

## Zadaci

---

Zadatak	TUNEL	TORANJ	PITA
Izvorni kôd	tunel.pas tunel.c tunel.cpp	toranj.pas toranj.c toranj.cpp	pita.pas pita.c pita.cpp
Ulazna datoteka	tunel.in	toranj.in	–
Izlazna datoteka	tunel.out	toranj.out	–
Vremensko ograničenje (po test podatku)	10 sekundi	10 sekundi	10 sekundi
Broj test podataka	10	10	10
Broj bodova (po test podatku)	3	5	7
Ukupno bodova	<b>30</b>	<b>50</b>	<b>70</b>
	<b>150</b>		

# TUNEL

---

Mirko stoji **na ulazu**, a Slavko **na izlazu** iz tunela. Oni **zapisuju** registracijske oznake automobila redom kojim oni prolaze pokraj njih i te podatke **javljaju** policijskoj patroli koja se nalazi par kilometara dalje.

Na osnovi Mirkovih i Slavkovih podataka policijska patrola može nepogrešivo utvrditi da su neki vozači pretjecali u tunelu, što je naravno strogo zabranjeno.

Napišite program koji će odrediti broj vozača za koje se **sa sigurnošću može tvrditi** da su počinili navedeni prekršaj.

Pretpostavljamo da je promet u tunelu protekao normalno tj. bez sudara, bez zaustavljanja itd.

## Ulazni podaci

Ulazna datoteka se sastoji od  $2N+1$  redaka.

U prvom retku nalazi se cijeli broj  $N$ ,  $1 \leq N \leq 1000$ , broj automobila.

U sljedećih  $N$  redaka nalaze se registracijske oznake tih automobila, redosljedom kojim su **ulazili** u tunel.

U sljedećih  $N$  redaka nalaze se registracijske oznake tih automobila, redosljedom kojim su **izlazili** iz tunela.

Registracijska oznaka nekog automobila sastoji se od najmanje **šest** i najviše **osam** znakova, a jedini dozvoljeni znakovi su velika slova engleske abecede (A-Z) i znamenke dekadskog sustava (0-9).

## Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati broj vozača koje policija sigurno može kazniti zbog nepropisnog pretjecanja.

## Test primjeri

**tunel.in**

```
4
ZG431SN
ZG5080K
ST123D
ZG206A
ZG206A
ZG431SN
ZG5080K
ST123D
```

**tunel.out**

```
1
```

**tunel.in**

```
5
ZG5080K
PU305A
RI604B
ZG206A
ZG232ZF
PU305A
ZG232ZF
ZG206A
ZG5080K
RI604B
```

**tunel.out**

```
3
```

**tunel.in**

```
5
ZG206A
PU234Q
OS945CK
ZG431SN
ZG5962J
ZG5962J
OS945CK
ZG206A
PU234Q
ZG431SN
```

**tunel.out**

```
2
```

# TORANJ

---

Na stolu se nalaze prazne plastične čaše označene brojevima od 1 do  $N$ , koje se mogu staviti jedna u drugu.

Zadan je niz poteza u obliku  $a \rightarrow b$  što znači da **cijelu** hrpu u kojoj se nalazi čaša  $a$  treba premjesiti **na** hrpu u kojoj se nalazi čaša  $b$ . Ako je  $a$  jednako  $b$  ili su  $a$  i  $b$  u istoj hrpi, onda se stanje čaša na stolu **ne mijenja**.

Na primjer, ako imamo 7 čaša i poteze  $1 \rightarrow 3$ ,  $2 \rightarrow 6$ ,  $3 \rightarrow 6$ ,  $4 \rightarrow 7$  i  $4 \rightarrow 2$ , stanje čaša na stolu će se mijenjati na sljedeći način:

```

                                     4
                                     7
                                     1       1       1
                                     3       3       3
                                     2       2 4     2
1 2 3 4 5 6 7 --> 2 3 4 5 6 7 --> 3 4 5 6 7 --> 4 5 6 7 --> 5 6 7 --> 5 6
```

Potezi se moraju izvršavati redoslijedom kojim su navedeni.

Napišite program koji će odabrati **nulti** potez tj. potez koji treba odigrati prije prvog poteza u zadanom nizu tako da na kraju broj čaša u **najbrojnijoj** hrpi bude **što je moguće veći**.

