

Sveučilište u Zagrebu  
**Fakultet Elektrotehnike i Računarstva**  
Natjecateljsko programiranje  
Zimski semestar 2011. / 2012.

Natjecateljsko Programiranje  
Završni ispit

**04. 02. 2011.**

Poštovane studentice i studenti!

Ispit (natjecanje) se sastoji od nekoliko zadataka različite težine. Zadaci **nisu** poredani po težini. Svi zadaci nose **jednak** broj 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.

Vaš program ulazne podatke čita sa standardnog ulaza (*stdin*), a izlaz ispisuje na standardni izlaz (*stdout*). Dobro pročitajte poglavlja ULAZNI PODACI i IZLAZNI PODACI kod svakog pojedinog zadatka. Na tom mjestu piše sve o formatu ulaznih i izlaznih podataka koji mora biti **strogo poštovan** (tj. **ne smijete** ispisivati nikakav dodatni tekst, kao "Rješenje je" i slično) kako bi vaša rješenja bila točno evaluirana. Za ilustraciju i bolje razumijevanje pogledajte poglavlje TEST PRIMJERI u kojem će se nalaziti nekoliko jednostavnijih primjera ulaznih podataka i njima odgovarajućih izlaznih podataka.

**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.

**Sustav za evaluaciju nalazi se na računalu s LINUX operativnim sustavom. Stoga, da bi se vaš program uspješno evaluirao, 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.

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)'.

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 obratite pažnju na **ograničenja** kod ulaznih podataka jer su ona izrazito bitna za osmišljavanje efikasnog algoritma.

Kod svakog zadatka naći ćete nekoliko test primjera koji vam mogu pomoći da bolje shvatite zadatak. Međutim, ako vaš program radi ispravno za zadane test primjere to još uvijek **nije garancija** niti da vaš program radi korektno za ostale test podatke niti da je dovoljno brz.

**Sretno!**

**Natjecateljsko programiranje – Završni ispit 2011./2012.**

ZADATAK	PLODOVI	HASHMAT	USPON	IPV6	MARIO
ulazni podaci	standardni ulaz				
izlazni podaci	standardni izlaz				
vremensko ograničenje	1 sekunda	1 sekunda	1 sekunda	1 sekunda	1 sekunda
memorijsko ograničenje	32 MB	32 MB	32 MB	32 MB	32 MB
broj bodova	100	100	100	100	100
	ukupno 500				

# Plodovi

Autor: Matija Šantl  
Broj bodova: 100

Vremensko ograničenje: 1 s  
Memorijsko ograničenje: 32 MB

Na dalekom otoku usred oceana živi osoba Iks. Kako se bliži zima, osoba Iks sakuplja razne plodove. Sprava pomoću koje suši plodove za zimu je u obliku pravokutne rešetke. Osoba Iks zna kolika je kalorijska vrijednost kojeg ploda, te na kojoj se poziciji na spravi za sušenje nalazi. Osoba Iks se zabrinula za svoje plodove jer je vidjela da se sprema oluja. Kako mu vjetar ne bi otpuhao plodove, učvršćuje ih drvenim štapovima koje može stavljati uzduž i poprijeko svoje sprave za sušenje, tako da plodovi kroz koje štap prolazi uzduž i poprijeko smatraju sigurnima od oluje. Nađite najveću sumu kalorijskih vrijednosti plodova koje osoba Iks može sačuvati od oluje.

**Napomena:** osoba Iks može učvrstiti 0 ili više plodova.

## Ulazni podaci

U prvom retku nalazi se dva prirodna broja **R** i **C** ( $1 \leq R \leq 10$ ,  $1 \leq C \leq 100$ ), koja redom označavaju broj redaka i stupaca sprave za sušenje.

U svakom od sljedećih **R** redaka nalazi se **C** cijelih brojeva  $M_{i,j}$  ( $-5000 \leq M_{i,j} \leq 5000$ ), koji redom označavaju kalorijsku vrijednost ploda koji se nalazi u *i*-tom retku i *j*-tom stupcu sprave za sušenje.

