5 5 6 6 22
3 4 6 5 22
6 3 7 6 22
5 6 8 7 22
4 5 5 8 22
```

izlaz

9

ulaz

```
7
0 0 4 3 1
6 2 9 7 2
6 7 10 9 3
4 0 6 3 1
0 6 6 9 3
0 3 6 6 2
7 0 8 2 2
```

izlaz

27