

Sastavni dio kompleta zadataka su i ove upute te uvodna stranica na kojoj se nalaze važni podaci o zadatcima. Molimo vas da i jedno i drugo pažljivo pročitate. Na ostalim stranicama nalaze se tri zadatka. Prilikom rješavanja zadataka preporučuje se korištenje olovke i papira za skiciranje i razradu algoritma.

Nakon završetka natjecanja, članovi županijskih povjerenstava poslat će izvorne kodove vaših rješenja na automatsku evaluaciju. Nakon završetka evaluacije za vašu županiju, rezultati evaluacije bit će dostupni putem sustava za evaluaciju na adresi <https://srednje.hsin.hr>.

Žalbe na rezultate evaluacije možete uputiti izravno Županijskom povjerenstvu (koje će vas obavijestiti o roku i načinu predaje žalbi) ili Državnom povjerenstvu na adresu dp-informatika@azoo.hr. Rok za predaju žalbi Državnom povjerenstvu istječe *u ponoć na dan natjecanja*. Rok žalbi Državno povjerenstvo može dodatno produžiti u slučaju većih problema sa zadatcima ili sustavom za evaluaciju.

Ulazni i izlazni podatci

Kod svakog pojedinog zadatka obratite pozornost na sekcije *Ulazni podatci* i *Izlazni podatci*. Tu su definirana pravila vezana uz format ulaznih i izlaznih podataka koji mora biti strogo poštovan kako bi vaša rješenja bila ispravno evaluirana. Vaš program sa standardnog ulaza mora očekivati samo zadane ulazne podatke, a na standardni izlaz ispisivati samo tražene izlazne podatke bez ikakvih dodatnih poruka. Ako vaš program bude čekao na unos nečeg drugog osim ulaznih podataka ili ispisivao nešto drugo osim izlaznih podataka (npr. *Unesite brojeve...*, *Rješenje je...* i slično), nećete dobiti bodove jer evaluator to ne očekuje.

Vaši programi ne smiju pristupati datotekama ili ih kreirati te ne smiju pokretati nove procese ili koristiti više od jedne dretve (threada).

Prilikom rješavanja nekog zadatka i testiranja njegovog rješenja preporučuje se korištenje operatora redirekcije ulaza kako ne biste više puta nepotrebno unosili podatke preko tipkovnice. Na primjer, ulazne podatke za neki od probnih primjera iz teksta zadatka možete spremirati u tekstualnu datoteku i testirati vaš program tako da ga pokrećete iz komandne linije na sljedeći način (pretpostavimo da se zadatak zove „Neboder“):

```
neboder < primjer.txt
```

Znak < je operator redirekcije ulaza i sve što se nalazi u datoteci `primjer.txt` bit će prosljeđeno vašem programu kao da je uneseno preko tipkovnice.

U slučaju Pythona potrebno je eksplicitno pozvati odgovarajući prevoditelj. Na primjer:

```
C:\Python35\python neboder.py < primjer.txt
```

Bodovanje

Da bi program koji rješava neki zadatak dobio maksimalan broj bodova, primijenjeni algoritam mora biti najprije točan, ali i efikasan tj. brz. Test podatci unaprijed su osmišljeni tako da će programi koji koriste manje efikasne, ali valjane algoritme, također dobiti određeni broj bodova (npr. od ukupno 60 bodova, vrlo spor algoritam dobit će npr. 20 bodova, dok će dobar algoritam, ali ne i najbolji, dobiti npr. 40 bodova). Jako brz program koji ne daje točne rezultate, naravno, ne donosi bodove. Valjanost algoritma je na prvom mjestu, a brzina izvršavanja na drugom.

Različite zadatke možete rješavati u različitim jezicima, imajući na umu da će rješenja u Pythonu, zbog prirode jezika, biti sporija od ekvivalentnih rješenja u drugim dozvoljenim jezicima i da bi zato mogla na nekim testnim primjerima premašiti vremensko ograničenje.

U (nekim) zadatcima **obratite pažnju na odjeljak *Bodovanje***. On opisuje testne primjere (te pripadni broj bodova) koji su potencijalno lakše rješivi jer sadrže manje brojeve ili su na neki način specifični.

Evaluacija

Bodove za pojedini test podatak dobit će samo oni programi koji budu generirali točan rezultat unutar navedenog vremenskog i memorijskog ograničenja, te regularno završe svoje izvođenje. Točnije, program se treba izvršiti do kraja. U programskim jezicima C/C++ to znači do `return 0;` na kraju funkcije `main` koja treba biti deklarirana kao `int main()`, ili do naredbe `exit(0);`.

Ne trebate predati izvršnu (`exe`) datoteku već samo predajete datoteku s izvornim kodom. Imena datoteka moraju odgovarati imenima zadataka, a ekstenzija datoteke mora odgovarati programskom jeziku i standardu u skladu s donjom tablicom (`.c`, `.cpp`, `.cxx` ili `.py`). Npr. ako se zadatak zove „Neboder“ i ako koristite C++11, predat ćete datoteku `neboder.cxx`. Na temelju odgovarajuće ekstenzije, za jezike C/C++ sustav za evaluaciju iz vašeg će izvornog kôda kreirati izvršnu datoteku na sljedeći način:

```
C          gcc -DEVAL -O2 -o neboder neboder.c -lm
C++       g++ -DEVAL -O2 -o neboder neboder.cpp
C++11    g++ -DEVAL -std=c++11 -O2 -o neboder neboder.cxx
```

Za rješenja u Pythonu 3, prva linija koda treba biti:

```
#!/usr/bin/python3
```

Za evaluaciju će se koristiti verzija 3.5, a prije izvođenja vaš kod bit će preveden u bytecode naredbom `python -m py_compile neboder.py`.

