



Infokup

Državno natjecanje / Srednja škola (I. podskupina)

Algoritmi (Pascal/C/C++) – Drugi dan



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

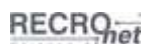
udruga mladih programera
dump



Glavni sponzor



Mali sponzori



Medijski pokrovitelji



Sadržaj

Upute za natjecatelje.....	2
Primjer pravilno napisanog programa.....	3
Zadaci	4
Zadatak: Frizbi	5
Zadatak: JURTEK.....	7
Zadatak: Akcija	10



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	Frizbi	JURTEK	Akcija
Ulazni podaci	Standardni ulaz	Standardni ulaz	Standardni ulaz
Izlazni podaci	Standardni izlaz	Standardni izlaz	Standardni izlaz
Vremensko ograničenje	2 sekunde	4 sekunde	1 sekunda
Memorijsko ograničenje (Stack)	4 MB	4 MB	4 MB
Memorijsko ograničenje (Heap)	32 MB	32 MB	32 MB
Broj bodova	50	70	80
Ukupno bodova		200	



Zadatak: Frizbi

Popularna igra Frizbi igra se tako da igrači jedan drugome dodaju plastični leteći disk zvan frizbi. Jedan igrač baci frizbi drugom igraču koji ga potom opet prosljeđuje nekom igraču iz grupe. Tako se na jednoj livadi okupila grupa od N igrača frizbija. Igrači su označeni brojevima od 1 do N .

Da bi igru učinili zanimljivijom, odlučili su umjesto jednog frizbija koristiti **dva frizbija** istovremeno – crveni i žuti. Svaki igrač je unaprijed odlučio kojem drugom igraču će dodavati crveni, a kojem igraču žuti frizbi (moguće je da će dodavati i crveni i žuti frizbi istoj osobi). Za početak, jedan igrač je uzeo crveni a neki drugi igrač je uzeo žuti frizbi, pa njihova igra može započeti.

Igrači igraju frizbi u koracima, tako da se u svakom koraku prosljede i crveni i žuti frizbi istovremeno. Moguće je da se oba frizbija nađu kod iste osobe istovremeno, dakle u istom koraku, a takva situacija zahtjeva posebno spretnost igrača koji su prethodno uvježbali igranje frizbija s obje ruke. Nas zanima koliko ima takvih igrača.

Ulaz

U prvom retku se nalaze tri prirodna broja N ($2 \leq N \leq 300\,000$), X i Y ($2 \leq X, Y \leq N$), pri čemu N označava ukupni broj igrača na livadi, X je redni broj igrača koji prvi baca crveni frizbi, a Y je redni broj igrača koji prvi baca žuti frizbi.

U drugom retku se nalazi jedan prirodni broj R ($2 \leq R \leq N$), koji označava broj igrača koji sudjeluju u igri s crvenim frizbijem. U sljedećih R redaka nalaze se po dva broja a i b ($2 \leq a, b \leq N$), koji označavaju da igrač s rednim brojem a uvijek dodaje crveni frizbi igraču s rednim brojem b .

U sljedećem retku nalazi se jedan prirodan broj S ($2 \leq S \leq N$), koji označava broj igrača koji sudjeluju u igri s žutim frizbijem. U sljedećih S redaka nalaze se po dva broja a i b ($2 \leq a, b \leq N$), koji označavaju da igrač s rednim brojem a uvijek dodaje žuti frizbi igraču s rednim brojem b .

Svaki igrač će u barem jednom trenutku imati neki frizbi u rukama.

Izlaz

U prvom i jedinom retku potrebno je ispisati broj igrača kod kojih se oba frizbija mogu naći istovremeno. Ukoliko se oba frizbija nikad neće naći kod iste osobe istovremeno, potrebno je ispisati 0.

NAPOMENA: Rješenja koja za svaki test primjer ispišu 0 dobit će 0 bodova.



Test podaci

	Test 1	Test 2	Test 3
Ulaz	7 1 3 7 1 2 2 3 3 4 4 5 5 6 6 7 7 4 3 7 1 3 7 1 3	9 1 5 5 1 2 2 3 3 4 4 9 9 4 5 5 6 6 2 2 7 7 8 8 7	9 1 7 6 1 2 2 3 3 4 4 5 5 6 6 3 7 7 8 8 2 2 9 9 4 4 6 6 5 5 9
Izlaz	1	0	1



Zadatak: JURTEK

Jure je odlučio u Hrvatskoj pokrenuti tvrtku Jur-tek za razvoj najnovijih elektroničkih integriranih krugova. Zaposlio je nekoliko domaćih inženjera koji su dobili zadatak osmisлити dizajn kruga za zbrajanje, temeljen na najnovijoj tehnologiji. Dostupne su im tri vrste logičkih krugova koji nose vrlo upečatljiva i znakovita imena: Z_0 , Z_1 i K .

