

zadaci

zadatak	rojs	nosorog	točke	lunapark
izvorni kôd	rojs.pas rojs.c rojs.cpp	nosorog.pas nosorog.c nosorog.cpp	tocke.pas tocke.c tocke.cpp	lunapark.pas lunapark.c lunapark.cpp
ulazni podaci	biblioteka	standardni ulaz		
izlazni podaci	biblioteka	standardni izlaz		
memorijsko ograničenje (heap)	32 MB			
memorijsko ograničenje (stack)	4 MB			
vremensko ograničenje (pentium3 na 700 mhz)	2 sekunde			
broj bodova	25	25	25	25
	100			

Nakon što je jučer plaćući srušio kuću svoje kćeri, naš Ljubo je odlučio danas srušiti i svoju kuću.

Kako je jučer za rušenje pozvao u pomoć skoro sve svoje prijatelje, a oni su samo jeli, pili i pričali viceve umjesto da pomažu, danas je odlučio pozvati samo **trojicu**, kako bi se rušenje obavilo uz što manji trošak.

Pošto se neki od njegovih prijatelja međusobno ne poznaju, a da bi se sva trojica osjećala što ugodnije, odlučio je pozvati ili tri prijatelja koji se **svi međusobno poznaju**, ili tri prijatelja koji se **svi međusobno ne poznaju**.

Napišite program koji će izabrati tri prijatelja koji zadovoljavaju gore navedene uvjete.

Napomena: rješenje će uvijek postojati.

biblioteka

Za rješavanje zadatka na raspolaganju je biblioteka funkcija `rojs_lib` koja sadrži tri funkcije:

Init - ovu funkciju je potrebno pozvati **točno jednom** i to **na početku** vašeg programa, bez argumenata. Ova funkcija vraća cijeli broj N , $6 \leq N \leq 30$, broj Ljubinih prijatelja. Prijatelji su označeni brojevima od 1 do N .

```
function Init : longint;
```

```
int Init(void);
```

Pitaj - ova funkcija se poziva s dva argumenta, brojem A i brojem B , tj. rednim brojevima dvaju prijatelja. Funkcija vraća **0** ako se ta dva prijatelja **poznaju**, a **1** ako se **ne poznaju**. Brojevi A i B moraju biti međusobno različiti.

Ukupni broj pitanja postavljenih pomoću ove funkcije smije biti **najviše 15**.

```
function Pitaj(a, b : longint) : longint;
```

```
int Pitaj(int a, int b);
```

Rjesenje - ovu funkciju je potrebno pozvati **na kraju** vašeg programa; kao argument prosljeđujete redne brojeve neka tri prijatelja koja zadovoljavaju uvjete iz zadatka. Ova funkcija ne vraća nikakvu vrijednost i **regularno završava izvođenje** vašeg programa.

```
procedure Rjesenje(a, b, c : longint);
```

```
void Rjesenje(int a, int b, int c);
```

Uputa za natjecatelje u Pascal-u

Na početku vašeg programa mora se nalaziti naredba `'uses rojs_lib'`.

Datoteku `rojs_lib.pas` stavite u isti direktorij u kojem se nalazi i vaše rješenje `rojs.pas`.
Kompajlirajte s `'ppc386 -So -O1 -XS rojs.pas'`.

Uputa za natjecatelje u C/C++

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

Datoteku `rojs_lib.h` stavite u isti direktorij u kojem se nalazi i vaše rješenje `rojs.c` odnosno `rojs.cpp`.

Testiranje

Vaš program možete testirati tako da na standardni ulaz unesete opis grafa međusobnih poznanstava, a biblioteka će na standardni izlaz ispisivati poruke o rezultatima pozivanja funkcija iz biblioteke.

Opis grafa je u sljedećem obliku:

U prvom retku se nalazi cijeli broj N , broj prijatelja.

U svakom od sljedećih N redaka se nalazi po N brojeva, na taj način stvarajući matricu $N \times N$.

Element u A -tom retku i B -tom stupcu u toj matrici je 0 (ako se A i B poznaju) ili 1 (ako se A i B ne poznaju).

Primjer

ulaz

```
6
0 1 0 1 0 1
1 0 0 0 1 0
0 0 0 1 1 0
1 0 1 0 1 1
0 1 1 1 0 0
1 0 0 1 0 0
```

rješenje (jedno od mogućih)

Rjesenje(1,4,6)

Zadan je string (tj. niz znakova) S.

Dozvoljene operacije nad tim nizom znakova su:

P **ispisuje** prvi znak stringa S

B **briše** prvi znak stringa S

O **okreće** string S naopačke (prvi znak postaje zadnji, drugi predzadnji, itd...)

Napišite program koji će pronaći **neki niz** dozvoljenih operacija kojima ćemo **ispisati riječ NOSOROG**.

