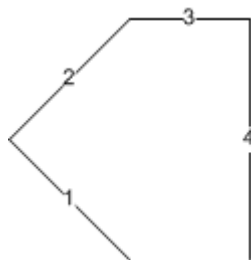


ZADATAK	ŠETNJA	FEJS	ISPIS
izvorni kôd	setnja.pas setnja.c setnja.cpp	fejs.pas fejs.c fejs.cpp	ispis.pas ispis.c ispis.cpp
izvršna datoteka	setnja.exe	fejs.exe	ispis.exe
ulazni podaci	standardni ulaz		
izlazni podaci	standardni izlaz		
vremensko ograničenje	1 sekunda		
broj bodova	20	30	50
	100		

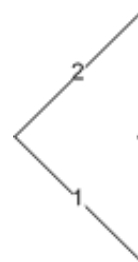
Mario je na poklon dobio prijenosni uređaj za navigaciju. Za razliku od ostalih sličnih uređaja, ovaj ne služi za traženje najkraće ili najbrže rute, već rute za nedjeljnu šetnju kroz prekrasan obližnji park. Uređaj svaki dan predloži novu rutu za šetnju.

Predložena ruta je podijeljena na nekoliko dijelova (etapa), a za svaku etapu su zadana dva broja (X, Y) koja opisuju koliko metara se šetač tijekom te etape treba pomaknuti prema **istoku** (ako je $X > 0$) ili **zapadu** (ako je $X < 0$), te koliko se treba pomaknuti prema **sjeveru** (ako je $Y > 0$) ili **jugu** (ako je $Y < 0$).

Primjerice, neka je predložena ruta koja se sastoji od četiri dijela: 1. etapa (-1, 1), 2. etapa (1, 1), 3. etapa (1, 0) i 4. etapa (0, -2). Za predloženu rutu (slika 1), Mario bi se na kraju šetnje našao 1 metar istočno od početne točke.



Slika 1 Predložena ruta



Slika 2 Ruta bez trećeg dijela

Kako po završetku šetnje ne bio previše udaljen od polazišta, Mario želi **iz predložene rute izbaciti jedan dio puta, tako da završetak rute bude što bliže početnoj točki**. Na primjer, ako preskoči treći dio odabrane rute (slika 2), tada se na kraju šetnje nalazi na početnoj točki (udaljenost je 0).

Napišite program koji će izračunati **završnu poziciju zadane rute**, te odrediti koja je **najkraća moguća udaljenost** od početne točke na kraju šetnje ukoliko se izbaci **tačno jedan dio puta**.

ULAZNI PODACI

U prvom retku ulaza se nalazi broj N ($1 < N \leq 30$), koji predstavlja broj dijelova zadane rute.

U sljedećih N redova se nalaze dva cjelobrojna broja X i Y ($-1000 \leq X, Y \leq 1000$) koji opisuju pojedini dio rute (pomak u smjeru istok-zapad i u smjeru sjever-jug).

IZLAZNI PODACI

U prvi redak izlaza je potrebno ispisati dva broja X i Y koja opisuju završnu poziciju (pomak u smjeru istok-zapad i u smjeru sjever-jug).

U drugi redak je potrebno ispisati jedan broj (**zaokružen na dvije decimale**) koji predstavlja najkraću udaljenost na kraju šetnje ukoliko se izbaci tačno jedan dio puta. Dozvoljena je pogreška u iznosu od 0,01.

Napomena: ukoliko se tačno ispiše opis povratka, a neispravno najkraća udaljenost ili obrnuto, natjecatelj dobiva pola bodova za taj test podatak.

PRIMJERI TEST PODATAKA

ulaz 4 -1 1 1 1 1 0 0 -2	ulaz 3 0 1 1 0 -1 -1	ulaz 4 0 5 5 0 0 -5 -5 0
izlaz 1 0 0.00	izlaz 0 0 1.00	izlaz 0 0 5.00

Objašnjenje prvog primjera: primjer iz zadatka (izbacuje se treći dio rute).

Objašnjenje drugog primjera: izbacivanjem prvog ili drugog dijela udaljenost na kraju rute će biti 1.00, a izbacivanjem trećeg dijela udaljenost bi bila 1.41.

Objašnjenje trećeg primjera: koji god se dio rute izbacuje, kraj rute će uvijek biti udaljen za 5.00 od kuće.

Društvene mreže na Internetu su prestale biti novost i postale su dio svakodnevice. Jedan od zanimljivijih fenomena koji su primijetili članovi društvene mreže Knjigalica je brz porast broja prijateljstava. U skladu s onom starom poslovicom "prijatelj mog prijatelja moj je prijatelj" većina ljudi **jednom dnevno prođe kroz sve svoje prijatelje te svakog njihovog prijatelja dodaje u svoj skup prijatelja**. Potreban je jedan dan da se novostvorena prijateljstva potvrde, pa tako ako su osobe A i B prijatelji, na dan X osoba A vidi samo potvrđena prijateljstva osobe B, dakle ona **nastala prije tog dana**.