## Izlazni podaci

U prvi i jedini redak ispišite najveći mogući zbroj kalorijskih vrijednosti koje osoba Iks može sačuvati za zimu.

## Test primjeri

**ULAZ :**

1 1  
5

**IZLAZ :**

5

**ULAZ :**

4 5  
1 -1 1 -1 1  
-1 2 -3 4 -5  
5 -4 3 -2 1  
-1 1 -1 1 -1

**IZLAZ :**

12

**ULAZ :**

1 1  
-5

**IZLAZ :**

0

## Pojašnjenja

**Pojašnjenje drugog test primjera:** osoba Iks učvršćuje 1. i 3. redak te 1., 3. i 5. stupac.

**Pojašnjenje trećeg test primjera:** osoba Iks ne želi sačuvati trule plodove, pa je bolje da ih uopće ne učvršćuje.

## Bodovanje

U test primjerima vrijednim ukupno 40% bodova **R** i **C** će biti u intervalu  $1 \leq R, C \leq 8$ .

# Hashmat

Autor: Bruno Rahle  
Broj bodova: 100

Vremensko ograničenje: 1 s  
Memorijsko ograničenje: 32 MB

Zadana je hash funkcija koja se računa na slijedeći način:

$$\left( \sum_{i=1}^n A^{n-i+1} \cdot (\text{ORD}(S_i) + B) \right) \% M$$

tj. jednaka je polinomu koji za koeficijente ima ASCII vrijednosti znakova stringa (nužno mala slova engleske abecede) zbrojene s nekom konstantom **B**, a za vrijednost **x** mu je uvrštena neka konstanta **A**. Na kraju je uzet ostatak pri dijeljenju dobivenog broja s **M**.

Na primjer, za **A=2**, **B=-96** i **M=1007**, string "banana" ima vrijednost hasha  $(64*2+32*1+16*14+8*1+4*14+2*1) \% 1007 = 450$ , dok za vrijednosti **A=12**, **B=-96**, **M=101** vrijednost hasha iste riječi je 57.

Za zadani tekst, potrebno je izračunati hash na svim podnizovima duljine **K** i ispisati redom sve vrijednosti. Na primjer ako je tekst "banana", a **K=2**, trebalo bi ispisati redom hasheve za podnizove "ba", "an", "na", "an", "na".

## Ulazni podaci

U prvom retku ulaznih test primjera unose se 4 cijela broja: **A** ( $1 \leq A \leq 1\,000$ ), **B** ( $-97 \leq B \leq 100$ ), **M** ( $2 \leq M \leq 10\,000$ ), **K** ( $1 \leq K \leq 10\,000$ ;  $K \leq L$ ).

U drugom redu string **S**, sastavljen od malih slova engleske abecede i ne duži od 100 000 znakova.

## Izlazni podaci

Ako sa **L** označimo dužinu **S**, onda je potrebno ispisati **L-K+1** redaka, u prvom rezultat hashiranja znakova stringa od 1. do **K**-tog, u drugom rezultat hashiranja znakova stringa počevši od 2. do **K+1**, u trećem rezultat hashiranja od 3. do **K+2**, itd. sve do posljednjeg retka u kojem ćemo ispisati rezultat hashiranja od **L-K+1** do **L**-tog znaka.

## Test primjeri

### ULAZ :

2 -96 1007 2  
banana

### IZLAZ :

10  
32  
58  
32  
58

### ULAZ :

7 -93 107 3  
politicar

### IZLAZ :

14  
38  
9  
84  
66  
51  
47

## Bodovanje

U test primjerima vrijednim ukupno 40% bodova, umnožak **K** i **L** bit će manji od 5 000 000.