Napomena: rješenje će uvijek postojati.

ulazni podaci

U prvom i jedinom retku se nalazi string S, minimalne duljine 5 i maksimalne duljine 100 znakova.

Jedini dozvoljeni znakovi u stringu su velika slova engleske abecede ('A' - 'Z')

izlazni podaci

U prvi i jedini redak treba ispisati neki niz znakova koji će predstavljati operacije potrebne za ispis riječi NOSOROG.

Broj operacija mora biti **manji ili jednak od 500**.

test primjeri

ulaz

NOOS00R00G

izlaz

PBPBBPBPBBPBPBP

ulaz

ZZAAROG0SONS

izlaz

BBBB0BPBPBPBPBPBP

ulaz

NOGETISE0ROSILE

izlaz

PBP0BBBPBPBPBPBP

točke

Zadano je N točaka u pravokutnom koordinatnom sustavu.

Dvije točke $A(x_1, y_1)$ i $B(x_2, y_2)$ možemo spojiti samo ako je $x_1 = y_2$.

Napišite program koji će odrediti neku permutaciju svih zadanih točaka tako da se svake dvije susjedne točke u toj permutaciji mogu spojiti po gore navedenom pravilu.

Napomena: rješenje će uvijek postojati.

ulazni podaci

U prvom retku se nalazi cijeli broj N , $2 \leq N \leq 10,000$, broj točaka.

U svakom od sljedećih N redaka se nalaze po dva cijela broja, x i y koordinata jedne točke. Svaka od koordinata će biti cijeli broj veći ili jednak od 1 i manji ili jednak od 1000.

izlazni podaci

U N redaka treba ispisati svih N točaka traženim redoslijedom iz teksta zadatka.

test primjeri

ulaz

4
5 1
2 4
4 3
1 2

izlaz

4 3
2 4
1 2
5 1

ulaz

5
4 2
5 2
3 4
2 1
2 3

izlaz

2 1
4 2
3 4
2 3
5 2

ulaz

9
8 6
8 4
7 3
6 8
3 6
4 6
1 2
6 1
6 7

izlaz

1 2
6 1
3 6
7 3
6 7
4 6
8 4
6 8
8 6

lunapark

Mirko gradi lunapark u svom rodnom selu. Kao i u svakom drugom lunaparku glavna atrakcija je vlak smrti kojeg Mirko gradi na zemljištu **kvadratnog** oblika koje je predstavljeno kao **ploča dimenzija $N \times N$** . **Svako polje** na tom zemljištu ima neku **zadanu visinu**.

Rutu vlaka treba napraviti tako da vlak kreće s nekog polja u nekom od **četiri smjera** (gore, dolje, lijevo i desno), a nakon svakog pomaka može skrenuti **lijevo**, **desno** ili **nastaviti ravno**, te ruta završava u nekom od polja.

Između svaka dva susjedna polja rute definiramo **nagib** kao **razliku visine** nekog polja i njemu **prethodnog** polja na ruti. Dakle, nagib je **negativan** ako tračnice idu **nizbrdo**, a **pozitivan** ako tračnice idu **uzbrdo**.

Ako je nagib X između neka dva susjedna polja rute **pozitivan**, onda nagib Y između sljedeća dva polja rute **mora biti negativan**, a ako je X **negativan ili nula**, onda za nagib Y **mora vrijediti $Y < -X$** . Drugim riječima, ako je vlak išao **uzbrdo**, onda nakon toga mora ići **nizbrdo**, a ako je išao **nizbrdo**, onda nakon toga može ići **nizbrdo pod bilo kojim nagibom** ili **uzbrdo**, ali **manje nego što je išao nizbrdo** u zadnjem pomaku. Osim toga, ruta mora **započinjati nizbrdo** (tj. nagib između prvog i drugog polja rute mora biti negativan).

Napišite program koji će odrediti duljinu **najdulje moguće rute** vlaka na zadanom zemljištu.

Napomena: ruta smije sijeći samu sebe.

ulazni podaci

U prvom retku se nalazi cijeli broj N , $2 \leq N \leq 300$, stranica zemljišta.

U svakom od sljedećih N redaka se nalazi po N brojeva koji predstavljaju visinu svakog polja. Svaki od tih brojeva je veći ili jednak od 1 i manji ili jednak od 1,000,000.

izlazni podaci

U prvi i jedini redak treba ispisati duljinu najdulje moguće rute iz teksta zadatka.

test primjeri

ulaz

3
2 9 4
6 2 3
7 8 1

izlaz

7

ulaz

4
5 5 5 5
5 2 1 5
5 4 3 5
5 5 5 5

izlaz

5

ulaz

5
99 98 99 95 94
99 98 97 96 92
99 99 30 40 93
99 28 20 22 20
99 21 29 19 19

izlaz

19