Računalo na kojem se vrši evaluacija je Linux računalo s Ubuntu 16.04 LTS 64-bitnim operativnim sustavom i sljedećim verzijama prevoditelja: `gcc 4.9`, `g++ 4.9`. Preporučujemo da svoja rješenja obavezno isprobate sa `gcc` ili `g++` prevoditeljima s gore navedenim opcijama, osobito ako koristite neki drugi alat (npr. Microsoft Visual Studio) tijekom natjecanja. Da bi vaše rješenje bilo uspješno prevedeno, morate koristiti samo standardne biblioteke, tj. ne smijete koristiti naredbe i funkcije specifične za Windows.

Primjeri pravilno napisanih programa

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

Ulaz: U prvom retku nalaze se 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.

C	C++	Python
<pre>#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; }</pre>	<pre>#include <iostream> using namespace std; int main(void) { int a, b; cin >> a >> b; cout << a + b << endl; cout << a - b << endl; return 0; }</pre>	<pre>#!/usr/bin/python3 a, b = map(int, input().split()) print(a + b) print(a - b)</pre>

Česte pogreške

Donosimo popis **čestih i nepotrebnih pogrešaka** na natjecanjima ovog tipa. Sve od sljedećeg može rezultirati time da će evaluator dodijeliti nula bodova vašem rješenju.

- **Manjak inicijalizacije lokalnih varijabli:** U jezicima C i C++ lokalne varijable (što uključuje i varijable deklarirane unutar funkcije `main`) ne inicijaliziraju se automatski te njihova početna vrijednost nije definirana. Obavezno inicijalizirajte vrijednosti svih lokalnih varijabli, inače je moguće da vaš program dobro radi lokalno, a dobije nula bodova na sustavu za evaluaciju.
- **Rješenje alocira premala polja:** Ako ne alocirate dovoljno velika polja, moguće je da vaš program dobro radi za primjere iz teksta zadatka, a dobije nula bodova na evaluatoru. Na primjer, ako s ulaza trebate pročitati niz od najviše 1000 znakova, u jeziku C potrebno je alocirati polje od najmanje 1001 elementa (`char s[1001]`).
- **Rješenje alocira prevelika polja:** Nepotrebno glomazna polja mogu prouzročiti da vaše rješenje prekorači memorijsko ograničenje i dobije nula bodova na evaluatoru. Primjerice, u jeziku C polje deklarirano sa `int x[1024][1024][100]` koristi 400MiB memorije.
- Manjak direktive `from sys import exit` u slučaju korištenja naredbe **`exit`** u jeziku Python.
- Korištenje **`conio.h`** u jezicima C i C++. Korištenje sintakse, tipova i funkcija specifičnih za Microsoftove C i C++ prevoditelje, npr. tip podataka **`int64`**. Korištenje naredbi **`itoa`** (koje nema pod Linuxom) ili **`atoi`** (koja radi malo drugačije pod Linuxom). Umjesto njih preporučamo korištenje funkcija `scanf` i `sprintf`.
- Rješenja čekaju na pritisnutu tipku nakon ispisa rezultata, npr. zbog naredbe **`system("pause")`**.
- Rješenja ispisuju rezultate u **pogrešnom formatu**, npr. ispisuju dva broja svaki u svoj redak umjesto u istom retku.
- Rješenja ispisuju **višak podataka** na standardni izlaz, npr. **debug informacije** ili poruke poput “Upišite broj:”, “Rješenje je:” i slično.
- Glavna funkcija u jezicima C i C++ definirana je kao `void main()` ili nema naredbe `return 0;`.
- Manjak potrebnih `include` direktiva u jezicima C i C++.

Dodatno, natjecateljima koji koriste C i C++ savjetujemo da prilikom testiranja obavezno uključite optimizacijsku opciju koja se koristi na sustavu na evaluaciju (`-O2`) te opcije koji upozoravaju na moguće probleme u kodu (primjerice `-Wall -Wextra`). Niže je ilustracija takvog prevođenja jednog rješenja sa Županijskog natjecanja 2016. godine. Prvo upozorenje koje je prijavio prevoditelj je lažna uzbuna, dok drugo ukazuje na stvarnu grešku (neinicijalizirana varijabla) zbog koje je natjecatelj nepotrebno izgubio bodove.

```
$ g++ -O2 -o poli.exe poli.cpp -Wall -Wextra
poli.cpp: In function 'int main()':
poli.cpp:16:20: warning: comparison between signed and unsigned integer expressions [-Wsign-compare]
   for (int i = 0; i < in.length(); ++i)
                   ^
poli.cpp:36:29: warning: 'curr' may be used uninitialized in this function [-Wmaybe-uninitialized]
   num[(curr == 0 ? 1 : curr) - 1] = val;
```