



Infokup

Državno natjecanje / Srednja škola (I. podskupina)

Algoritmi (Pascal/C/C++) – Prvi dan



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

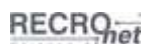
udruga mladih programera
dump



Glavni sponzor



Mali sponzori



Medijski pokrovitelji



Sadržaj

Upute za natjecatelje.....	2
Primjer pravilno napisanog programa.....	3
Zadaci	4
Zadatak: Svemir	5
Zadatak: POPO	6
Zadatak: Vojske	7



Upute za natjecatelje

Kod svakog pojedinog zadatka obratite pozornost na poglavlja **ulazni podaci** i **izlazni podaci**. Tu su definirana pravila vezana za format ulaznih i izlaznih podataka koji mora biti strogo poštovan kako bi vaša rješenja bila točno evaluirana. Za ilustraciju i bolje razumijevanje pogledajte poglavlje *primjer pravilno napisanog programa*.

Ulaz i izlaz treba se odvijati preko standardnog ulaza i standardnog izlaza (to znači **cin**, **cout**, **printf** i/ili **scanf**). Vaš program sa standardnog ulaza (**cin** ili **scanf**) mora očekivati samo ulazne podatke, a na standardni izlaz (**cout** ili **printf**) mora ispisivati samo izlazne podatke. 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 za taj zadatak, jer evaluator to ne očekuje. U pisanju programa dozvoljeno je koristiti samo standardne biblioteke, pa je tako primjerice zabranjeno koristiti naredbe **clrscr()**; ili **getch()**; jer su one dio **conio.h** biblioteke koja nije standardna u jeziku **C/C++**.

Važno je napomenuti da ne trebate kreirati izvršnu datoteku (.exe), jer će ju sustav sam kreirati iz izvornog koda na sljedeći način:

- **pascal**: `fpc -O1 -XS -oPRG.exe PRG.pas`
- **C**: `gcc -o PRG.exe PRG.c -std=c99 -O2 -s -static -lm`
- **C++**: `g++ -o PRG.exe PRG.cpp -O2 -s -static -lm`

(gdje je PRG ime programa)

Računalo na kojem se izvode programi i mjerena su vremenska ograničenja je Linux računalo s 2GHz radnog takta procesora.

Vaš program treba biti pisan u programskom jeziku C, C++ ili Pascal i mora regularno završiti svoje izvođenje. Program se treba izvršiti do kraja tj. do **return 0**; na kraju funkcije 'main' koja treba biti deklarirana kao **int main()**, ili naredbom **exit(0)**; Pogledajte priložene primjere. U programskom jeziku pascal program se treba izvršiti do kraja tj. Do **'end.'** ili naredbom **halt(0)**; Vaši programi **ne smiju pristupati** nikakvim datotekama **niti ih kreirati**, kršenje ovog pravila rezultirati će gubitkom bodova za taj zadatak. Bilo kakav pokušaj **pristupanja sistemskim datotekama računala na kojem se nalazi evaluator ili pokušaj upravljanja tim računalom** rezultirat će **diskvalifikacijom** tog natjecatelja. **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 budu generirali **točan rezultat unutar navedenog vremena i memorijskog ograničenja**. Obratite pažnju da svi zadaci ne nose 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.

Prilikom rješavanja zadataka preporučuje se korištenje olovke i papira za skiciranje i razradu algoritma.





Primjer 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.

Rješenje u programskom jeziku C

```
#include <stdio.h>

int main()
{
    int a, b;
    scanf("%d%d", &a, &b);
    printf("%d\n", a+b);
    printf("%d\n", a-b);
    return 0;
}
```

Rješenje u programskom jeziku C++

```
#include <iostream>
using namespace std;

int main()
{
    int a, b;
    cin >> a >> b;
    cout << a+b << endl;
    cout << a-b << endl;
}
```

Rješenje u programskom jeziku Pascal

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





Zadaci

U tablici možete pogledati ograničenja za zadatke:

Zadatak	Svemir	POPO	Vojske
Ulazni podaci	Standardni ulaz	Standardni ulaz	Standardni ulaz
Izlazni podaci	Standardni izlaz	Standardni izlaz	Standardni izlaz
Vremensko ograničenje	4 sekunda	4 sekunda	1 sekunda
Memorijsko ograničenje (Stack)	4 MB	4 MB	4 MB
Memorijsko ograničenje (Heap)	64 MB	64 MB	64 MB
Broj bodova	40	50	60
Ukupno bodova		150	



Zadatak: Svemir

Galaktička federacija odlučila je zatvoriti svoju poslovnicu u galaksiji „X57F34“ i cijelu svoju flotu preseliti u matičnu galaksiju u prvom kvadrantu. Brodovi lete kroz veliki međugalaktički prostor pravokutnog oblika. Kako mnogi njihovih svemirski brodovi lete blizu jedan drugom, primjetili su da neki brodovi imaju problema sa navigacijskim sustavom.

Okupio se tim stručnjaka, na čelu s kapetanom Kirkom, koji moraju što prije riješiti taj problem. Promatrajući kartu i koordinate svakog broda, zaključili su da probleme imaju samo oni brodovi kojima se neki brod nalazi u smjeru gore i desno. Dakle, navigacijski sustav ne radi svakom svemirskom brodu A za kojeg vrijedi da postoji barem jedan drugi brod B takav da je koordinata x broda B strogo veća od koordinate x broda A i koordinata y broda B strogo veća od koordinate y broda A.

