
zadaci

zadatak	robot	mravi	nevenka	chip
izvorni kôd	robot.pas robot.c robot.cpp	mravi.pas mravi.c mravi.cpp	nevenka.pas nevenka.c nevenka.cpp	chip.pas chip.c chip.cpp
ulazni podaci	biblioteka	standardni ulaz		
izlazni podaci	biblioteka	standardni izlaz		
memorijsko ograničenje (heap)	16 MB	16 MB	16 MB	128 MB
memorijsko ograničenje (stack)	2 MB			
vremensko ograničenje (pentium4 na 1.6 ghz)	1 sekunda	2 sekunde	7 sekundi	4 sekunde
broj bodova	40	60	60	90
	250			

robot

Maloljetni Mirko želi provaliti u jednu praznu kuću na koju je prije dosta vremena bacio oko.

Kako bi je prije provale prvo istražio, na crnom tržištu je nabavio robota i teleportirao ga u unutrašnjost kuće.

Robot se sada nalazi negdje **unutar** kuće i Mirko njime upravlja pomoću jednostavne naredbe `Hodaj` koja robotu naređuje u kojem od **četiri smjera** da se pomakne (gore, desno, dolje ili lijevo) kako bi na taj način istražio sobu u kojoj se nalazi.

Tlocrt kuće možemo zamisliti kao pravokutnu mrežu koja se sastoji od manjih kvadratića organiziranih u retke i stupce. Neki od kvadratića predstavljaju **prazan prostor**, dok ostali predstavljaju **zidove**. U jednom od kvadratića koji predstavlja prazan prostor se nalazi i robot.

Sobu definiramo kao maksimalan skup susjednih kvadratića (u **vodoravnom** i **okomitom** smjeru, **ne u dijagonalnom!**).

Napišite program koji će odrediti **površinu sobe** (broj kvadratića na koje robot može doći) u kojoj se nalazi robot pomoću **najviše 10,000 poziva** funkcije `Hodaj`. Površina sobe će biti manja ili jednaka od **1000**.

biblioteka

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

Init - ovu funkciju je potrebno pozvati **točno jednom** i to **na početku** vašeg programa, bez argumenata. Ova funkcija ne vraća nikakvu vrijednost.

```
procedure Init;  
void Init(void);
```

Hodaj - ova funkcija se poziva s jednim argumentom - to je smjer u kojem želimo pomaknuti robota; **1 - gore, 2 - desno, 3 - dolje, 4 - lijevo**. Ako je pomak uspio, funkcija **pomiče** robota na traženu poziciju i vraća vrijednost **1**, a ako je na tom mjestu zid, pozicija robota se **ne mijenja** i funkcija vraća vrijednost **0**.

```
function Hodaj(smjer : longint) : longint;  
int Hodaj(int smjer);
```

Rjesenje - ovu funkciju je potrebno pozvati **na kraju** vašeg programa; kao argument prosljeđujete traženu površinu sobe. Ova funkcija ne vraća nikakvu vrijednost i **regularno završava izvođenje** vašeg programa.

```
procedure Rjesenje(povrsina : longint);  
void Rjesenje(int povrsina);
```

Upute za natjecatelje u Pascal-u

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

Također, potrebno je preuzeti datoteke `robotlib.o` i `robotlib.ppu` s evaluacijskog sustava.

Upute za natjecatelje u C/C++

Potrebno je preuzeti datoteke `robotlib.h` i `robotlib.o` s evaluacijskog sustava.

U RHIDE-u otvorite projekt (Project->Open) i nazovite ga `robot`. U njega naredbom Project->AddItem dodajte datoteke `robot.c` odnosno `robot.cpp` (vaš program) i `robotlib.o`.

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

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

Testiranje

Vaš program možete testirati tako da na standardni ulaz unesete sljedeće:

U prvom retku cijeli brojevi `R` i `S`, dimenzije kuće (redak i stupac)

U svakom od sljedećih `R` redaka po `S` znakova, tlocrt kuće. Znak `'X'` označava zid, znak `'.'` (točka) označava prazan prostor, a znak `'R'` označava poziciju robota.

Biblioteka će na standardni izlaz ispisivati opširne poruke o rezultatima pomicanja robota.

mravi

Po tankoj i dugačkoj napetoj žici gmižu mravi zanemarivih dimenzija.

