



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

# game

G— 1.0

ბაზამ და შაზამ ახალი თამაში მოიგონეს. სათამაშო დაფა წარმოადგენს უჯრედოვან ბადეს  $R$  სტრიქონისა (გადანომრილია  $0, \dots, R - 1$ ) და  $C$  სვეტისაგან (გადანომრილია  $0, \dots, C - 1$ ). განვსაზღვროთ  $(P, Q)$ , როგორც  $P$  სტრიქონისა და  $Q$  სვეტის გადაკვეთაზე არსებული უჯრედი. ყოველი უჯრედი შეიცავს არაუარყოფით მთელ რიცხვს და თამაშის დაწყებისას ყველა ამ რიცხვის მნიშვნელობა არის 0.

თამაში მიმდინარეობს შემდეგნაირად: ნებისმიერ დროს ბაზამ შეიძლება მოითხოვოს ორიდან ერთ-ერთი ოპერაცია:

- განაახლოს უჯრედი  $(P, Q)$ , რაც გამოიწვევს მასში ახალი მნიშვნელობის ჩაწერას;
- დაავალოს შაზას, უდიდესი საერთო გამყოფის (უსგ) გამოთვლა იმ მართკუთხა არის უჯრედებისათვის, რომლის მოპირდაპირე კუთხეებია  $(P, Q)$  და  $(U, V)$  მათი ჩათვლით.

ბაზა შეასრულებს  $N_U + N_Q$  მოქმედებას (განაახლებს უჯრედებს  $N_U$ -ჯერ და დასვამს  $N_Q$  შეკითხვას) და წავა კრიკეტის სათამაშოდ.

თქვენი ამოცანაა დაეხმაროთ შაზას სწორი პასუხების მოძებნაში.

### მაგალითი

დავუშვათ  $R = 2$  და  $C = 3$ , და ბაზა აკეთებს შემდეგ განახლებებს:

- ანახლებს  $(0, 0)$  უჯრედს რიცხვით 20;
- ანახლებს  $(0, 2)$  უჯრედს რიცხვით 15;
- ანახლებს  $(1, 1)$  უჯრედს რიცხვით 12.

20	0	15
0	12	0

მიღებული ბადე ნაჩვენებია ნახაზზე ზემოთ. ბაზას შეუძლია მოითხოვოს უსგ-ს გამოთვლა შემდეგ მართკუთხა არეებზე:

- მართკუთხედზე მოპირდაპირე კუთხეებით  $(0, 0)$  და  $(0, 2)$ : ამ მართკუთხედში წერია სამი მთელი რიცხვი 20, 0 და 15, და მათი უსგ არის 5.
- მართკუთხედზე მოპირდაპირე კუთხეებით  $(0, 0)$  და  $(1, 1)$ : ამ მართკუთხედში წერია ოთხი მთელი რიცხვი 20, 0, 0 და 12, და მათი უსგ არის 4.

ახლა დავუშვათ ბაზა აკეთებს შემდეგ განახლებებს:

- ანახლებს  $(0, 1)$  უჯრედს მნიშვნელობით 6;
- ანახლებს  $(1, 1)$  უჯრედს მნიშვნელობით 14.

20	6	15
0	14	0

ახალი ბადე ნაჩვენებია ნახაზზე ზემოთ. ბაზამ შეიძლება დასვას შეკითხვა უსგ-ს გამოსათვლელად შემდეგ მართკუთხა არეებზე:

- მართკუთხედზე მოპირდაპირე კუთხეებით  $(0, 0)$  და  $(0, 2)$ : მართკუთხედში ამჯერად წერია სამი მთელი რიცხვი 20, 6 და 15, და მათი უსგ არის 1.
- მართკუთხედზე მოპირდაპირე კუთხეებით  $(0, 0)$  და  $(1, 1)$ : მართკუთხედში ამჯერად წერია ოთხი მთელი რიცხვი 20, 6, 0 და 14, და მათი უსგ არის 2.

ამ მომენტისათვის ბაზამ სულ შეასრულა  $N = 9$  მოქმედება (ხუთი განახლება პლუს ოთხი შეკითხვა).

## იმპლემენტაცია

თქვენი გაგზავნილი ფაილი უნდა ახდენდეს `init()` და `update()` პროცედურებისა და `calculate()` ფუნქციის იმპლემენტაციას ქვემოთ აღწერილი წესით.

თქვენს დასახმარებლად კომპიუტერში არსებული ამოხსნის შაბლონები (`game.c`, `game.cpp` და `game.pas`) შეიცავენ ფუნქციას `gcd2(X, Y)` უდიდესი საერთო გამყოფის გამოსათვლელად ორი მთელი არაუარყოფითი `X` და `Y` რიცხვისათვის. თუ `X = Y = 0`, მაშინ `gcd2(X, Y)` დააბრუნებს `0`-ს.

ეს ფუნქცია საკმარისად სწრაფია სრული ამოხსნის მისაღებად. კერძოდ, უარეს შემთხვევაში მისი მუშაობის დრო პროპორციულია `log(X + Y)`-ის.