Pomozite stručnjacima, na čelu s kapetanom Kirkom, u rješavanju ovog problema i za zadane koordinate brodova izračunajte koliko brodova trenutno ima funkcionalan navigacijski sustav.

NAPOMENA: ne postoje dva broda koja se nalaze na potpuno istoj poziciji.

Ulaz

U prvom retku se nalazi prirodan broj N ($1 \leq N \leq 1\,000\,000$), koji predstavlja broj svemirskih brodova u flotu.

U sljedećih N redaka nalaze se po dva cijela broja x_i i y_i ($0 \leq x_i, y_i \leq 1\,000\,000$), koji predstavljaju koordinate i-tog broja u floti.

Izlaz

U prvi i jedini redak potrebno je ispisati broj svemirskih brodova koji nemaju problema sa navigacijskim sustavom.

Test podaci

	Test 1	Test 2	Test 3
Ulaz	2 6 9 8 14	5 1 1 5 6 4 7 2 3 5 3	7 9 6 4 7 2 6 7 7 8 6 14 3 1 5
Izlaz	1	3	5



Zadatak: POPO

Nekoć davno, davno, jednom dalekom zemljom kraljevao je kralj uz pomoć svojih podanika. Svaki podanik ima svoju **nadređenu osobu**, koja je ili kralj ili neki drugi podanik. Promotrimo li bilo kojeg podanika, pa zatim popišemo slijed nadređenih osoba počevši od tog podanika, na posljetku ćemo uvijek doći do kralja. Svaki podanik ima svoj osobni broj, koji je veći od 1 a manji od milijun, i osobni brojevi svih podanika su različiti. Kralj ima osobni broj 1.

Kralj je odlučio uvesti porez na postojanje (POPO), koji će plaćati podanici, možda ne svi. Porez iznosi jedan groš po osobi godišnje.

Podanike je kralj podijelio u dvije grupe: u jednoj su porezni obveznici, a u drugoj sakupljači poreza. Pritom moraju vrijediti tri pravila:

- porezni obveznik ne može biti sakupljač poreza
- nadređena osoba poreznog obveznika je obavezno sakupljač poreza
- kralj je sakupljač poreza

Vaš zadatak je pomoći kralju da odredi iznos najvećeg mogućeg prihoda od poreza POPO.

Ulaz

U prvom redu, cijeli broj **N** ($1 \leq n < 1\,000\,000$), broj podanika (ne uključuje kralja).

U svakom od idućih **N** redova, dva broja, koji predstavljaju za pojedinog podanika: njegov osobni broj pa osobni broj nadređene osobe.

Izlaz

U prvom retku potrebno je ispisati jedan cijeli broj koji predstavljaju iznos najvećeg mogućeg prihoda od poreza.

Test podaci

	Test 1	Test 2	Test 3
Ulaz	3 3 1 4 2 2 1	5 3 6 2 6 6 1 5 3 4 6	5 6 1 3 6 4 6 2 1 5 2
Izlaz	2	3	3



Zadatak: Vojske

U nadaleko poznatoj zemlji Nigdjezemskoj sukobile se dvije vojske, vojska X i vojska Y.

Vojnici dviju sukobljenih strana nalaze se na dvodimenzionalnoj mapi od N redova i M stupaca ($2 \leq N, M \leq 1000$). Cilj vojske X je ukrasti zastavu vojske Y (zastava je na mapi označena sa znakom F) i zatim se zajedno okupiti na bilo kojem polju na mapi da proslave svoju pobjedu.

Vojska X je odlučila provesti misiju „Vrana“ svrha koje je ukrasti zastavu bez da to primijete vojnici vojske Y. Stoga oni žele, u svakom trenutku za vrijeme trajanja misije, da udaljenost od bilo kojeg vojnika vojske X do njemu najbližeg vojnika vojske Y bude veća ili jednaka R. Svaki vojnik se u jednom koraku smije kretati samo za jedno polje u smjeru gore, dolje, lijevo ili desno. Udaljenost od polja A do polja B se određuje kao broj koraka koji je potreban zamišljenom vojniku na polju A da dođe do polja B.

Pomozite generalu vojske X da odredi najveći mogući takav R pri kojem je moguće ostvariti misiju (ukrasti zastavu i proslaviti svoju pobjedu okupljanjem na bilo kojem odabranom mjestu).

Ulaz

U prvom retku nalaze se dva cijela broja N, M (broj redaka i stupaca matrice). Zatim slijedi N redaka koji opisuju mapu. U svakom retku znak točke označava prazno polje mape, znakovi X i Y označavaju vojnike i znak F označava zastavu.

Izlaz

U jednom retku jedan cijeli broj koji označava najmanju moguću vrijednost za broj R.

Test podaci

	Test 1	Test 2	Test 3
Ulaz	4 4 ...F .Y.. ...X X...	4 4 .Y.F .Y.. ..YY X...	4 4 .Y.F .Y.. ..X. X...
Izlaz	2	0	2