

Infokup

Državno natjecanje / Srednja škola (II. podskupina)

Algoritmi (Pascal/C/C++) – Drugi dan



Agencija za odgoj i obrazovanje
Education and Teacher Training Agency



MINISTARSTVO ZNANOSTI, OBRAZOVANJA
I ŠPORTA REPUBLIKE HRVATSKE

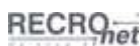
udruga mladih programera
dump



Glavni sponzor



Mali sponzori



Medijski pokrovitelji



Microsoft

Microsoft Innovation Center
Ljubljana

Microsoft Innovation Center
Vratiska



Sadržaj

Upute za natjecatelje.....	2
Primjer pravilno napisanog programa.....	3
Zadaci	4
Zadatak: Dizalice	5
Zadatak: AMTEL.....	7
Zadatak: Nogomet	9



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;
}
```

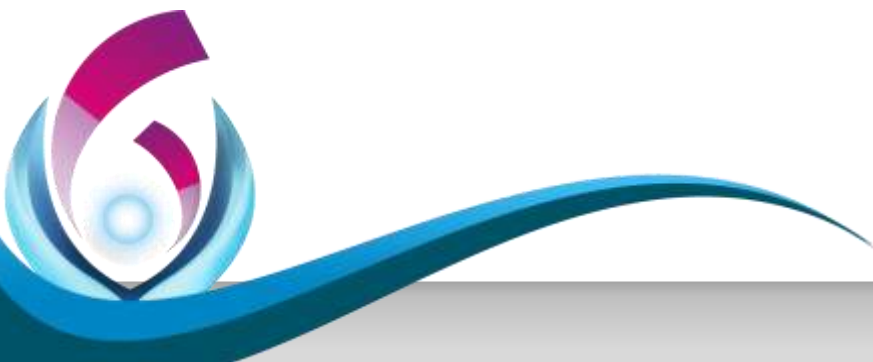
Rješenje u programskom jeziku C++

```
#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	Dizalice	AMTEL	Nogomet
Ulazni podaci	Standardni ulaz	Standardni ulaz	Standardni ulaz
Izlazni podaci	Standardni izlaz	Standardni izlaz	Standardni izlaz
Vremensko ograničenje	1 sekunda	4 sekunda	1 sekunda
Memorijsko ograničenje (Stack)	4 MB	4 MB	4 MB
Memorijsko ograničenje (Heap)	64 MB	32 MB	32 MB
Broj bodova	50	70	80
Ukupno bodova		200	



Zadatak: Dizalice

U jednom našem obalnom gradu postoji tvrtka „Dižemo sve d.o.o.“. Spomenuta se tvrtka bavi utovarom teretnih kontejnera u brodove. Tvrtka ima dva radnika koji upravljaju **dvjema dizalicama**. Njihov zadatak je u **što kraćem roku** utovariti sve kontejnere, s tim da svaka dizalica prenosi jedan po jedan kontejner.

Zadana je kvadratna mreža na kojoj se nalaze znakovi '.' koji predstavljaju **prazan prostor** i znakovi '#' koji predstavljaju **zapreku** preko koje se dizalica ne može kretati.

Na početku radnog vremena obje se dizalice nalaze u gornjem lijevom uglu mreže (na položaju (0,0)). Ruka dizalice može se pomicati samo u smjerovima paralelnima s rubovima mreže (gore, dolje, lijevo i desno). Za pomicanje s jednog polja na drugo potrebna je jedna sekunda, a vrijeme potrebno za podizanje i spuštanje kontejnera je zanemarivo. Dizalice su neovisne jedna o drugoj, što znači da se mogu naći na istom polju u isto vrijeme.

Za svaki od kontejnera zadan je početni položaj i položaj na koji ga je potrebno premjestiti. Vaš zadatak je izračunati koliko je **najmanje vremena** potrebno za **prebaciti sve kontejnere**, tako da radnici mogu što prije otići prezalagajiti kakav brzi obrok.

Ulaz

U prvom retku se nalaze cijeli brojevi **N** i **M** ($1 \leq N, M \leq 15$), koji predstavljaju širinu i visinu kvadratne mreže.

U sljedećih **N** redaka se nalazi po **M** znakova koji prikazuju prepreke ('#') i slobodan prostor ('.') u mreži.

