

zadaci

zadatak	zvrk	xl	deva	vid
izvorni kôd	zvrk.pas zvrk.c zvrk.cpp	xl.pas xl.c xl.cpp	deva.pas deva.c deva.cpp	vid.pas vid.c vid.cpp
ulazni podaci	standardni ulaz	standardni ulaz	biblioteka	standardni ulaz
izlazni podaci	standardni izlaz	standardni izlaz	biblioteka	standardni izlaz
vremensko ograničenje (Athlon 64 3000+)	1 sekunda	1 sekunda	5 sekunde	2 sekunde
memorijsko ograničenje (heap)	32 MB	32 MB	64 MB	64 MB
memorijsko ograničenje (stack)	8 MB	8 MB	16 MB	16 MB
broj bodova	50	60	90	100
	300			

zvrk

Kada iz **binarnog zapisa** nekog prirodnog broja **uklonimo sve znamenke lijevo od najdesnije jedinice** dobit ćemo broj kojeg nazivamo **zvrk** početnog broja.

Npr. zvrk broja 6 tj. $(110)_2$ je 2 tj. $(10)_2$, a zvrk broja 40 tj. $(101000)_2$ je 8 tj. $(1000)_2$.

Napišite program koji će izračunati **zbroj zvrkova** svih brojeva **između** zadanih brojeva A i B (**uključivo**).

ulazni podaci

U prvom i jedinom retku se nalaze prirodni brojevi A i B, $1 \leq A \leq B \leq 10^{15}$.

izlazni podaci

U prvi i jedini redak treba ispisati traženi zbroj iz teksta zadatka.

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

test primjeri

ulaz

176 177

izlaz

17

ulaz

5 9

izlaz

13

ulaz

25 28

izlaz

8

Kvadratna ploča se sastoji od polja poredanih u N redaka i N stupaca. Svako polje može sadržavati **rimsku znamenku** ili biti **prazno**.

Primjer ploče iz 3. test primjera:

I	I	X	V		L	X
X	L		I	X	V	I
	I	V	I	X		X
L	I	X		V	I	X
X		X	I	X	I	
L	I	V	L		X	X
V	I		X	I	X	L

Na početku se nalazimo na polju **u sredini** ploče (broj N će biti neparan, a srednje polje prazno) i cilj nam je koracima u četiri glavna smjera (gore, dolje, lijevo i desno) "iskoračati" što dulji niz **uzastopnih** rimskih brojeva **počevši od broja 1**, pri tome koristeći **točno jedno prazno** polje za odvajanje brojeva.

Npr. kad krenemo iz sredine koraci *gore - dolje - gore - gore - lijevo - desno - dolje - gore - lijevo* sadrže prva tri rimska broja (*I - prazno - I - I - prazno - I - I - I - prazno*). Broj smatramo **potpuno iskoračanim** tek kada nakon njegovih znamenaka zakoračimo na prazno polje.

Rimske znamenke koje se mogu pojaviti na ploči su **I, V, X, L, C i D** odgovaraju redom dekadskim vrijednostima 1, 5, 10, 50, 100 i 500.

Rimski broj dobijemo iz dekadskog prikaza tako da pretvaramo znamenku po znamenku dekadskog prikaza, silazno po vrijednosti: tako je točan prikaz broja 499 u rimskom sustavu CDXCIX, a ne ID. Poseban slučaj su također i brojevi 4, 9, 40, 90 i 400, koje dobijamo tako da manju rimsku znamenku stavimo ispred veće (tako je npr. XL zapravo 40).

Napišite program koji će odrediti **koliko uzastopnih rimskih brojeva počevši od broja 1 sadrži najdulji niz** koji možemo na gore opisani način iskoračati na zadanoj ploči.

ulazni podaci

U prvom retku se nalazi neparni prirodni broj N , $1 \leq N \leq 99$.

