

## Upute natjecateljima

---

Ovo natjecanje sastoji se od rješavanja **6 problemskih zadataka** u vremenu od **3 sata** (180 minuta). Zadatke je potrebno rješavati u jednom od slijedećih programskih jezika: Pascal, C ili C++. Natjecatelji u Pascalu moraju programirati u programskom alatu **FreePascal**, a natjecatelji u C-u i C++-u programskom alatu **DJGPP**. Dozvoljeno je koristiti editor po izboru te pomoću navedenih alata prevoditi izvorni kod u izvršnu datoteku.

Za početak, provjerite da ste dobili od vašeg voditelja **točno 9 stranica** (uključujući ovu koju trenutno čitate). Morate imati dvije stranice s uputama (ovu i još jednu), jednu uvodnu stranicu s tablicom zadataka, i još 6 stranica s tekstovima zadataka.

Tijekom natjecanja **ne smijete komunicirati** ni sa jednom osobom, osim voditelja natjecanja. To znači da morate **raditi samostalno i ne smijete koristiti Internet**. Također, zabranjena je upotreba bilo kakvih ranije napisanih programa ili dijelova programa.

Po isteku vremena predviđenog za natjecanje, morate voditelju natjecanja predati disketu sa snimljenim izvornim kôdovima rješenja. Nakon natjecanja, sustav za evaluaciju će testirati vaša rješenja na, unaprijed zadanim, test podacima i dodijeliti vam određeni broj bodova. Na kraju svakog zadatka dani su primjeri test podataka. Ti primjeri služe da bi vam tekst zadataka bio što je moguće jasniji te za provjeru formata ulaza i izlaza, a ne služe za provjeru ispravnosti vašeg programa. Ako vaš program radi na tim primjerima, to **nije garancija** da će raditi za ijedan službeni test podatak.

Pripazite na to da ne nosi svaki zadatak jednak broj bodova. Lakše i brže rješivi zadaci nose manje bodova, dok teži nose više bodova. Svaki test podatak u nekom zadatku nosi jednak broj bodova. Ukupni broj bodova na nekom zadatku je jednak zbroju bodova test podataka koji se poklapaju sa službenim rješenjem. Ukupan broj bodova jednak je zbroju bodova na svim zadacima.

Primjetite da **svi zadaci imaju ulaz sa standardnog ulaza (tipkovnice) i izlaz na standardni izlaz (monitor)**. Sve informacije o zadacima (ime zadatka, ulaz, izlaz, vremensko i memorijsko ograničenje, način bodovanja) možete naći na uvodnoj stranici s naslovom *Zadaci*. Ako vam nije jasno nešto u vezi načina provedbe ovog natjecanja, odmah postavite pitanje voditelju da vam to razjasni.

Tijekom cijelog natjecanja možete postavljati pitanja voditelju natjecanja u vezi zadataka. Dozvoljena su pitanja **koja razjašnjavaju nejasnoće u tekstu zadatka**. Ne smijete postavljati pitanja u vezi rješavanja zadataka. Prije nego postavite pitanje, pročitajte još jednom zadatak, jer je moguće da ste u prethodnom čitanju preskočili dio teksta zadatka.

## Upute natjecateljima

---

### VAŽNO za C/C++!

Glavni program (glavna funkcija) **mora** biti deklariran kao: `int main(void) { ... }`.

Program mora završiti svoje izvođenje naredbom `return 0;` unutar funkcije `main` ili naredbom `exit(0);`.

**Zabranjeno** je koristiti biblioteke `<conio.h>` i `<cconio>`, kao i sve funkcije deklarirane u ovim bibliotekama (npr. `clrscr()`; `getch()`; `getche()`; i sl.). Zabranjeno je koristiti i **sve** sistemske (nestandardne) biblioteke.

