



ZADATAK	KAMPANJA	MJESEC	ŠETNJA	TRAMPOLIN
<b>ulazni podaci</b>	standardni ulaz			
<b>izlazni podaci</b>	standardni izlaz			
<b>vremensko ograničenje</b>	1 sec	5 sec	1 sec	0.5 sec
<b>memorijsko ograničenje</b>	128 MB	256 MB	256 MB	64 MB
<b>broj bodova</b>	<b>100</b>	<b>100</b>	<b>100</b>	<b>100</b>
	<b>400</b>			



Ususret skorim izborima, predsjednik Josip Ivočić planira krenuti na turneju po Hrvatskoj prilikom koje će držati govore u Zagrebu i Splitu. Kako bi pružila odgovarajuću sigurnost, policija tijekom cijele turneje treba nadzirati **sve gradove** kroz koje će predsjednik proći (uključujući Zagreb i Split).

Naravno, policija želi uštediti novce iz proračuna. Zbog toga će isplanirati put predsjednika od Zagreba do Splita i **natrag do Zagreba** tako da treba nadzirati **najmanji mogući** broj gradova.

Ovom prilikom pretpostavimo da se Hrvatska sastoji od **N** gradova označenih brojevima od 1 do **N**, te **M jednosmjernih** cesta, od kojih svaka povezuje neka dva različita grada. Zagreb je označen brojem 1, a Split brojem 2.

Napišite program koji računa minimalni broj gradova koje treba nadzirati tako da postoji put od Zagreba do Splita i natrag koji prolazi samo kroz gradove koje policija nadzire.

### ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi **N** i **M** ( $2 \leq N \leq 100$ ,  $2 \leq M \leq 200$ ), broj gradova i broj jednosmjernih cesta koji ih povezuju.

U sljedećih **M** redaka nalaze se po dva različita prirodna broja **A**, **B** ( $1 \leq A, B \leq N$ ), početna i završna točka koju spaja dotična jednosmjerna cesta. Neće postojati dvije ceste između dva grada koje su istog smjera, ali je moguće da postoje dvije ceste između dva grada različitih smjerova.

### IZLAZNI PODACI

U prvi i jedini redak izlaza ispišite minimalni broj gradova koje treba nadzirati.

### BODOVANJE

U test podacima ukupno vrijednima 20 bodova bit će  $N \leq 20$ .

**Napomena:** test podaci bit će takvi da će rješenje uvijek postojati.

### PRIMJERI TEST PODATAKA

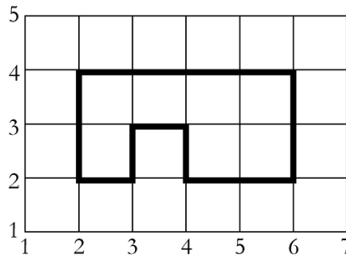
<b>ulaz</b>	<b>ulaz</b>
6 7	9 11
1 3	1 3
3 4	3 4
4 5	4 2
5 1	2 5
4 2	5 3
2 6	3 6
6 3	6 1
<b>izlaz</b>	2 7
6	7 8
	8 9
	9 1
	<b>izlaz</b>
	6

**Objašnjenje prvog primjera:** predsjednik može putovati ovim nizom gradova: 1 -> 3 -> 4 -> 2 -> 6 -> 3 -> 4 -> 5 -> 1. Kako smo kroz sve gradove prošli barem jednom, rješenje je 6.



Potpuno automatiziranu postaju na Mjesecu održava servisni robot M1RK8. Robot se kreće po pruzi koja ima **kružnu putanju**, tj. čini ciklus. Pruga se sastoji od **ravnih dijelova** koji idu u smjeru sjever, jug, istok i zapad, te od oštih **zavoja** pod pravim kutom lijevo i desno. Duljina svakog ravnog dijela pruge prirodan je broj izražen u metrima, tako da se cijela pruga može prikazati u koordinatnom sustavu na način da tračnice prolaze kroz **cjelobrojne koordinate** i **paralelne** su s koordinatnim osima. Pruga **ne siječe** sama sebe i različiti dijelovi pruge nikada se ne preklapaju niti dodiruju.

Primjer pruge sa 8 zavoja:



Nakon pada meteora u blizini postaje, većina senzora na postaji prestala je raditi pa nije poznata ni **pozicija** servisnog robota niti **smjer** po pruzi u kojem je okrenut. Kako bismo započeli s popravcima na postaji korištenjem servisnog robota, prvi je cilj odrediti njegovu poziciju na pruzi.

Taj posao neće biti lagan budući da robot odgovara samo na jednu vrstu naredbe: „**hodaj K**“, pri čemu je **K** cijeli broj metara za koji se robot treba pomaknuti po pruzi. Nakon izvođenja naredbe, robot vraća **dva broja: L i R**. Broj **L** govori koliko je puta **pri izvođenju zadnje naredbe** robot skrenuo **lijevo**, a **R** je broj skretanja **desno**.