U svakom od sljedećih N redaka nalazi se po N znakova koji predstavljaju izgled ploče. Jedinici znakovi koji će se pojaviti u opisu ploče su 'I', 'V', 'X', 'L', 'C', 'D' i '.' (točka), koja označava prazno polje.

izlazni podaci

U prvi i jedini redak treba ispisati traženi broj rimskih brojeva iz teksta zadatka.

test primjeri

ulaz	ulaz	ulaz
3	5	7
I.I	.IV.I	IIXV.LX
I.V	II.II	XL.IXVI
I..	XV.VI	.IVIX.X
izlaz	..I..	LIX.VIX
	...II	X.XIXI.
	izlaz	LIVL.XX
6		VI.XIXL
	11	izlaz
		51

Reality show "Survivor" popularan je u cijelom svijetu pa tako i u vrućoj Africi gdje se trenutno s krivotvorenim putovnicama nalaze i dvojica naših poznatih državljana. Kada su jednog dana u lokalnom tjedniku ugledali poziv na audiciju, istovremeno su se i značajno pogledali, odazvali se te prošli u finalno natjecanje i sada se zajedno s drugim natjecateljima dokazuju u raznim vještinama i igrama spretnosti.

Tako su naši junaci, koji se sada zovu Abdul i Yasser, od zoološkog vrta u Marrakeshu dobili prototip najnovijeg modela deve kako bi prije početka masovne proizvodnje ispitali neka njena mehanička svojstva i svojim konstruktivnim primjedbama pridonijeli što većem tržišnom uspjehu ovog hit-proizvoda, a uz to u jednoj, samo za njih dvojicu izmišljenoj igri, pokušali osvojiti glavnu nagradu u iznosu od 1,000,000 marokanskih dirhama (oko 222,000,000 malgaških arijarija)!

U pustinji se nalazi N gradova, $2 \leq N \leq 1000000$ (milijun). Neki gradovi su međusobno spojeni cestama, a svaki grad je direktnom cestom spojen s **najviše 5 drugih gradova**.

Abdul i Yasser će dobiti popis od K gradova, $2 \leq K \leq 8$, koje moraju obići jašući na devi samo. Gradovi su označeni oznakama od a_1 do a_K , a deva na kojoj jašu **dnevno** može prijeći **najviše L cesta**, $1 \leq L \leq 10$.

Oni moraju putovanje **početi** u gradu a_1 i moraju u **točno** K dana posjetiti i prenoćiti u **svih** $K-1$ preostalih gradova **u bilo kojem redoslijedu** i na kraju se **vratiti** u grad a_1 . Pri tome svaki od gradova a_2 do a_K moraju posjetiti točno jednom, dok grad a_1 moraju posjetiti točno dvaput (na početku i na kraju puta).

Ali Abdul i Yasser nemaju nikakvu kartu pustinje pa ne znaju koji su gradovi povezani cestama, a koji nisu. Međutim, slanjem SMS poruke sa svog turbo mobitela oni mogu saznati **sve susjedne gradove** nekog grada. Kako su naši junaci u bijegu, financijska sredstva su im ograničena i imaju novaca za **najviše 5000 SMS poruka**.

Napišite program koji će im pomoći i pronaći neki put koji će zadovoljavati gore navedena pravila.

biblioteka

Za rješavanje zadatka na raspolaganju je biblioteka funkcija **devalib** koja sadrži četiri funkcije:

Init - ovu funkciju je potrebno pozvati **točno jednom** i to **na početku** vašeg programa. Ova funkcija vraća vrijednosti brojeva N i L .

```
procedure Init(var n,l : longint);  
void Init(int *n,int *l);
```

Gradovi - ova funkcija vraća broj i popis gradova koje trebamo obići, redom od a_1 do a_K .

```
procedure Gradovi(var k : longint;var niz: array of longint);  
void Gradovi(int *k,int niz[]);
```