**Zabranjeno** je koristiti funkcije `itoa()` i `ltoa()` jer one ne postoje u standardu jezika C/C++. Umjesto tih funkcija možete koristiti funkciju `sprintf()`, koja ima i veće mogućnosti primjene, deklariranu u `<stdio.h>` i `<cstdio>`.

**Zabranjeno** je koristiti sve funkcije za rad s datotekama (`fopen()`, `fclose()`, `fprintf()`, `fscanf()`, ...), kao i deklarirati varijable tipa `FILE *`.

**Dozvoljeno** je koristiti sve ostale standardne biblioteke (koje su dio jezika), uključujući i STL (Standard Template Library) u jeziku C++.

### VAŽNO za Pascal!

Program **mora** regularno završiti svoje izvođenje naredbom `end.` unutar glavnog programa ili naredbom `halt;`.

**Zabranjeno** je koristiti bilo kakve biblioteke, a posebno biblioteku `crt`, tj. zabranjeno je u programu imati direktivu `uses`. To znači da u programu ne smije biti naredbi `clrscr()` i `readkey()`.

**Nepoštivanje ovih uputa ili nepridržavanjem formata izlaznih podataka rezultirat će nepovratnim gubitkom bodova. Nemojte ispisivati ništa što se u zadatku ne traži**, kao npr. poruke tipa 'Rjesenje je:' ili 'Unesite brojeve' i slično!

**Napomena:** ista ovakva ograničenja će biti i na ovogodišnjem Državnom natjecanju.

Nakon natjecanja, sva eventualna pitanja, komentare i mišljenja o zadacima uputite na donju e-mail adresu.

Sretno i uspješno natjecanje!

Autor zadataka, test podataka i rješenja

Luka Donđivić, ldondjivic@yahoo.com

## Zadaci

Zadatak	R2	ABC	KOLONE	SJECIŠTA	STOL	STRAŽA
Izvorni kôd	r2.bas r2.pas r2.c r2.cpp	abc.bas abc.pas abc.c abc.cpp	kolone.bas kolone.pas kolone.c kolone.cpp	sjecista.bas sjecista.pas sjecista.c sjecista.cpp	stol.bas stol.pas stol.c stol.cpp	straza.bas straza.pas straza.c straza.cpp
Ulazna datoteka	standardni ulaz ( <b>tipkovnica</b> )					
Izlazna datoteka	standardni izlaz ( <b>zaslon</b> )					
Memorijsko ograničenje (heap)	32 MB	32 MB	32 MB	32 MB	32 MB	32 MB
Memorijsko ograničenje (stack)	8 MB	8 MB	8 MB	8 MB	8 MB	8 MB
Vremensko ograničenje (po test podatku)	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda
Broj test podataka	10	10	10	10	10	10
Broj bodova (po test podatku)	1	2	4	4	8	11
Ukupno bodova	<b>10</b>	<b>20</b>	<b>40</b>	<b>40</b>	<b>80</b>	<b>110</b>
	<b>300</b>					

**Napomena:** Vremensko ograničenje je mjereno na računalu baziranom na dva procesora AMD Athlon MP 2600+ i operativnim sustavom Linux. Program u C-u i C++-u treba kompajlirati sa sljedećim opcijama: `-O2 -lm -static`, a program u Pascalu sa `-O1 -XS`. U programskom jeziku QBasicu dozvoljeno vrijeme izvršavanja programa je dvostruko veće.

## 1. R2

---

Za broj  $S$  kažemo da je aritmetička sredina brojeva  $R1$  i  $R2$  ako vrijedi  $S=(R1+R2)/2$ . Mirko je Slavku poklonio cijele brojeve  $R1$  i  $R2$ . Slavko je izračunao njihovu aritmetičku sredinu  $S$  koja je također bila cijeli broj, ali mu je nakon toga netko ukrao  $R2$ ! Pomozite Slavku da vrati  $R2$ !

### Ulazni podaci

U prvom i jedinom redu nalaze se dva cijela broja  $R1$  i  $S$ , oba između  $-1000$  i  $1000$ .