თქვენი პროცედურა: **init()**

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

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

## აღწერა

თქვენმა ამოხსნამ უნდა მოახდინოს ამ პროცედურის იმპლემენტაცია.

პროცედურა იძლევა ბადის ზომებს და ინიციალიზაციას უკეთებს ყველა გლობალურ ცვლადს და მონაცემთა სტრუქტურას. იგი გამოძახებული უნდა იქნას მხოლოდ ერთხელ `update()` და `calculate()` ფუნქციების გამოძახებამდე.

## პარამეტრები

- `R`: სტრიქონების რაოდენობა.
- `C`: სვეტების რაოდენობა.

თქვენი პროცედურა: **update()**

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

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

## აღწერა

თქვენმა ამოხსნამ უნდა მოახდინოს ამ პროცედურის იმპლემენტაცია.

ეს პროცედურა გამოიძახება მაშინ, როცა ბაზა ანიჭებს მნიშვნელობას ბადის რომელიმე უჯრედს.

## პარამეტრები

- $P$ : ბადის სტრიქონი ( $0 \leq P \leq R - 1$ ).
- $Q$ : ბადის სვეტი ( $0 \leq Q \leq C - 1$ ).
- $K$ : უჯრედის ახალი მნიშვნელობა ( $0 \leq K \leq 10^{18}$ ). შესაძლოა იგი ძველ მნიშვნელობასაც უდრიდეს.

თქვენი ფუნქცია: **calculate()**

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

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

## აღწერა

თქვენმა ამოხსნამ უნდა მოახდინოს ამ პროცედურის იმპლემენტაცია.

ამ ფუნქციამ უნდა გამოთვალოს უდიდესი საერთო გამყოფი ყველა რიცხვისათვის ( $P, Q$ ) და ( $U, V$ ) მოპირდაპირე კუთხეებით განსაზღვრულ მართკუთხედში. შევნიშნოთ, რომ უჯრედები ( $P, Q$ ) და ( $U, V$ ) ასევე მიეკუთვნებიან მართკუთხედს.

თუკი ყველა რიცხვი მოცემულ მართკუთხედში 0-ის ტოლია, მაშინ ამ ფუნქციამაც 0 უნდა დააბრუნოს.

## პარამეტრები

- $P$ : მართკუთხედის ზედა-მარცხენა უჯრედის სტრიქონი ( $0 \leq P \leq R - 1$ ).
- $Q$ : მართკუთხედის ზედა-მარცხენა უჯრედის სვეტი ( $0 \leq Q \leq C - 1$ ).
- $U$ : მართკუთხედის ქვედა-მარჯვენა უჯრედის სტრიქონი ( $P \leq U \leq R - 1$ ).
- $V$ : მართკუთხედის ქვედა-მარჯვენა უჯრედის სვეტი ( $Q \leq V \leq C - 1$ ).
- *Returns*: ამ მართკუთხედში შემავალი ყველა მთელი რიცხვი უსგ, ან 0, თუ ყველა ეს რიცხვი 0-ის ტოლია.

## სეანსის ნიმუში

შემდეგი სეანსი აღწერს ზემოთ მოყვანილ მაგალითს:

ფუნქციის გამოძახება	Returns
<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

## შეზღუდვები

- დროის ლიმიტი: 2 წამი
- მეხსიერების ლიმიტი: 512 MiB
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ , სადაც  $K$  არის ბაზას მიერ ბადის შესაბამის უჯრედში მოთავსებული რიცხვი.

## ქვეამოცანები

შეზღუდვები ქვეამოცანებისათვის იხილეთ ამოცანის ინგლისურ ვერსიაში

ქვეამოცანა	ქულა	R	C	$N_U$	$N_D$
1	10				
2	28				
3	26				
4	36				

## ექსპერიმენტირება

თქვენი კომპიუტერიდან შესატანი მონაცემები გრადერმა უნდა წაიკითხოს ფაილით `game.in`. ამ ფაილს უნდა ჰქონდეს შემდეგი ფორმატი:

- სტრიქონი 1: `R C N`
- მომდევნო `N` ცალი სტრიქონი: თითო მოქმედება თითო სტრიქონში იმ თანმიმდევრობით, რომლითაც ეს მოქმედებები მოხდა.

თითოეული მოქმედებისათვის სტრიქონს უნდა ჰქონდეს შემდეგ ორი ფორმატიდან ერთ-ერთი :

- თუ მითითებულია `update(P, Q, K) : 1 P Q K`
- თუ მითითებულია `calculate(P, Q, U, V) : 2 P Q U V`

მაგალითად, ზემოთ მოტანილი მაგალითი წარმოდგენილი უნდა იყოს შემდეგი ფორმატით:

```
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
```

## შენიშვნები პროგრამირებული ენებისათვის

**C/C++** თქვენ უნდა გამოიყენოთ `#include "game.h"`.

**Pascal** You must define the `unit Game`. All arrays are numbered beginning at `0` (not `1`).

რადგან რიცხვები ბადის უჯრედებში ძალიან დიდი შეიძლება იყოს, C/C++-ის მომხმარებლებმა უმჯობესია გამოიყენონ `long long` ტიპი, ხოლო Pascal-ის მომხმარებლებმა `Int64` ტიპი.