Napomene: Početna pozicija robota i pozicija nakon svake naredbe uvijek je točka sa **cjelobrojnim** koordinatama. Ako se robot zaustavi na zavoju, okreće se u smjeru daljnjeg kretanja pruge **prije nego završi** sa izvođenjem naredbe, pa se u idućoj naredbi nastavlja kretati u istom smjeru po pruzi.

Na primjer, ako se robot nalazi na koordinatama (3, 3) i okrenut je prema istoku, nakon izvođenja naredbe „hodaj 4“ završit će na poziciji (6, 2) okrenut prema sjeveru te vratiti rezultat „2 1“, što znači da je pri izvođenju naredbe napravio ukupno dva skretanja lijevo i jedno skretanje desno. Nakon toga, naredba „hodaj 1“ pomaknut će robota na (6, 3) i vratiti „0 0“.

Napišite program koji će izvršavanjem **ne više od 5000** naredbi oblika „hodaj **K**“ odrediti poziciju robota. Program treba nakon toga pozvati naredbu „**kraj X Y**“, pri čemu **X** i **Y** predstavljaju **završne koordinate robota** nakon izvršavanja naredbi.

## INTERAKCIJA

Prije interakcije s robotom, potrebno je sa standardnog ulaza učitati sljedeće podatke.

- U prvom retku nalazi se prirodan broj **N** ( $6 \leq N \leq 10\,000$ ), broj zavoja na pruzi koji je također jedan broju ravnih dijelova pruge.
- U sljedećih **N** redaka navedene su koordinate zavoja redom kojim dolaze po pruzi: za svaki redak dva cijela broja **X** i **Y** ( $1 \leq X \leq 100\,000$ ,  $1 \leq Y \leq 100\,000$ ).

Napomene: **Smjer** kojim se kreće robot može biti smjer kojim su navedene koordinate ili suprotan. Pruga će biti takva da je poziciju robota moguće odrediti, tj. test podaci će imati svojstvo da ne postoje dvije različite orijentirane točke na pruzi za koje vrijedi da svaki niz naredbi „hodaj“ daje isti rezultat.



Nakon učitavanja gornjih podataka moguće je zadavati naredbe na standardni izlaz. Nakon svake izdane naredbe „hodaj **K**“ obavezno napravite *flush* izlaza, te učitajte dva broja **L** i **R** sa standardnog ulaza: broj lijevih i desnih skretanja. **K** mora biti između 1 i 1 000 000 000, uključivo.

Kada vaš program odredi koordinate robota, na standardni izlaz treba ispisati „kraj **X Y**“, napraviti *flush* izlaza, te završiti izvođenje.

## BODOVANJE

U test podacima ukupno vrijednima 40 bodova vrijedit će  $N \leq 100$  i  $X, Y \leq 50$ .

U test podacima ukupno vrijednima 60 bodova vrijedit će  $N \leq 100$  i  $X, Y \leq 100\,000$ .

U test podacima ukupno vrijednima 80 bodova vrijedit će  $N \leq 2000$  i  $X, Y \leq 100\,000$ .

## TESTIRANJE

Svoja rješenja možete testirati na dva načina, lokalno ili putem sustava za evaluaciju. Za obje varijante potrebno je prvo stvoriti datoteku koja će sadržavati test primjer kojim želite testirati svoje rješenje.

- U prvi redak datoteke potrebno je zapisati cijeli broj **N**, broj zavoja.
- U sljedećih **N** redaka potrebno je zapisati koordinate zavoja redom kojim dolaze po pruzi: za svaki redak dva cijela broja **X** i **Y**.
- U sljedeći redak potrebno je zapisati dva cijela broja **X0** i **Y0** te riječ **S**. **X0** i **Y0** predstavljaju početne koordinate robota na pruzi, a **S** smjer robota. Riječ **S** treba biti „isti“ ako se robot kreće u smjeru u kojem su navedeni zavoji, ili „suprotni“ ako se robot kreće u suprotnom smjeru. Početne koordinate mogu biti na zavoju ili na ravnom dijelu pruge.

Primjer ulazne datoteke za testiranje koja odgovara ranijem primjeru pruge sa slike:

```
8
2 4
6 4
6 2
4 2
4 3
3 3
3 2
2 2
3 3 suprotni
```

Za testiranje putem sustava za evaluaciju potrebno je prvo *submitati* izvorni kod vašeg rješenja putem stranice SUBMIT, a zatim poslati test podatak putem stranice TEST.

Lokalno testiranje vrši se putem mjesec\_test datoteke koju je moguće dohvatiti putem sustava za evaluaciju. Potrebno je napisati sljedeću naredbu:

```
./mjesec_test ./vase_rjesenje ulazna_datoteka
```

Koji god način testiranja odabrali, na izlaz ćete dobiti informaciju je li vaš program točno riješio vaš test primjer, te informacije o naredbama koje je vaš program zadao, odgovorima koje je vaš program dobio, te pozicijama robota nakon svakog pomaka.



Mirko je svim svojim prijateljima nabavio ulaznice za nadolazeći koncert grupe Zaz. Sada šeće po svom kvartu, sastaje se s prijateljima i dijeli im ulaznice.