### Izlazni podaci

U prvi i jedini red ispišite  $R2$ .

### Primjeri test podataka

**ulaz**

11 15

**izlaz**

19

**ulaz**

4 3

**izlaz**

2

## 2. ABC

---

Zadana su tri prirodna broja A, B i C. Brojevi nisu dani u tom redoslijedu, ali je poznato je da je A manji od B, a B manji od C.

Kako bi ljepše izgledali, potrebno ih je presložiti u neki redoslijed, a upravo to je vaš zadatak.

### Ulazni podaci

U prvom redu nalaze se tri prirodna broja A, B i C, ne nužno u tom redoslijedu. Sva tri broja će biti manja od ili jednaka 100.

U drugom redu nalaze se tri velika slova 'A', 'B' i 'C' (bez razmaka) koja predstavljaju traženi redoslijed ispisivanja.

### Izlazni podaci

Ispišite brojeve A, B i C zadanim redoslijedom u jedan red, odvojene s po jednim razmakom.

### Primjeri test podataka

**ulaz**

1 5 3

ABC

**izlaz**

1 3 5

**ulaz**

6 4 2

CAB

**izlaz**

6 2 4

### 3. KOLONE

---

Poznato je da se mravi uvijek kreću u koloni. Međutim, manje je poznato što se događa kada se dvije kolone mrava sretnu u prolazu koji je preuzak da bi se dva mrava mogla mimoći. Jedna teorija kaže da u tom slučaju mravi preskaču jedni druge.

Od trenutka kada se kolone sretnu, svake sekunde svaki mrav preskoči (ili biva preskočen, kako se dogovore) mrava ispred sebe tako da ta dva mrava zamijene mjesta, ali samo ako se drugi mrav kreće u suprotnom smjeru.

Potrebno je odrediti redoslijed mrava u prolazu nakon  $T$  sekundi.

#### Ulazni podaci

U prvom redu nalaze se prirodani brojevi  $N_1$  i  $N_2$ , broj mrava u prvoj i drugoj koloni.

U sljedeća dva reda nalaze se redoslijedi mrava u prvoj i drugoj koloni (od prvog prema posljednjem). Svaki mrav je jedinstveno označen velikim slovom engleske abecede (među svim mravima ne postoje dva s istom oznakom).

U zadnjem redu nalazi se cijeli broj  $T$  ( $0 \leq T \leq 50$ ), broj sekundi proteklih od susreta kolona.

#### Izlazni podaci

Ispišite redoslijed mrava nakon  $T$  sekundi. Naš kut gledanja je takav da nam prva kolona nailazi s lijeve, a druga kolona s desne strane.

#### Primjeri test podataka

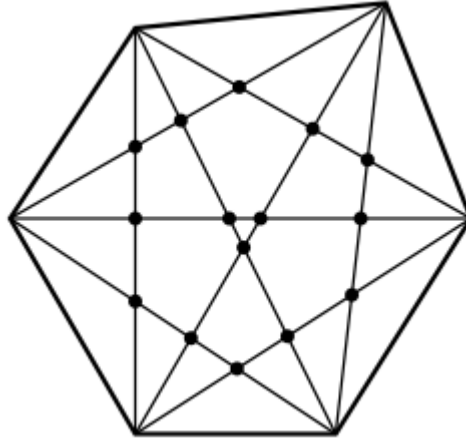
<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
3 3	3 3	3 4
ABC	ABC	JLA
DEF	DEF	CRUO
0	2	3
<b>izlaz</b>	<b>izlaz</b>	<b>izlaz</b>
CBADEF	CDBEAF	CARLUJO

## 4. SJECIŠTA

---

Promotrite konveksni poligon s  $N$  vrhova, sa svojstvom da mu se nijedne tri dijagonale ne sijeku u istoj točki. Odredite koliko taj poligon ima sjecišta dijagonala.