# Uspón

Autor: Dino Šantl  
Broj bodova: 100

Vremensko ograničenje: 1 s  
Memorijsko ograničenje: 32 MB

Poznato je da C zgrada FER-a ima 13 katova. Ali manje je poznata činjenica da ponekad ne rade liftovi. Veliki problem nastaje kada završe završni ispiti i svi navale na uvide u C zgradu a liftovi ne rade. Svi znamo da je tada uspon po stepenicama veoma stresan, te su neki studenti došli na ideju da bi se na C zgradu mogli penjati s vanjske strane. Vaš zadatak je odrediti na **koliko** se načina može doći na **vrh** zgrade. Kako se neki ispiti još ispravljaju **zabranjeno** je prolaziti po **prozorima** tih prostorija da se ne bi ometalo profesore i asistente u radu.

Zgrada je zadana svojim dimenzijama visinom i širinom ( $V \times S$ ), te je zadano  $K$  parova brojeva  $X, Y$  koji predstavljaju cijelobrojne koordinate prozora na koji se **ne smije doći**.

Student se prije penjanja uvijek nalazi na dnu zgrade, te na sredini širine zgrade ( $S / 2$ ), ako  $S$  nije paran broj, potrebno je zaokružiti  $S / 2$  na manje. Početna koordinata je:  $(S / 2, 0)$ .

Student se smije kretati po zgradi gore, gore-lijevo i gore-desno, ili matematički napisano ako je student na prozoru sa koordinatom  $(x, y)$  smije ići na prozore s koordinatama  $(x, y+1)$ ,  $(x-1, y+1)$ ,  $(x+1, y+1)$ , naravno ako je na tim koordinatama prozor koji nije zabranjen i ako koordinate ne prelaze dimenzije zgrade.

## Ulazni podaci

U prvom retku nalaze se dva prirodna broja,  $V$  i  $S$  ( $1 \leq V, S \leq 1\,000$ ), međusobno odvojena razmakom.

U drugom retku nalazi se prirodan broj  $K$  ( $0 \leq K \leq 1\,000$ ).

U sljedećih  $K$  redaka nalazi se dva broja  $X$  i  $Y$  međusobno odvojena razmakom. ( $0 \leq X, Y \leq 1\,000$ ).

## Izlazni podaci

U prvi i jedini redak ispišite na koliko se načina može doći do vrha zgrade modulo 100 007.

## Test primjeri

**ULAZ :**

4 3

3

0 2

1 3

2 2

**IZLAZ :**

6

**ULAZ :**

3 3

0

**IZLAZ :**

7

**ULAZ :**

3 3

2

1 1

1 2

**IZLAZ :**

2

## Pojašnjenja

### Pojašnjenje prvog test primjera:

Oznakom **X** označen su zabranjeni prozori, oznakom **O** označeni su dopušteni prozori te oznakom **I** označena je početna koordinata studenta.

O	X	O
X	O	X
O	O	O
O	I	O

Početna koordinata: **(1, 0)**

Zabranjeni prozori: **(0, 2)** , **(1, 3)** , **(2, 2)**

**Broj načina na koje se može doći do vrha: 6**

Načini dolaska do vrha:

(1, 0)-(0, 1)-(1, 2)-(0, 3),

(1, 0)-(1, 1)-(1, 2)-(0, 3),

(1, 0)-(2, 1)-(1, 2)-(0, 3),

(1, 0)-(0, 1)-(1, 2)-(2, 3),

(1, 0)-(1, 1)-(1, 2)-(2, 3),

(1, 0)-(2, 1)-(1, 2)-(2, 3).

# IPv6

Autor: Bruno Rahle  
Broj bodova: 100

Vremensko ograničenje: 1 s  
Memorijsko ograničenje: 32 MB

Haker Mirko živi u svijetu budućnosti. U tom svijetu Internet kakav danas poznajemo u potpunosti je cenzuriran. Posljednje utvrde istinitih riječi sulude ambicije pomahnitalih korporacija ruše kao kule od karata. Zli moćnici natjerali su čitav svijet da prijeđe na internetski protokol verzije 6 (sa 4 kakav je danas u primjeni). Mirko se dokopao popisa IPv6 adresa koje su u vlasništvu tih korporacija. Sada mu je cilj "ishakirati" ih.

