



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

**game**

Serbia — 1.0

Ivan i Dimitrije igraju igru. Tabla za igru se sastoji od  $R$  redova obeleženih sa  $0, \dots, R - 1$ , i  $C$  kolona obeleženih sa  $0, \dots, C - 1$ . Neka je  $(P, Q)$  polje u redu  $P$  i koloni  $Q$ . Na svakom polju se nalazi nenegativni ceo broj. Na početku igre se na svim poljima nalazi broj  $0$ .

Igra se odvija na sledeći način. U svakom potezu Ivan može:

- Promeniti vrednost polja  $(P, Q)$ , u koju on želi;
- Pita Dimitrija da izračuna najveći zajednički delilac (GCD) brojeva koji se nalaze u pravougaoniku čija su suprotna temena polja  $(P, Q)$  i  $(U, V)$ .

Ivan će odigrati ukupno  $N_U + N_Q$  poteza ( $N_U$  puta će promeniti vrednost nekog polja i  $N_Q$  puta će postaviti upit Dimitriju) pre nego što mu igra dosadi i ode da spava.

Vaš zadatak je da pomognete Dimitriju da odgovori tačno na sve upite.

### Primer

Neka je  $R = 2$  i  $C = 3$ , i Ivan počinje igru sledećim potezima:

- Promena polja  $(0, 0)$  na 20;
- Promena polja  $(0, 2)$  na 15;
- Promena polja  $(1, 1)$  na 12.

20	0	15
0	12	0

Izgled table posle ovih poteza je prikazan na slici iznad. Ivan onda može pitati za GCD brojeva u sledećim pravougaonicima:

- Sa suprotnim temenima u  $(0, 0)$  i  $(0, 2)$ : Tri broja u ovom pravougaoniku su 20, 0 i 15, i njihov GCD je 5.
- Sa suprotnim temenima u  $(0, 0)$  i  $(1, 1)$ : Četiri broja u ovom pravougaoniku su 20, 0, 0 i 12, i njihov GCD je 4.

Sada recimo da Ivan odigra sledeće poteze:

- Promena polja  $(0, 1)$  na 6;
- Promena polja  $(1, 1)$  na 14.

20	6	15
0	14	0

Novi izgled table je prikazan na slici iznad. Ivan sada pita za GCD brojeva u sledećim pravougaonicima:

- Sa suprotnim temenima u  $(0, 0)$  i  $(0, 2)$ : Sada su tri broja u ovom pravougaoniku 20, 6 i 15, a njihov GCD je 1.
- Sa suprotnim temenima u  $(0, 0)$  i  $(1, 1)$ : Sada su četiri broja u ovom pravougaoniku 20, 6, 0 i 14, i njihov GCD je 2.

Ovde je Ivan ukupno napravio  $N_U = 5$  promena polja i  $N_Q = 4$  upita Dimitriju.

## Implementacija

Vi treba da priložite datoteku koja sadrži implementacije procedura `init()` i `update()` i funkcije `calculate()`, kao što je opisano u nastavku.

Da bi Vam pomogli, jedan šablon rešenja se nalazi na Vašem računaru: (`game.c`, `game.cpp` i `game.pas`) sadrže funkciju `gcd2(X, Y)` koja računa najveći zajednički delilac za data dva nenegativna cela broja  $X$  i  $Y$ . Ako je  $X = Y = 0$  onda `gcd2(X, Y)` vraća 0 kao rezultat.

Ova funkcija je dovoljno brza da se osvoje svi poeni; napomenimo da je vreme izvršavanja u najgorem slučaju proporcionalno  $\log(X + Y)$ .

**Tvoja procedura:** `init()`

C/C++ `void init(int R, int C);`

Pascal `procedure init(R, C : LongInt);`

## Opis

Vi morate priložiti implementaciju ove procedure.

Ova procedura daje početnu veličinu table i omogućava da inicijalizujete potrebne globalne promenljive i strukture podataka. Ona će biti pozvana samo jednom, pre bilo kog poziva procedura `update()` ili `calculate()`.

## Argumenti (parametri)

- `R`: Broj redova.
- `C`: Broj kolona.

## Tvoja procedura: `update()`

C/C++ `void update(int P, int Q, long long K);`

Pascal `procedure update(P, Q : LongInt; K : Int64);`

## Opis

Vi morate priložiti implementaciju ove procedure.