Potom, u idućem se retku nalazi broj **K** ($1 \leq K \leq 15$), koji predstavlja broj kontejnera koje je potrebno premjestiti. Nakon toga, u sljedećih **K** redaka se nalaze po četiri broja **X1, Y1, X2 i Y2** ($1 \leq X1, Y1, X2, Y2 \leq 15$), koji predstavljaju položaj u mreži na kojem se kontejner nalazi i položaj na kojeg ga je potrebno premjestiti. Oba spomenuta položaja kontejnera uvijek će biti položaji slobodnog prostora u kvadratnoj mreži i mapa će biti takva da obje dizalice mogu doći do tih položaja.

Izlaz

U prvi i jedini redak ispisati jedan cijeli broj koji predstavlja najkraće moguće vrijeme, u sekundama, u kojem radnici mogu premjestiti sve kontejnere.



Test podaci

	Test 1	Test 2	Test 3
Ulaz	3 3#. .#. 2 1 1 3 3 1 3 3 1	4 3 ..# #.. .#. ... 3 1 1 2 3 3 1 4 2 4 1 3 3	4 4 #..# #... #..# 4 1 1 4 2 4 3 2 2 2 3 1 4 4 2 3 4
Izlaz	4	10	11



Zadatak: AMTEL

Inženjeri u Amtelu, najvećoj svjetskoj tvrtci za izradu integriranih krugova, dobili su zadatak osmisli dizajn kruga za zbrajanje, temeljen na najnovijoj tehnologiji. Dostupne su im tri vrste logičkih krugova koji nose vrlo upečatljiva 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 K ima 3 ulaza: u_0 , u_1 i u_s , i izlaz i . Ako se na ulazu u_s nalazi logička vrijednost 0, onda će na izlazu i biti logička vrijednost istovjetna onoj na ulazu u_0 a ako se na ulazu u_s 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 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 broj m , ($m \leq 10\,000\,000$), koji predstavlja maksimalan broj logičkih krugova koji inženjeri smiju upotrijebiti.

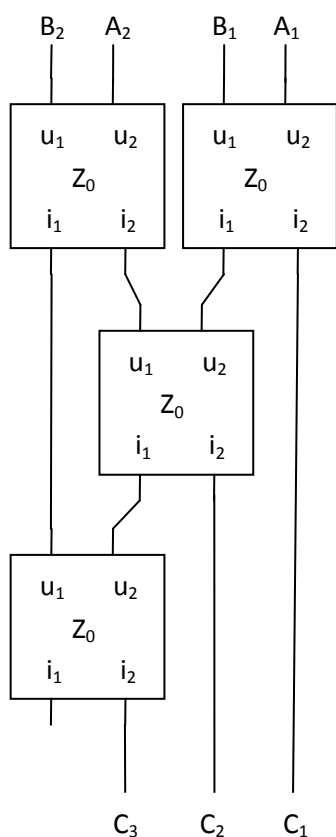
Izlaz

U jednom retku, jedan cijeli broj koji predstavlja najmanji mogući broj nivoa koji je potrebno upotrijebiti da bi se izvelo funkcionalno zbrajalo, ili broj -1 ako zbrajalo nije moguće izvesti. U taj broj se ubrajaju i ulazni nivo i izlazni nivo.

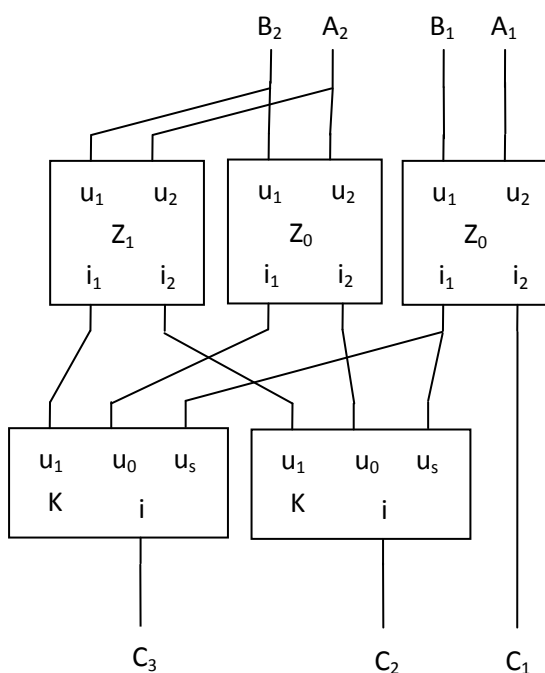
Test podaci

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

Slika uz primjer 1:



Slika uz primjer 2:





Zadatak: Nogomet

U malom selu Zagvozd otvoren je novi nogometni klub. Poznato je da je nogomet najpopularniji sport u Hrvatskoj i zbog toga su sva djeca odmah pohrlila na upis.