Susjedi - ova funkcija simulira slanje jedne SMS poruke. Ona vraća broj i popis gradova koji su susjedni gradu kojeg smo naveli u varijabli grad. Ovu funkciju je dozvoljeno pozvati **najviše 5000** puta.

```
procedure Susjedi(grad : longint;var broj : longint;var susjed : array of  
longint);  
void Susjedi(int grad,int *broj,int susjed[]);
```

Rjesenje - ovu funkciju je potrebno pozvati **na kraju** vašeg programa; kao argument prosljeđujete gradove redom kojim ste ih obilazili. Prvi i zadnji grad naveden u nizu **mora biti grad a_1** . Ova funkcija **regularno završava** izvođenje vašeg programa.

```
procedure Rjesenje(duljina : longint;var grad : array of longint);  
void Rjesenje(int duljina, int gradovi[]);
```

Upute za natjecatelje u Pascal-u

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

Također, potrebno je preuzeti biblioteku s evaluacijskog sustava.

Upute za natjecatelje u C/C++

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

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

Izvršnu datoteku možete kreirati i kompajliranjem pomoću komandne linije (npr. za C)
`gcc -o deva.exe deva.c devalib.o`.

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

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

Testiranje

Vaš program možete testirati tako da na standardni ulaz unesete podatke u sljedećem formatu:

u prvom retku brojevi N i L

u drugom retku broj K

u trećem retku brojevi $a_1 a_2 \dots a_K$

u svakom od sljedećih N redaka podaci o susjednim gradovima; u i -tom retku se nalaze podaci o susjedima i -tog grada: prvi broj je broj susjeda, a nakon njega slijedi popis susjeda

Test primjer

7 2

4

1 2 4 6

2 2 5

3 1 3 7

2 2 4

2 3 5

4 1 4 6 7

1 5

2 2 5

Primjer moguceg rjesenja: (1 5 6 5 4 3 2 1)

vid

Mali Vid još ne zna ni čitati, a već se zakačio na geometriju!

Otkad mu je njegov privatni učitelj otkrio čarobni svijet još čarobnijih geometrijskih objekata, Vid je zaboravio na svoje najdraže igračke – Barbiku-hašku-tužiteljicu, maminu pilates loptu i tatine tenis reket, i sada po cijele dane i noći igra jednu te istu, samo njemu zanimljivu igru:

U koordinatnom sustavu u ravnini nalazi se određeni broj točaka. Koordinate svih točaka su **cijeli brojevi**, a **niti jedne** dvije točke **nemaju** istu x ili y koordinatu.

Za dvije točke A i B kažemo da su '**supervidljive**' ako se niti jedna druga točka **ne nalazi unutar pravokutnika** sa stranicama paralelnima s koordinatnim osima kojem se točke A i B nalaze u **nasuprotnim** (dijagonalnim) vrhovima.

Na početku igre koordinatni sustav je **prazan**, a u svakom koraku igre dodajemo **jednu po jednu** točku.

Napišite program koji će nakon svake dodane točke odrediti **trenutni broj parova supervidljivih točaka**.

Jedan par točaka čine dvije različite točke pri čemu nije bitno koja je prva, a koja druga navedena točka. Tako npr. parove [(1,3)(5,1)] i [(5,1)(1,3)] smatramo jednakima i ne brojimo ih dva puta.

ulazni podaci

U prvom retku se nalazi prirodni broj N , $2 \leq N \leq 5000$, broj točaka.

U svakom od sljedećih N redaka se nalaze po dva cijela broja X i Y , $0 \leq X, Y \leq 1000000$ (milijun), koordinate točaka redom kojim se dodaju u koordinatni sustav.

izlazni podaci

U N redaka treba ispisati N cijelih brojeva; u i -ti redak treba ispisati traženi broj parova supervidljivih točaka nakon i -tog koraka.

test primjeri

ulaz

3
1 4
3 3
2 5

izlaz

0
1
3

ulaz

4
2 3
3 4
1 2
4 1

izlaz

0
1
2
5

ulaz

7
1 5
2 7
3 8
5 1
6 2
7 3
4 4

izlaz

0
1
2
5
9
13
10