Krugovi Z_0 , Z_1 imaju dva ulaza i dva izlaza, a logičke vrijednosti izlaza ovisno o onima na ulazu dane su prema ovim tablicama:

Z_0			
u_1	u_2	i_1	i_2
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0

Z_1			
u_1	u_2	i_1	i_2
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	1

Krug Z_0 možemo shvatiti kao da zbraja dva bita, dakle rezultat na izlazima može se protumačiti kao broj ulaza koji imaju logičku vrijednost 1. Kako postoje samo dva ulaza, to rezultat mogu biti samo brojevi 0, 1 ili 2, a u binarnom sustavu to se piše kao 00, 01 i 10.

Slično tomu, Krug Z_1 možemo shvatiti kao da zbraja dva bita na ulazima i tome još pridodaje broj jedan.

Dakle, izlaz i_2 ovih logičkih krugova predstavlja znamenku rezultata manje težine, a izlaz i_1 , predstavlja znamenku više težine koju je zatim potrebno pribrojiti drugim bitovima više težine želimo li ostvariti binarno zbrajalo.

Tri prethodna paragrafa su, naravno, samo pojašnjenje gornje dvije tablice koje o svemu navedenog precizno i točno govore same za sebe.

Krug K ima 3 ulaza: u_0 , u_1 i u_5 , i izlaz i . Ako se na ulazu u_5 nalazi logička vrijednost 0, onda će na izlazu i biti logička vrijednost istovjetna onoj na ulazu u_0 a ako se na ulazu u_5 nalazi logička vrijednost 1, tada će na izlazu i biti logička vrijednost istovjetna onoj na ulazu u_1 .



Inženjerima je zadano da osmisle zbrajalo koje zbraja dva cijela broja, A i B , koji nisu negativni. Brojevi su zadani binarnim sustavom pomoću točno n bitova svaki, tako da se logičke vrijednosti bitova nalaze na linijama ulaznog nivoa, dakle za broj A na linijama $A_1 - A_n$ i za broj B na linijama $B_1 - B_n$.

Inženjeri zatim smještaju logičke krugove tipa Z_0 , Z_1 i K u daljnje nivoe. **Svaki ulaz logičkog kruga na nekom nivou smije biti povezan isključivo s nekim (bilo kojim) izlazom iz bilo kojeg prethodnog nivoa, ili s nekom linijom ulaznog nivoa.**

U posljednjem, izlaznom nivou, nalaze se linije rješenja, njih točno $n+1$, označeni sa $C_1 - C_{n+1}$. Za te linije vrijedi isto pravilo kao i za ulazne linije logičkih krugova: **smiju biti povezane s nekim, bilo kojim, izlazom iz bilo kojeg prethodnog nivoa, ili s nekom linijom ulaznog nivoa.**

Ulaz

U prvom retku zadan je cijeli broj n ($1 \leq n \leq 256$) koji predstavlja broj bitova broja A , a ujedno i broj bitova broja B .

U drugom retku nalazi se cijeli broj m , ($m \leq 1000$), koji predstavlja maksimalan broj nivoa koji inženjeri smiju upotrijebiti. U broj m se ubrajaju i ulazni nivo i izlazni nivo.

Izlaz

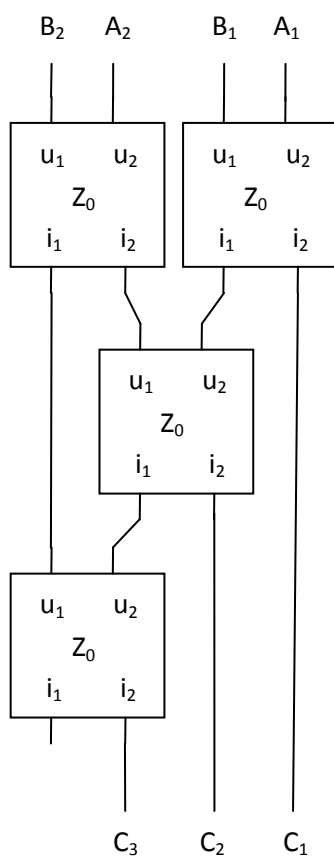
U jednom retku, jedan cijeli broj koji predstavlja najmanji mogući broj logičkih krugova koji je potrebno upotrijebiti da bi se izvelo funkcionalno zbrajalo, ili broj -1 ako zbrajalo nije moguće izvesti.