S obzirom na njihove druge obveze djeca su odabrala različite termine treninga koji njima odgovaraju. Svako dijete koje je polaznik nogometnog tečaja odabralo je jedan ili više termina. Svaki termin ima ograničen broj polaznika.

Trener je odabrao neke polaznike, s obzirom na njihove nogometne sposobnosti. Za svakog pojedinog odabranog polaznika odabrao je nekoliko termina koje najbolje odgovaraju tom polazniku.

Potrebne je rasporediti polaznike u termine s time da se zadovolji što je moguće više trenerovih ograničenja uz zadovoljena sva ograničenja polaznika (odnosno da pohađaju **jedan od termina** koji su naveli kao mogući) i moraju biti zadovoljena sva ograničenja broja polaznika svakog pojedinog termina.

Ispisati najmanji mogući broj trenerovih ograničenja koje je potrebno prekršiti da bi postojao raspored koji zadovoljava sva ostala ograničenja.

Napomena:

Svako dijete potrebno je dodijeliti u točno jedan termin.

Ukoliko je za nekog polaznika trener odabrao više različitih termina tada smatramo da trenerovo ograničenje nije prekršeno u slučaju da je polaznik dodijeljen u bilo koji od navedenih termina.

Ulaz

U prvom retku nalaze se 3 cijela broja: broj N ($1 \leq N \leq 100$) – broj polaznik, zatim broj T ($1 \leq T \leq 30$) -broj termina, i zatim broj E ($1 \leq E \leq 150$) – broj trenerovih ograničenja

U drugom retku nalaze se brojevi C_i ($1 \leq i \leq T$), ($1 \leq C_i \leq N$) – maksimalni kapacitet pojedinog termina.

U idućih N redaka nalazi se popis termina koje je odabrao pojedini polaznik:

prvi broj u retku je cijeli broj K_i (broj termina koje je pojedini polaznik odabrao)

slijedi K_i brojeva ($1 \leq K_i \leq T$) (redni brojevi termina u koje polaznik može pripadati)

Sljedećih E redaka predstavlja trenerova ograničenja:

prvi broj u retku je R_i - redni broj polaznika, ($1 \leq R_i \leq N$)

drugi broj u retku je redni broj termina za kojeg trener smatra da je dobar odabir za pojedinog polaznika.



Izlaz

Jedan broj, minimalni broj trenerovih ograničenja koja je potrebno prekršiti da bi se ostvario navedeni raspored.

Test podaci

	Test 1	Test 2	Test 3
Ulaz	4 3 1 2 2 2 2 1 2 1 3 1 3 2 1 3 1 2	5 3 5 1 1 7 2 1 2 2 1 2 3 1 2 3 2 1 3 1 3 5 3 1 3 4 2 1 1 5 1	10 5 5 1 3 1 1 11 5 1 2 3 4 5 4 2 3 4 5 4 1 2 4 5 5 1 2 3 4 5 3 1 3 5 4 1 2 3 4 4 1 2 4 5 3 1 2 4 3 2 3 5 5 1 2 3 4 4 2 5 3 2 4 4 1 6 3
Izlaz	0	1	1

Objašnjenje za primjer 1:

U selu se nalaze 4 dječaka koji su polaznici. Postoje tri termina za treniranje te je kapacitet svakog od njih jednak 2 polaznika.

Prvi polaznik može pripadati terminima 1 i 2, drugi i treći polaznik samo terminu 3, dok četvrti polaznik može pripadati terminu 1 i 3.

Trener je postavio samo ograničenje da bi prvom polazniku najbolje odgovarao drugi termin.

Ukoliko djecu rasporedimo prema sljedećem rasporedu

1. polaznik -> 2. termin
2. polaznik i 3. polaznik -> 3. termin
4. polaznik -> 1. termin

tada su zadovoljena sva ograničenja, stoga je rješenje 0.