
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. Na ostalim stranicama se nalaze **četiri** zadatka.

Nakon isteka **tri sata** od trenutka kada ste počeli s rješavanjem morate prekinuti rad na računalu prethodno snimivši vaša rješenja na hard disk vašeg računala i na dobivenu disketu. Prilikom evaluacije vaših rješenja u obzir će se uzimati **samo** one datoteke koje nose imena koja su navedena na uvodnoj stranici.

Dobro pročitate poglavlja **ULAZNI PODACI** i **IZLAZNI PODACI** kod svakog pojedinog zadatka. Tamo 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 **PRIMJERI TEST PODATAKA** 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.

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

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")
```

Vaš program treba **regularno završiti** svoje izvođenje.

Program u Pascal-u treba se izvršiti do kraja tj. do 'end.' ili ako za završetak izvršavanja koristite naredbu 'halt', treba je koristiti bez parametara ili kao 'halt(0)'.
Program u C/C++-u treba se izvršiti do kraja tj. do 'return 0;' u funkciji 'main' koja treba biti deklarirana kao 'int main(void)' ili ako za završetak izvršavanja koristite naredbu 'exit', treba je koristiti kao 'exit(0)'.

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 teksta 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 je bilo unijeto preko tipkovnice.

Upute za natjecatelje

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;
}
```

Sustav za evaluaciju nalazi se na računalu s **LINUX** operativnim sustavom. Stoga, da bi se vaš program uspješno evaluirao na Sustavu, **ne smijete** koristiti naredbe i funkcije specifične za **DOS**. Tako npr. u Pascalu ne smijete koristiti **CRT** modul, a u C/C++-u **conio.h** i slične module.

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**.

Obratite pažnju da ne nose svi zadaci jednak broj bodova. Lakši i brže rješivi zadaci nose manje bodova, a teži zadaci za čije je rješavanje potrebno više vremena, znanja i koncentracije nose više bodova. 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 preporučuje se korištenje olovke i papira za skiciranje i razradu algoritma.

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

Povjerenstvo za izradu zadataka i provedbu natjecanja