

Upute za natjecatelje

Sastavni dio kompleta zadataka je i **uvodna stranica** na kojoj se nalaze mnogi **bitni podaci** o zadacima i molimo vas da je **vrlo pažljivo** pročitate i da prilikom rješavanja pošujete **sve** navedene specifikacije.

Prije isteka vremena predviđenog za rješavanje zadataka morate **sva vaša rješenja poslati Sustavu za evaluaciju**. Za bodovanje vaših rješenja u obzir se uzimaju **samo** one datoteke koje su pravovremeno i na ispravan način poslane Sustavu za evaluaciju.

Dobro pročitate poglavlja **ULAZNI PODACI** i **IZLAZNI PODACI** kod svakog pojedinog zadatka. Na tom mjestu piše sve o formatu ulaznih i izlaznih podataka koji mora biti **strogo poštovan** kako bi vaša rješenja bila točno evaluirana. Za ilustraciju i bolje razumijevanje pogledajte poglavlje **TEST PRIMJERI** u kojem će se nalaziti nekoliko jednostavnijih primjera ulaznih podataka i njima odgovarajućih izlaznih podataka.

Vaši programi **ne čitaju** ulazne podatke iz ulaznih datoteka i **ne zapisuju** izlazne podatke u izlazne datoteke. Sav ulaz i izlaz treba se odvijati preko **standardnog ulaza** (stdin) i **standardnog izlaza** (stdout). Vaš program sa standardnog ulaza mora očekivati **samo** ulazne podatke, a na standardni izlaz mora ispisivati **samo** izlazne podatke. Ako će vaš program čekati na unos nečeg drugog osim ulaznih podataka i ispisivati nešto drugo (npr. "Unesite brojeve...", "Rjesenje je..." i slično) osim izlaznih podataka, neće dobiti bodove za taj zadatak.

Ako je za rješavanje nekog zadatka na raspolaganju **biblioteka** s funkcijama, **nemojte ništa niti učitavati sa standardnog ulaza niti ispisivati na standardni izlaz** jer u tom slučaju nećete dobiti bodove za taj zadatak.

Također, vaši programi **ne smiju pristupati** nikakvim datotekama **niti ih kreirati**. Kršenje ovog pravila rezultirat će gubitkom bodova za taj zadatak.

Program kod kojeg se prilikom izvršavanja pojavi bilo kakav 'runtime error', **neće dobiti bodove** za taj test podatak.

Obratite pažnju da osim vremenskog ograničenja postoji i **memorijsko ograničenje**. Ako će vaš program prilikom izvršavanja koristiti više memorije nego što je dozvoljeno, neće dobiti bodove za taj zadatak.

Ne trebate kreirati izvršnu (exe) datoteku. Sustav za evaluaciju će iz vašeg izvornog kôda kreirati izvršnu datoteku na sljedeći način (pretpostavimo da se zadatak zove *liftovi*):

pascal

```
fpc -O1 -XS liftovi.pas
```

c

```
gcc -o liftovi.exe liftovi.c -std=c99 -O2 -s -static -lm
```

c++

```
g++ -o liftovi.exe liftovi.cpp -O2 -s -static -lm
```

(ukoliko koristite alat DJGPP, umjesto "g++" u gornjoj naredbi napišite "gpp")

Upute za natjecatelje

Slijede upute kako da vaše rješenje na Sustavu bude prihvaćeno (da se uspješno kompajlira) te da radi prema očekivanjima:

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 npr. ne smije biti naredbi `clrscr()` i `readkey()`.

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 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>`.

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

Za dodjelu bodova važan je samo točan ispis rezultata. Prilikom evaluacije nitko neće gledati vaš izvorni kôd već će on samo biti korišten za izradu izvršne datoteke, a bodove za pojedini test podatak će dobiti samo oni programi koji će generirati **točan rezultat** i **regularno završiti** svoje izvođenje **unutar predviđenog vremenskog i memorijskog ograničenja**.