Mirkov kvart možemo zamisliti kao koordinatnu ravninu. Mirko šeće po cjelobrojnim točkama te ravnine, a iz svake točke može **jednim korakom** prijeći u neku od njoj susjednih **osam** točaka (gore, dolje, lijevo, desno ili dijagonalno u bilo kojem smjeru).

Svaki Mirkov prijatelj živi u nekoj cjelobrojnoj točki  $(x, y)$ , a Mirko će se s pojedinim prijateljem naći nedaleko od te točke. Preciznije, sastanak će se dogoditi u točki koja je od prijateljevog doma  $(x, y)$  udaljena za **najviše P koraka**, gdje **P** ovisi o tome koliko je dotični prijatelj voljan hodati do mjesta sastanka.

Na kraju svoje šetnje Mirko se prisjetio **kojim redom** se sastao sa svojim prijateljima. Izračunajte **najmanji broj koraka** koji je Mirko morao učiniti tijekom šetnje. Početna i konačna Mirkova pozicija nije vam poznata.

### ULAZNI PODACI

U prvom retku nalazi se prirodan broj  $N$  ( $2 \leq N \leq 200\,000$ ), broj prijatelja s kojima se Mirko sastao.

Svaki od sljedećih  $N$  redaka opisuje jednog Mirkovog prijatelja: u njemu se nalaze tri cijela broja  $x, y$  i  $P$  ( $0 \leq x, y, P \leq 200\,000$ ) čije je značenje opisano u tekstu zadatka. **Prijatelji su dani redoslijedom u kojem im je Mirko podijelio ulaznice.**

### IZLAZNI PODACI

U jedini redak ispišite traženi najmanji mogući broj Mirkovih koraka.

### BODOVANJE

U test podacima ukupno vrijednima 30 bodova, svi brojevi na ulazu (broj  $N$  i brojevi  $x, y$  i  $P$  za svakog prijatelja) bit će manji ili jednaki 200.

U test podacima ukupno vrijednima dodatnih 30 bodova, za svakog Mirkovog prijatelja broj  $P$  bit će manji ili jednak 10.

### PRIMJERI TEST PODATAKA

<b>ulaz</b>	<b>ulaz</b>
3	4
3 10 2	3 3 5
8 4 2	7 11 5
2 5 2	20 8 10
	30 18 3
<b>izlaz</b>	<b>izlaz</b>
4	19

**Objašnjenje 1. primjera:** Mirko je mogao započeti šetnju u točki  $(4, 8)$ , sastati se s prvim prijateljem, prošetati (dva koraka) do točke  $(6, 6)$ , sastati se s drugim prijateljem, prošetati (dva koraka) do točke  $(4, 5)$  i tamo se sastati s trećim prijateljem.



Na svijetu ima mnogo akcijskih superheroja: Batman, Spiderman, Superman, Nepisman i ostali. U toj širokoj lepezi filmskih junaka pojavio se i jedan fini gospodin, Kickass. Danas je odlučio biti wannabe Spiderman pa je odabrao jedan niz visokih nebodera i počeo skakati po njima.

Ovaj niz ima  $N$  nebodera, označenih brojevima od 1 do  $N$  redom s lijeva na desno, a naš se junak na početku nalazi na neboderu označenom brojem  $K$ . Kickass, nažalost, ima prilično ograničene sposobnosti, pa s nekog nebodera može skočiti samo na njemu **susjedni**, lijevo ili desno, i to samo onda kada je susjedni neboder **niži ili jednake visine** od nebodera na kojem se trenutno nalazi. Ipak, da ne bi ispao slabić, Kickass je na nekim neboderima unaprijed dao postaviti **trampoline** te sa takvih nebodera može skočiti na **bilo koji** drugi neboder, bez obzira na njegovu poziciju ili visinu.

Napišite program koji računa **najveći mogući broj različitih nebodera** koji naš junak može posjetiti jednim nizom skokova počevši od nebodera  $K$ . Posjeti li skačući neki neboder više puta, brojimo ga samo jednom. Također, početni neboder  $K$  brojimo čak i ako se ne vrati na njega.

### ULAZNI PODACI

U prvom retku nalaze se prirodni brojevi  $N$  i  $K$  ( $3 \leq N \leq 300\,000$ ,  $1 \leq K \leq N$ ), ukupan broj nebodera i početna pozicija Kickassa.

U drugom retku nalazi se  $N$  prirodnih brojeva manjih od  $10^6$ , visine nebodera s lijeva na desno.

U trećem retku nalazi se niz od  $N$  znakova, a pritom je svaki znak ili točka ili veliko slovo "T". Ako je  $i$ -ti znak veliko slovo "T", onda se na  $i$ -tom neboderu nalazi trampolin.

### IZLAZNI PODACI

U jedini redak ispišite traženi najveći mogući broj različitih posjećenih nebodera.

### PRIMJERI TEST PODATAKA

<b>ulaz</b>	<b>ulaz</b>
6 4	10 1
12 16 16 16 14 14	10 7 3 1 1 9 8 2 4 10
.T....	..T..T....
<b>izlaz</b>	<b>izlaz</b>
5	7