Sva prijateljstva su dvosmjerna, ako je A prijatelj B, tada je i B prijatelj A.

Kako društvena mreža Knjigalica još nije uvela mogućnost brisanja prijatelja, u jednom trenutku će **svi biti prijatelji svima**. Napišite program koji će odrediti **nakon koliko dana** će svi korisnici međusobno biti prijatelji. Za svaki dan ispišite **koliko novih prijateljstava** će se stvoriti taj dan.

ULAZNI PODACI

U prvom redu nalaze se prirodni brojevi N i M ($1 \leq N \leq 50$, $1 \leq M \leq N*(N-1)/2$), broj registriranih korisnika i početni broj prijateljstava.

U sljedećih M redaka nalazi se popis početnih prijateljstava. U svakom retku nalaze se dva broja, A i B ($1 \leq A \leq N$, $1 \leq B \leq N$, $A < B$), koji opisuju prijateljstvo između osobe A i osobe B. **Ulazni podaci nikad neće sadržavati isto prijateljstvo više od jednom**. Također, ulazni podaci će biti takvi da će rješenje uvijek postojati.

IZLAZNI PODACI

U prvi red potrebno je ispisati jedan prirodni broj, broj dana nakon kojih će svi korisnici međusobno biti prijatelji.

U svakom od nekoliko sljedećih redaka ispišite po jedan prirodni broj, broj prijateljstava nastalih taj dan.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
3 2	5 4	5 4
1 2	1 2	1 2
2 3	2 3	1 3
	3 4	1 4
	4 5	1 5
izlaz	izlaz	izlaz
1	2	1
1	3	6
	3	

Perica je na poklon dobio antikni iglični pisač, star više od 10 godina, kojeg može kontrolirati pomoću svog računala. Znatizeljno istražujući ponašanje pisača, otkrio je da pisač prepoznaje tri vrste naredbi:

- `POSTAVI (X)` – Ovom naredbom se u memoriju pisača postavlja znak naveden unutar zagrade.
- `ISPISI` – Pisač ispisuje na papir znak koji se nalazi u memoriji.
- `SLJEDECI (X)` – Ovom naredbom se pisaču privremeno zadaje znak koji vrijedi samo za sljedeći ispis, a nakon toga se automatski postavlja znak koji je određen prošlom naredbom `POSTAVI`.

Na primjer, sljedećih 8 naredbi će ispisati niz znakova „AABAA“:

`POSTAVI(A)`, `ISPISI`, `ISPISI`, `POSTAVI(B)`, `ISPISI`, `POSTAVI(A)`, `ISPISI`, `ISPISI`.

Nakon puno razmišljanja, Perica je zaključio da isti niz znakova može ispisati u samo 7 naredbi na sljedeći način:

`POSTAVI(A)`, `ISPISI`, `ISPISI`, `SLJEDECI(B)`, `ISPISI`, `ISPISI`, `ISPISI`.

Od tada, kada god vidi neki niz znakova, Perica razmišlja koji je **najmanji broj naredbi** u kojem njegov pisač može ispisati taj niz. Izradite program koji će pomoći Perici tako da za zadan niz znakova odredi taj minimalan broj naredbi.

Napomena: Prva naredba u nizu uvijek mora biti „`POSTAVI`“.

ULAZNI PODACI

U prvom i jedinom redu nalazi se niz znakova koje pisač treba ispisati. Niz sadrži isključivo velika slova engleske abecede, bez razmaka i interpunkcijskih znakova, a duljina niza nije veća od 10 000 znakova.

IZLAZNI PODACI

U prvi i jedini red treba ispisati broj naredbi koliko je najmanje potrebno za ispis na pisač.

BODOVANJE

U test podacima ukupno vrijednima 50% bodova, duljina zadanog niza biti će manja od 25 znakova.

PRIMJERI TEST PODATAKA

ulaz	ulaz	ulaz
NN	IOIX	BABCBACA
izlaz	izlaz	izlaz
3	7	13

Pojašnjenje prvog primjera: Potrebno je minimalno 3 naredbe: `POSTAVI(N)`, `ISPISI`, `ISPISI`.

Pojašnjenje drugog primjera: Potrebno je minimalno 7 naredbi, na primjer: `POSTAVI(I)`, `ISPISI`, `SLJEDECI(O)`, `ISPISI`, `ISPISI`, `POSTAVI(X)`, `ISPISI`.

Pojašnjenje trećeg primjera: Potrebno je minimalno 13 naredbi, na primjer: `POSTAVI(B)`, `ISPISI`, `SLJEDECI(A)`, `ISPISI`, `ISPISI`, `SLJEDECI(C)`, `ISPISI`, `ISPISI`, `POSTAVI(A)`, `ISPISI`, `SLJEDECI(C)`, `ISPISI`, `ISPISI`.