Mravi se cijelo vrijeme kreću **konstantnom** brzinom od **1 mm/s**. Kada neki mrav dođe do neke prepreke (**drugog mrava** ili nekog **kraja žice**), odbije se od nje tj. **promijeni smjer** i nastavi se kretati u tom smjeru i dalje istom brzinom.

Zadan je **početni raspored** mrava na žici i za svakog od njih **inicijalni smjer** kretanja (**lijevo** ili **desno**). Mravi, kao što su poredani u početnom rasporedu, označeni su brojevima **od 1 do N**, **slijeva na desno**. Niti jedna dva mrava se **na početku ne nalaze** na istoj poziciji.

Napišite program koji će odrediti raspored mrava na žici nakon nekog zadanog vremena.

ulazni podaci

U prvom retku se nalaze dva cijela broja D i V , $2 \leq D \leq 200,000$, $1 \leq V \leq 1,000,000$, duljina žice (u milimetrima) i zadano vrijeme (u sekundama).

U drugom retku se nalazi cijeli broj N , $1 \leq N \leq 70,000$, broj mrava. Taj broj je također manji i od zadane duljine žice.

U svakom od sljedećih N redaka se nalazi početna pozicija i inicijalni smjer svakog mrava, redom od mrava označenog brojem 1 do mrava označenog brojem N .

To su cijeli broj i znak, međusobno odvojeni jednim razmakom, udaljenost mrava od lijevog kraja žice (u milimetrima) i inicijalni smjer svakog mrava. Inicijalni smjer je označen slovom **'L'** (**lijevo**) ili **'D'** (**desno**).

izlazni podaci

U prvi i jedini redak treba ispisati završne pozicije (tj. udaljenost od lijevog kraja žice, u milimetrima) svih mrava, redom od mrava označenog brojem 1 do mrava označenog brojem N .

test primjeri

ulaz	ulaz	ulaz
3 5	5 5	8 10
1	2	5
1 D	2 D	1 L
izlaz	4 L	3 L
0	izlaz	4 D
	1 3	6 L
		7 D
		izlaz
		1 2 4 7 7

nevenka

Mirko i Slavko su se zaljubili u svoju drugaricu iz razreda Nevenku.

Osim u školi, oni joj se i inače pokušavaju što više približiti pa tako i kada ona nedjeljom odlazi u crkvu. To se odvija na sljedeći način:

Grad u kojem oni žive zamislimo kao **neusmjereni povezani graf**. Neki vrhovi tog grafa su istaknuti i u njima se nalaze ili Nevenkina **kuća** ili **neka od crkvi** u koje ona voli ići ili **neki od kafića** u kojima se zadržavaju Mirko i Slavko.

Udaljenost dvaju vrhova u grafu definiramo kao najmanji broj bridova koji povezuju ta dva vrha.

Mirko i Slavko znaju gdje stanuje Nevenka i gdje su njezine omiljene crkve.

Te su se nedjelje Mirko i Slavko smjestili u neka **dva (različita)** kafića. Nevenka poznaje konobarice u svakom od kafića pa su joj one dojavile gdje su se smjestili Mirko i Slavko.

Kako Nevenka baš i ne voli njih dvojicu, čim je saznala gdje se oni nalaze, odlučila je **u koju** će crkvu i **kojim putem** ići kako bi u svakom trenutku na putu do crkve bila **što dalje** od bilo kojeg od njih dvojice (koji će je loviti, na način kako će poslije biti opisano)

No i Mirko i Slavko imaju prijatelja Branka koji živi odmah do Nevenkine kuće i čuo je kada je Nevenka roditeljima rekla u koju će crkvu i kojim putem ići. Branko to odmah javlja Mirku i Slavku i oni **istovremeno** kada i Nevenka izlazi iz kuće kreću u lov.

Lov se odigrava na sljedeći način:

Svi se kreću **istom brzinom** od jednog brida u minuti; nije nužno da su Mirko ili Slavko stalno u pokretu. Nevenka ide po svom svima poznatom putu bez obzira na to kako se Mirko i Slavku kreću, dok oni nastoje da joj **netko od njih dvojice u nekom trenutku** priđe na što **manju udaljenost**.

U trenutku kad ona uđe u crkvu, Mirko i Slavko prestaju s lovom na Nevenku.