Međutim, kako je broj adresa koje su u njihovom vlasništvu ogroman, a Mirko nema neograničene resurse, odlučio je da bi mu se najviše isplatilo da se koncentrira na neki dio tih adresa. Kako je trenutno zauzet planiranjem napada i snimanjem anonimnih video klipova s mačkama, traži od vas da mu pomognete pronaći **niz uzastopnih IP adresa koje koristi najviše korporacija**.

Jedna IPv6 adresa sastoji se od **128 bitova podataka**, a prikazuje se kao 8 nizova od 4 heksadekadske znamenke odvojene sa ':'. Na primjer, 23a5:0201:bbdf:0000:0000:8888:0000:0000 jedna je takva adresa. Kako je to puno brojeva za napisati, koriste se slijedeće dvije pokrate:

- **početne nule u svakom segmentu mogu se preskočiti**, tj. gornja se IPv6 adresa može zapisati i kao 23a5:201:bbdf:0:0:8888:0:0
- **tačno jedan niz segmenata koji su svi jednaki 0 može se izbaciti tako da se umjesto njega zapiše '::'**; drugim riječima gornja se adresa može zapisati i kao 23a5:201:bbdf::8888:0:0 ili 23a5:201:bbdf:0:0:8888::.

Svaka se IPv6 adresa može predstaviti i kao cijeli broj, u kojem je najsignifikantnija znamenka (ona koja donosi najveću vrijednost) na prvome mjestu (dakle u našem primjeru, najjača bi znamenka bila 23A5, a IP adresa predstavlja broj 47379749536320959222311704767370362880).

**Popis IP adresa** koje su u Mirkovom posjedu sastoji se od **niza raspona** IPv6 adresa koje pripadaju zlim korporacijama. **Raspon je zadan kao par IP adresa**. Na primjer, ::7f00:1 i ::7f00:100 predstavljaju niz od 256 adresa koji počinje s adresom ::7f00:1 a završava s ::7f00:100 (uključivo).

Mirko će hakirati IP adrese koje se pojavljuju u najviše raspona. Npr, neka su zadani rasponi IP adresa:

- ::7f00:1 i ::7f00:100
- ::1 i 1::
- ::7f00:50 i ::7fff:0

Mirko će napasti adrese iz raspona ::7f00:50 do ::7f00:100 jer se one pojavljuju u 3 raspona.

## Ulazni podaci

U prvom retku nalazi se prirodni broj **N** ( $1 \leq N \leq 100\,000$ ), broj raspona.

U svakom od slijedećih **N** redova nalaze se jedan **par IP adresa** koji predstavlja jedan raspon IP adresa. Sve IP adrese bit će valjane, i još će sva slova koja se pojavljuju u njima biti mala. Prva IP adresa uvijek će predstavljati manji broj od druge. **IP adrese nisu nužno jedinstvene**.

**Upozorenje:** test podaci su relativno veliki, pa obratite pažnju na to da ne koristite spore metode učitavanja.



### Izlazni podaci

U prvom i jedinom retku potrebno je ispisati traženi raspon IPv6 adresa. Ako postoji više rješenja, potrebno je ispisati ono u kojem je prva IP adresa najmanja moguća. **IPv6 adrese ispišite u potpunom obliku (bez korištenja pokrata).**