Pogledajmo primjer jednog pravilno napisanog programa:

Zadatak: Napišite program koji će zbrojiti i oduzeti dva cijela broja.

Ulaz: U prvom retku se nalaze dva cijela broja A i B, međusobno odvojena jednim razmakom.

Izlaz: U prvi redak ispišite zbroj, a u drugi redak razliku brojeva A i B.

pascal

```
program p(input,output);
var
  a,b : integer;
begin
  readln(a,b);
  writeln(a+b);
  writeln(a-b);
end.
```

c

```
#include <stdio.h>
int main(void)
{
  int a,b;
  scanf("%d%d",&a,&b);
  printf("%d\n",a+b);
  printf("%d\n",a-b);
  return 0;
}
```

c++

```
#include <iostream>
using namespace std;
int main(void)
{
  int a,b;
  cin >> a >> b;
  cout << a+b << endl;
  cout << a-b << endl;
  return 0;
}
```

Upute za natjecatelje

Prilikom rješavanja nekog zadatka i testiranja njegovog rješenja preporučuje se korištenje **operatora redirekcije ulaza** kako ne biste više puta nepotrebno unosili podatke preko tipkovnice. Npr., od ulaznih podataka za neki od test primjera iz zadatka možete napraviti tekstualnu datoteku i testirati vaš program tako da ga pokrećete na sljedeći način (pretpostavimo da se zadatak zove *liftovi*):

```
liftovi <liftovi.txt
```

Znak < je operator redirekcije ulaza i sve što se nalazi u datoteci *liftovi.txt* bit će prosljeđeno vašem programu na isti način kao da ste sadržaj te datoteke unijeli preko tipkovnice.

Težina zadataka je takva da većina natjecatelja neće uspjeti riješiti sve zadatke, tako da ako i ne riješite neki zadatak, nemojte biti obeshrabreni nego rješavajte ostale zadatke.

Da bi program koji rješava problem iz nekog zadatka dobio maksimalni broj bodova, primijenjeni algoritam mora biti valjan i efikasan tj. brz. Test podaci su unaprijed osmišljeni i koncipirani na način da će programi koji koriste neke manje efikasne, ali valjane algoritme, također dobiti određeni broj bodova (npr. od ukupno 60 bodova, jako loš i spor algoritam će dobiti npr. 20 bodova, dok će dobar algoritam, ali ne i najbolji dobiti npr. 40 bodova). Programi koji će raditi jako brzo za sve test podatke, ali neće davati točne rezultate, naravno, neće donositi bodove. Znači, valjanost algoritma je na prvom mjestu, a brzina izvršavanja na drugom.

Prilikom rješavanja zadataka obratite pažnju na **ograničenja** kod ulaznih podataka jer su ona **izrazito bitna** za osmišljavanje efikasnog algoritma.

Kod svakog zadatka naći ćete nekoliko test primjera koji vam mogu pomoći da bolje shvatite zadatak. Međutim, ako vaš program radi ispravno za zadane test primjere to **još uvijek nije garancija** niti da vaš program radi korektno za ostale test podatke niti da je dovoljno brz.

Za vrijeme natjecanja **nije dozvoljeno** korištenje pisanih materijala, medija za pohranu podataka i elektroničkih uređaja kao ni komunikacija s drugim natjecateljima. Jedino što je dozvoljeno je korištenje **olovke i papira**, koje ujedno i preporučujemo za **skiciranje i razradu algoritama**, a za vrijeme natjecanja često snimajte vaša rješenja na hard disk i koristite opciju "Backup" na Sustavu za evaluaciju.

Cijelo vrijeme tijekom natjecanja možete postavljati pitanja u vezi zadataka isključivo koristeći dobiveni obrazac.

Želimo vam puno znanja, koncentracije i sreće na ovom i daljnjim natjecanjima.

Povjerenstvo za izradu zadataka i provedbu natjecanja