Za neki par različitih kafića koje su Mirko i Slavko odabrali kažemo da **garantira najmanju udaljenost X**, ako se oni za bilo koju crkvu i bilo koji put koji Nevenka odabere, kao i za neki od načina njihovog kretanja za vrijeme lova, mogu približiti Nevenki na tu udaljenost.

Napišite program koji će pomoći Mirku i Slavku da **odrede neki par kafića** koji bi im garantirali **najmanju moguću** udaljenost X.

nevenka

ulazni podaci

U prvom retku se nalazi cijeli broj V , $5 \leq V \leq 1000$, broj vrhova u grafu.

U drugom retku se nalazi indeks vrha u kojem se nalazi kuća od Nevenke.

U trećem retku se nalazi popis vrhova u kojima se nalaze Nevenkine omiljene crkve. Prvi broj je broj crkvi (manji ili jednak od 200), a ostali brojevi su indeksi vrhova, uzlazno sortirani. U četvrtom retku se nalazi popis kafića. Prvi broj je broj kafića (manji ili jednak od 200), a ostali brojevi su indeksi vrhova, uzlazno sortirani.

U petom retku se nalazi cijeli broj B , broj bridova u grafu.

U svakom od sljedećih B redaka se nalaze po dva cijela broja $V1$ i $V2$, $V1 < V2$, koja nam kažu da se između vrhova s indeksima $V1$ i $V2$ nalazi brid.

izlazni podaci

U prvi i jedini redak treba ispisati indekse vrhova u kojima se nalaze traženi kafići iz teksta zadatka, prvo manji pa onda veći od njih.

Ako postoji više rješenja, ispisati bilo koje.

test primjeri

ulaz

```
14
4
3 8 9 11
3 1 7 14
13
1 2
2 3
3 4
4 5
5 6
5 7
6 8
3 9
3 10
10 11
10 12
12 13
13 14
```

izlaz

```
1 7
```

ulaz

```
17
17
2 4 15
3 1 9 11
18
1 2
2 3
2 10
3 4
4 5
5 6
5 17
6 7
7 8
8 9
10 11
10 17
11 12
12 13
13 14
14 15
15 16
16 17
```

izlaz

```
9 11
```

chip

Zamislimo dvodimenzionalnu pločicu **kvadratnog** oblika, smještenu u koordinatnom sustavu tako da se donji lijevi kut pločice nalazi u **ishodištu**, a gornji desni kut u nekoj točki s **cjelobrojnim** koordinatama.

Na pločici su zadane neke točke. Svaku od njih moramo spojiti s nekom od stranica pločice pomoću **točno jedne vodoravne** ili **okomite** dužine. Dvije različite dužine **ne smiju se međusobno sijeći**, niti neka dužina smije **prolaziti** kroz neku **drugu točku**.

Napišite program koji će za svaku zadanu točku odrediti dužinu koja ju spaja sa stranicom pločice, ali tako da **ukupna duljina** svih tih dužina bude **minimalna** i da vrijede gore navedena pravila.

ulazni podaci

U prvom retku se nalazi cijeli broj A , $2 \leq A \leq 30$, veličina stranice kvadratne pločice.

U drugom retku se nalazi cijeli broj N , $1 \leq N \leq 100$, broj zadanih točaka na pločici.

U svakom od sljedećih N redaka se nalaze dva cijela broja X i Y , $1 \leq X, Y \leq A-1$, koordinate neke točke na pločici. Na jednom mjestu na pločici smije se nalaziti samo jedna točka.

izlazni podaci

U prvi redak treba ispisati **ukupnu duljinu** traženih dužina.

U svaki od sljedećih N redaka treba ispisati jednu od četiri riječi '**GORE**', '**DOLJE**', '**LIJEVO**' ili '**DESNO**', zavisno od toga s kojom je stranicom pločice odgovarajuća (redom kojim su zadane) točka povezana.

Ako postoji više različitih rješenja minimalne ukupne duljine svih dužina, ispišite **bilo koje**. Ako ne postoji rješenje, ispišite **-1**.

test primjeri

ulaz	ulaz	ulaz
6	10	10
4	5	4
3 1	8 1	5 1
3 5	4 3	5 2
5 4	1 2	4 3
1 4	3 9	6 3
izlaz	8 5	izlaz
4	izlaz	13
DOLJE	8	DOLJE
GORE	DOLJE	DESNO
DESNO	DOLJE	DOLJE
LIJEVO	LIJEVO	DESNO
	GORE	
	DESNO	