Na slici je prikazan jedan takav poligon sa 6 vrhova.



**Napomena:** poligon je konveksan ako su mu svi unutarnji kutevi manji od 180 stupnjeva.

### Ulazni podaci

U prvom i jedinom redu nalazi se prirodni broj  $N$ ,  $3 \leq N \leq 100$ .

### Izlazni podaci

U prvi i jedini red ispišite broj sjecišta dijagonala poligona.

### Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
3	4	6
<b>izlaz</b>	<b>izlaz</b>	<b>izlaz</b>
0	1	15

## 5. STOL

---

Mirko je kupio stan i želi pozvati što više ljudi na večeru da proslavi. Da bi to bilo moguće, potreban mu je veliki drveni stol pravokutnog oblika za koji bi mogli sjesti svi gosti i on sâm. Broj ljudi koji može sjesti za stol jednak je opsegu stola. Mirko je odlučio kupiti onaj stol koji stane u njegov stan, a da za njega može sjesti što više ljudi.

Zadan je tlocrt stana. Mirko stol mora postaviti tako da mu rubovi budu paralelni s rubovima tlocrta. Odredite koliko ljudi Mirko može pozvati na večeru.

### Ulazni podaci

U prvom redu nalaze se dva prirodna broja  $R$  i  $S$  ( $1 \leq R, S \leq 400$ ) dimenzije stana.

U sljedećih  $R$  redova nalazi se po  $S$  znakova (bez razmaka) koji označavaju nalazi li se zid na nekom kvadratu Mirkovog stana (znak 'X') ili ne (znak '.').

Stol može zauzimati samo one kvadrate na kojima se ne nalazi zid.

### Izlazni podaci

U prvi i jedini red ispišite broj ljudi koje Mirko može pozvati na večeru.

### Primjeri test podataka

<b>ulaz</b>	<b>ulaz</b>	<b>ulaz</b>
2 2	4 4	3 3
..	X.XX	X.X
..	X..X	.X.
<b>izlaz</b>	..X.	X.X
	..XX	<b>izlaz</b>
7	<b>izlaz</b>	3
	9	



## 6. STRAŽA

---

Ispred vojne baze nalazi se sustav rovova, koji promatramo u tlocrtu (u ravnini). Svaki rov možemo predstaviti dužinom. Po noći, dok vojnici spavaju, tri stražara čuvaju rovove. Kažemo da se stražari vide ako postoji rov (ili niz rovova) duž cijele dužine između njih i na toj dužini se ne nalazi treći stražar.

Iz sigurnosnih razloga stražari se moraju rasporediti tako da svaki stražar vidi ostalu dvojicu. Na koliko se načina mogu rasporediti?

### Ulazni podaci

U prvom redu nalazi se prirodni broj  $N$  ( $1 \leq N \leq 20$ ), broj rovova. U sljedećih  $N$  redaka nalaze se podaci o svakom rovu: četiri prirodna broja  $X1, Y1, X2, Y2$  (svi manji od ili jednaki 1000), gdje  $X1$  i  $Y1$  predstavljaju koordinate jednog, a  $X2$  i  $Y2$  koordinate drugog kraja rova.

Rovovi u ulazu se mogu preklapati ili dodirivati krajevima (vidi drugi primjer).

### Izlazni podaci

U prvi i jedini red ispišite broj načina na koji se stražari mogu rasporediti.

### Primjeri test podataka

**ulaz**

```
3
2 2 3 2
3 2 3 3
3 3 2 3
```

**izlaz**

```
0
```

**ulaz**

```
4
5 1 7 1
1 1 5 1
4 0 4 4
7 0 3 4
```

**izlaz**

```
1
```

**ulaz**

```
6
0 0 1 0
0 0 0 1
1 0 1 1
0 1 1 1
0 0 1 1
1 0 0 1
```

**izlaz**

```
8
```