Ova procedura se poziva svaki put kada Ivan promeni broj na nekom polju.

## Argumenti (parametri)

- `P`: Redni broj reda table u kome se nalazi polje ( $0 \leq P \leq R - 1$ ).
- `Q`: Redni broj kolone table u kome se nalazi polje ( $0 \leq Q \leq C - 1$ ).
- `K`: Ceo broj koji predstavlja novu vrednost u tom polju ( $0 \leq K \leq 10^{18}$ ). Može biti ista kao trenutna vrednost.

## Vaša funkcija: `calculate()`

C/C++ `long long calculate(int P, int Q, int U, int V);`

Pascal `function calculate(P, Q, U, V : LongInt) : Int64;`

## Opis

Vi morate priložiti implementaciju ove procedure.

Ova funkcija treba da računa najveći zajednički delilac svih brojeva u pravougaoniku čija su dva suprotna temena  $(P, Q)$  i  $(U, V)$ .

Ako se u svim ćelijama pravougaonika nalazi vrednost nula, funkcija vraća nula, kao rezultat.

### Argumenti (parametri)

- $P$ : Red u kome se nalazi polje koje predstavlja gornje levo teme pravougaonika ( $0 \leq P \leq R - 1$ ).
- $Q$ : Kolona u kojoj se nalazi polje koje predstavlja gornje desno teme pravougaonika ( $0 \leq Q \leq C - 1$ ).
- $U$ : Vrsta u kojoj se nalazi polje koje predstavlja donje levo teme pravougaonika ( $P \leq U \leq R - 1$ ).
- $V$ : Kolona u kojoj se nalazi polje koje predstavlja donje desno teme pravougaonika ( $Q \leq V \leq C - 1$ ).
- *Vraća*: Najveći zajednički delilac svih celih brojeva koji se nalaze u pravougaoniku, ili 0 ako sva polja pravougaonika sadrže broj nula.

---

## Primer izvršavanja

Sledeće izvršavanje opisuje gornji primer:

Poziv funkcije	Šta vraća
<code>init(2, 3)</code>	
<code>update(0, 0, 20)</code>	
<code>update(0, 2, 15)</code>	
<code>update(1, 1, 12)</code>	
<code>calculate(0, 0, 0, 2)</code>	5
<code>calculate(0, 0, 1, 1)</code>	4
<code>update(0, 1, 6)</code>	
<code>update(1, 1, 14)</code>	
<code>calculate(0, 0, 0, 2)</code>	1
<code>calculate(0, 0, 1, 1)</code>	2

---

## Ograničenja

- Vremensko ograničenje: Pogledati englesku verziju zadatka
- Memorijsko ograničenje: Pogledati englesku verziju zadatka
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ , gde je  $K$  bilo koja od vrednosti koju Ivan stavlja na polje table.

---

## Podzadaci

Pogledati englesku verziju zadatka!

Subtask	Points	R	C	$N_U$	$N_Q$	Time limit	Memory limit

---

## Lokalno testiranje

Primer grejdera koji se nalazi na Vašem računaru čita ulazne podatke iz datoteke `game.in`. Ta datoteka ima sledeći format:

- red 1: `R C N`
- sledećih  $N$  redova: po jedna radnja u svakom redu, u redosledu u kome se dešavaju.

Svaki red sa radnjom ima jedan od sledećih formata:

- da bi označili `update(P, Q, K) : 1 P Q K`
- da bi označili `calculate(P, Q, U, V) : 2 P Q U V`

Na primer, za niz radnji iz prethodnog primera treba sastaviti datoteku sa sledećim sadržajem:

```
2 3 9
1 0 0 20
1 0 2 15
1 1 1 12
2 0 0 0 2
2 0 0 1 1
1 0 1 6
1 1 1 14
2 0 0 0 2
2 0 0 1 1
```

---

## Napomene vezane za programski jezik

**C/C++** Vi treba da dodate red `#include "game.h"`.

**Pascal** Vi treba da definišete `unit Game`. Svi nizovi su numerisani počev od `0` (ne od `1`).

Pošto brojevi na tabli mogu da budu vrlo veliki, C/C++ programerima se preporučuje da koriste `long long` tip, a Pascal programerima se preporučuje da koriste `Int64` tip.