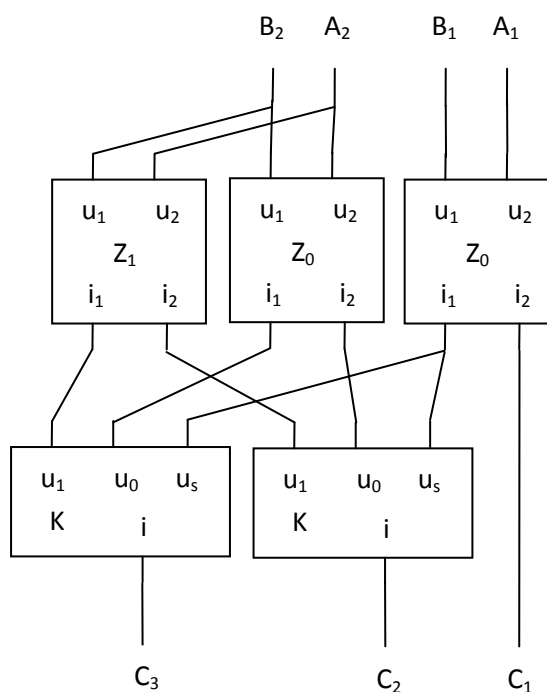
Test podaci

	Test 1	Test 2	Test 3
Ulaz	2 5	2 4	8 6
Izlaz	4	5	59

Slika uz primjer 1:



Slika uz primjer 2:





Zadatak: Akcija

U jednom velikom američkom gradu održava se parada za Dan Zahvalnosti. U paradi sudjeluje velik broj ljudi među kojima se skriva opasni kriminalac kojeg policija pokušava uhvatiti. Kako bi ga napokon uspješno priveli, policajci žele što bolje isplanirati akciju uhićenja.

Parada se održava na području pravokutnog oblika, koje je zapovjedništvo policije podijelilo na **jedinične sektore**. Da bi lakše isplanirali akciju, na karti su numerirali sektore po recima i stupcima, na način da se gornji lijevi sektor nalazi u prvom retku i prvom stupcu.

Policajci su pokrenuli svoje informatičke sustave i napravili satelitsku snimku parade, a računalo je odmah prebrojalo i ispisalo koliko se ljudi nalazi u svakom pojedinom sektoru. Svi sudionici parade, uključujući i kriminalca, kreću se toliko polagano da možemo smatrati da se uopće ne kreću.

Kako bi se što lakše približili kriminalcu, policajci su se odlučili pomicati po sektorima u samo dva smjera - **dolje** ili **desno** (jer iz suprotnog smjera sije sunce).

Policajci se kreću nepoznatom brzinom, a kada stignu u sektor u kojem se nalazi kriminalac, pričekat će početak akcije.

Kako bi se dodatno osigurali da neće pobuditi sumnju, odlučili su biti malo oprezniji i neprimjetniji. Odlučili su da u svaki sektor sa **K** sudionika parade smije **iz pojedinog smjera** ući najviše **K/3** policajaca (koristimo cjelobrojno dijeljenje).

Od zapovjedništva policije tog velikog američkog grada dobit ćete podatke o broju ljudi u svakom sektoru, položaj svih policajaca na terenu i položaj kriminalca. Zapovjedništvo vas moli da izračunate **koliko najviše policajaca** može doći u sektor u kojem se nalazi kriminalac, a da ne izazovu sumnju.

Ulaz

U prvom retku nalaze se dva broja **R** i **S** ($1 \leq R, S \leq 30$), visina i širina cijelog područja na kojem se održava parada.

U drugom retku nalaze se dva broja prirodna broja **X** i **Y** ($1 \leq X \leq R, 1 \leq Y \leq S$), koji predstavljaju broj retka i broj stupca sektora u kojem se nalazi kriminalac.

U svakom od slijedećih **R** redaka nalazi se po **S** cijelih brojeva **$a_{i,j}$** ($0 \leq a_{i,j} \leq 6000$), koji predstavljaju broj ljudi u sektoru u *i*-tom retku i *j*-tom stupcu, ($1 \leq i \leq R, 1 \leq j \leq S$).

U sljedećem retku se nalazi broj **N** ($1 \leq N \leq 2000$), koji predstavlja broj policajaca na terenu. U slijedećih **N** redaka nalaze se po dva prirodna broja **x_i** i **y_i** ($1 \leq x_i \leq R, 1 \leq y_i \leq S$), koji predstavljaju broj retka i stupca sektora u kojem se nalazi *i*-ti policajac.



Izlaz

U prvi i jedini redak ispisati najveći mogući broj policajaca koji mogu stići u sektor u kojem se nalazi kriminalac bez da izazovu sumnju.

Test podaci

	Test 1	Test 2	Test 3
Ulaz	3 3 3 3 3 3 3 3 3 3 3 3 7 3 1 1 1 3 2 1	3 4 3 2 3 3 3 3 3 3 3 3 3 3 3 3 4 1 2 1 4 2 1 2 3	4 3 4 3 3 6 10 6 7 2 3 6 4 3 3 3 5 1 2 1 3 2 1 2 2 3 1
Izlaz	3	2	2