### Test primjeri

#### ULAZ :

3

::7f00:1 ::7f00:100  
::1 1::  
::7f00:50 ::7fff:0

#### IZLAZ :

0000:0000:0000:0000:0000:0000:7f00:0050 0000:0000:0000:0000:0000:0000:7f00:0100

---

#### ULAZ :

5

0000:0000:0000:0000:0000:0000:0000:0001 0000:0000:0000:0000:0000:0000:0000:0011  
0000:0000:0000:0000:0000:0000:0000:0003 0000:0000:0000:0000:0000:0000:0000:0004  
0000:0000:0000:0000:0000:0000:0000:0000 0000:0000:0000:0000:0000:0000:0000:000e  
0000:0000:0000:0000:0000:0000:0000:0009 0000:0000:0000:0000:0000:0000:0000:000d  
0000:0000:0000:0000:0000:0000:0000:0001 0000:0000:0000:0000:0000:0000:0000:0005

#### IZLAZ :

0000:0000:0000:0000:0000:0000:0000:0003 0000:0000:0000:0000:0000:0000:0000:0004

---

#### ULAZ :

5

5cc:3140::3af 1c83::16f  
e8f5:f7ff:62db:ff6b:ab80:: f000::e  
3545:3203:5000::13bc 8c7c:32ae:f1e:4fa0::3c:1b6d:1736  
4b63:e630:: b000::eb:59f9  
a489:9910:: ac9c:4000::d8c

#### IZLAZ :

4b63:e630:0000:0000:0000:0000:0000:0000 8c7c:32ae:0f1e:4fa0:0000:003c:1b6d:1736

### Bodovanje

U test primjerima vrijednim ukupno 40% bodova, sve IP adrese koristit će samo posljednja 32 bita podataka.

U test primjerima vrijednim ukupno 60% bodova, sve IP adrese koristit će samo posljednjih 64 bitova podataka.

U test primjerima vrijednim ukupno 50% bodova, sve IP adrese bit će zapisane u potpunom obliku (dakle bez korištenja gore opisanih pokrata).

U test primjerima vrijednim ukupno 40% bodova, broj **N** bit će manji od 1000.

# Mario

Autor: Ivan Katanić  
Broj bodova: 100

Vremensko ograničenje: 1 s  
Memorijsko ograničenje: 32 MB

Mirko je izmislio novu verziju igre Super Mario. U igri se nalazi **N** stupova različitih veličina označenih brojevima **1** do **N** slijeva na desno, Mario se na početku igre nalazi na najlijevijem stupu. Cilj igre je doći do posljednjeg stupa na kojem se nalazi princeza a Mario može u jednom potezu skočiti na bilo koji stup koji se nalazi desno od trenutnog i čija je visina **strogo veća** od visine trenutnog stupa. Prvi (najlijeviji) stup je **najmanje**, a posljednji (najdesniji) stup **najveće** visine.

Mirka zanima **na koliko se načina** može uspješno odigrati njegova igra te vas moli za pomoć! Kako taj broj može biti jako velik ispišite samo njegov ostatak pri dijeljenju s 10 007.

## Ulazni podaci

U prvom retku ulaza nalazi se prirodni broj **N** ( $1 \leq N \leq 100\,000$ ).

U sljedećih **N** redaka nalaze se visine stupova, od nalijevijeg do najdesnijeg. Visine su prirodni brojevi manji od 1 000 000 (milijun).

## Izlazni podaci

U prvi i jedini redak izlaza ispišite ostatak pri dijeljenju s 10 007 broja načina na koje Mario može doći do posljednjeg polja i spasiti princezu.

## Test primjeri

<b>ULAZ :</b>	<b>ULAZ :</b>	<b>ULAZ :</b>
2	3	6
5	1	4
9	2	9
	3	6
<b>IZLAZ :</b>	<b>IZLAZ :</b>	<b>IZLAZ :</b>
1	2	7
		10
		7

## Pojašnjenja

**Pojašnjenje 1. test primjera:** Jedini način da se spasi princeza jest da se skoči s prvog na drugo polje.

**Pojašnjenje 2. test primjera:** Jedan način da se spasi princeza jest da se s prvog polja odmah skoči na treće, a drugi da se s prvog skoči na drugo, pa s drugog na treće.

## Bodovanje

U test primjerima vrijedim ukupno 50% bodova, broj **N** će biti manji ili jednak 1 000.