## Ulazni podaci

U prvom retku datoteke nalaze se dva cijela broja  $N$  i  $M$ ,  $2 \leq N \leq 10000$ ,  $0 \leq M \leq 100000$ . Broj  $N$  je broj čaša na stolu, a broj  $M$  broj poteza.

U sljedećih  $M$  redaka navedeni su potezi. U svakom od tih redaka nalaze se po dva cijela broja  $a$  i  $b$  koji označavaju potez  $a \rightarrow b$ .

## Izlazni podaci

U prvi i jedini redak izlazne datoteke treba ispisati traženi nulti potez kao dva cijela broja  $a$  i  $b$  (međusobno odvojena jednim razmakom) koji označavaju potez  $a \rightarrow b$ .

**Napomena:** rješenje ne mora biti jedinstveno.

## Test primjeri

toranj.in

```
5 3
1 2
5 3
4 1
```

toranj.out

```
4 5
```

toranj.in

```
6 4
5 2
3 3
1 3
6 2
```

toranj.out

```
3 5
```

toranj.in

```
10 9
3 7
5 9
10 10
9 5
8 7
3 7
6 1
4 2
2 6
```

toranj.out

```
7 2
```

# PITA

---

Napišite program koji pogađa zamišljeni niz koji se sastoji **samo** od znakova 0 i 1 i **nepoznate** je duljine.

Maksimalna duljina zamišljenog niza bit će **100 znakova**, a vaš program mora pogoditi niz s **najviše 300 poziva** funkcije `Pitaj`.

## Biblioteka (library)

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

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

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

**Pitaj** - funkcija koja se poziva s nizom znakova 0 i 1 (stringom) kao argumentom. Ova funkcija vraća vrijednost **1** ako je taj niz znakova podniz (**nužno uzastopnih elemenata**) nepoznatog niza, a **0** ako nije. Ovu funkciju moguće je pozvati **najviše 300 puta**.

```
function Pitaj(a : string) : integer;  
int Pitaj(const char *a);
```

**Rjesenje** - potrebno ju je pozvati **na kraju** vašeg programa; kao argument prosljeđujete niz znakova 0 i 1 (string) za koji **tvrdite** da je **identičan** nepoznatom nizu. Ova funkcija ne vraća nikakvu vrijednost i **regularno završava izvođenje** vašeg programa.

```
procedure Rjesenje(a : string);  
void Rjesenje(const char *a);
```

Biblioteka `lib` generira datoteku `pita.log`. U njoj su redom popisana sva pitanja koja je postavio vaš program, kao i dobiveni odgovori. Ukoliko postavite više od 300 pitanja, u posljednjem, 301. retku ove datoteke bit će ispisana poruka 'Broj pitanja je veci od 300!'. Ako nakon najviše 300 pitanja pozovete funkciju `Rjesenje`, u posljednjem retku bit će ispisana poruka 'Tocno rjesenje nakon ??? pitanja.' ili 'Pogresan odgovor.'.

## Upute za natjecatelje u Pascal-u

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

Također, potrebno je kopirati datoteke `lib.o` i `lib.ppu` iz direktorija `c:\dmih\pita\lib_pas` u direktorij u kojem se nalazi vaš program.

## Upute za natjecatelje u C-u

Potrebno je kopirati datoteke `lib.h` i `lib.o` iz direktorija `c:\dmih\pita\lib_c` u direktorij u kojem se nalazi vaš program.

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

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

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

# PITA

---

## Testiranje

Vaš program možete testirati tako da kreirate datoteku `pita.in` u koju ćete upisati neki niz znakova 0 i 1. Taj niz će onda pri idućem pokretanju vašeg programa 'zamisliti' biblioteka.

**Napomena:** za potrebe evaluacije koristit će se biblioteka bez gore navedene mogućnosti testiranja.

## Primjer

**pita.in**

```
1000101
```

Niz od 5 ispravnih poziva funkcija iz biblioteke bio bi npr.:

**Pascal**

```
Init;  
Pitaj('101');  
Pitaj('11');  
Pitaj('10001001');  
Rjesenje('1000101');
```

**C**

```
Init();  
Pitaj("101");  
Pitaj("11");  
Pitaj("10001001");  
Rjesenje("1000101");
```

U datoteci `pita.log` tada će biti zapisano:

```
Pitanje 1 : 101 ..... DA  
Pitanje 2 : 11 ..... NE  
Pitanje 3 : 10001001 ..... NE  
Rjesenje je 1000101 ..... DA  
Tocno rjesenje nakon 3 pitanja.
```