

## Zadaci

---

Zadatak	JEZIK	MRAVI	TOKI
Izvorni kôd	jezik.pas jezik.c jezik.cpp	mravi.pas mravi.c mravi.cpp	toki.pas toki.c toki.cpp
Ulazna datoteka	jezik.in	mravi.in	toki.in
Izlazna datoteka	jezik.out	mravi.out	toki.out
Vremensko ograničenje (po test podatku)	5 sekundi	5 sekundi	5 sekundi
Broj test podataka	10	10	10
Broj bodova (po test podatku)	5	6	9
Ukupno bodova	<b>50</b>	<b>60</b>	<b>90</b>
	<b>200</b>		

## JEZIK

---

Kod analiziranja kôda u nekom programskom jeziku ponekad je korisno znati postoje li naredbe koje se sigurno nikad neće izvršiti. Naime, ako takve naredbe postoje, onda je to pokazatelj da možda negdje postoji neka greška.

Za ovu priliku promatramo jednostavni programski jezik koji se sastoji od sljedećih naredbi:

RADI                    program se nastavlja izvršavati od sljedeće naredbe

IDI a                    program se nastavlja izvršavati od a-te naredbe

IDI a ILI b            program se nastavlja izvršavati od a-te ili b-te naredbe

Program se uvijek počinje izvršavati od **prve** naredbe.

Napišite program koji će odrediti **broj naredbi koje se nikad neće izvršiti**.

### Ulazni podaci

U svakom retku ulazne datoteke nalazi se jedna naredba, u i-tom retku i-naredba.

U zadnjem retku (iza zadnje naredbe) se nalazi samo znak '!' (točka). Taj redak nije dio programa i ne smatramo ga naredbom.

Broj naredbi će biti manji ili jednak od 10,000.

### Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženi broj naredbi iz teksta zadatka.

### Test primjeri

**jezik.in**

RADI  
RADI  
RADI  
.

**jezik.out**

0

**jezik.in**

IDI 1  
RADI  
.

**jezik.out**

1

**jezik.in**

RADI  
IDI 4 ILI 6  
RADI  
IDI 3  
RADI  
IDI 8  
RADI  
RADI  
.

**jezik.out**

2

# MRAVI

---

Jedno od najcjenjenijih zanimanja u mravljnoj populaciji je prometni policijac. Razlog tome su **uski tuneli** mravinjaka u kojima se mravi koji idu iz suprotnih smjerova **ne mogu mimoići**. Na svu sreću, duž tunela se nalaze **rupe** ('**ugibališta**') u kojima je to moguće izvesti, a **na početku i na kraju** tunela mravi se mogu mimoilaziti i **bez ugibališta**.

Prometniku je poznata duljina tunela kao i lokacije svih rupa. Svakog jutra on dobije popis vremena dolazaka svih mrava na pojedine krajeve tunela i njegov je zadatak organizirati promet u tunelu, tj. omogućiti svim mravima da prođu kroz tunel. To se, naravno, može izvesti na različite brže ili sporije načine, no u onom trenutku kad zadnji mrav prođe kroz tunel naš prijatelj završava s poslom za taj dan pa je jasno da mravi moraju što brže proći kroz tunel.

Brzina svakog mrava je **1 cm/s**. U jedno ugibalište **stane beskonačno** mnogo mrava, a **širina** ugibališta je **zanemariva**. Također, mravi su toliko mali da prometnik njihovu **duljinu** prilikom izračuna **zanemaruje**.

Napišite program koji će izračunati **minimalno vrijeme kada će svi mravi proći kroz tunel**.

## Ulazni podaci

U prvom retku ulazne datoteke nalaze se cijeli brojevi  $D$  i  $U$  međusobno odvojeni jednim razmakom,  $1 \leq D \leq 1,000,000$ ,  $1 \leq U \leq 100,000$ ,  $D > U$ , broj  $D$  je duljina tunela u centimetrima, a broj  $U$  je broj ugibališta. U sljedećih  $U$  redaka nalaze se lokacije ugibališta (u centimetrima, gledajući slijeva na desno) u **strogo rastućem redoslijedu**.

U sljedećem retku se nalazi cijeli broj  $L$ ,  $1 \leq L \leq 100,000$ , broj mrava koji dolaze na lijevi kraj tunela. U sljedećih  $L$  redaka nalaze se vremena dolazaka mrava na lijevi kraj tunela (u sekundama), svaki od tih brojeva je manji ili jednak od 2,000,000.

U sljedećem retku se nalazi cijeli broj  $R$ ,  $1 \leq R \leq 100,000$ , broj mrava koji dolaze na desni kraj tunela. U sljedećih  $R$  redaka nalaze se vremena dolazaka mrava na desni kraj tunela (u sekundama), svaki od tih brojeva je manji ili jednak od 2,000,000.

## Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženo vrijeme (u sekundama) iz teksta zadatka.

## Test primjeri

**mravi.in**

5 1  
2  
1  
3  
1  
2

**mravi.out**

8

**mravi.in**

10 1  
3  
1  
0  
1  
2

**mravi.out**

16

**mravi.in**

10 2  
4  
6  
2  
0  
4  
1  
0

**mravi.out**

14

# TOKI

---

**Podniz** nekog niza znakova  $A$  definiramo kao jedan ili više (**ne nužno uzastopnih**) znakova niza  $A$  pri čemu redosljed znakova ostaje sačuvan.

Zadana su dva niza znakova, niz VOKI i niz TOKI.

Napišite program koji će odrediti **bilo koji najkraći podniz** niza VOKI takav da taj podniz **nije** podniz niza TOKI.

## Ulazni podaci

U prvom retku ulazne datoteke nalazi se niz VOKI, a u drugom retku niz TOKI.

Jedini znakovi koji se mogu pojaviti u tim nizovima biti će mala slova engleske abecede ('a'-'z'), a broj znakova u svakom od nizova će biti manji ili jednak od 2000.

**Napomena:** ulazni podaci će biti takvi da će rješenje, ne nužno jedinstveno, uvijek postojati.

## Izlazni podaci

U prvi redak izlazne datoteke treba ispisati duljinu traženog podniza, dok u drugi redak bilo koji podniz iz teksta zadatka.

## Test primjeri

**toki.in**

ababaa  
abbaa

**toki.out**

3  
bab

**toki.in**

babab  
babba

**toki.out**

3  
aab

**toki.in**

banana  
anbnaanbaan

**toki.out**

5  
banna