



## International Olympiad in Informatics 2013

6-13 July 2013  
Brisbane, Australia  
Day 2 tasks

# hra

Slovak — 1.1

Usama a Mišoľ hrajú hru na mriežke (štvorčekovanom papieri) s  $R$  riadkami očíslovanými  $0, \dots, R - 1$  a  $C$  stĺpcami očíslovanými  $0, \dots, C - 1$ . Políčko v riadku  $p$  a stĺpci  $q$  označme  $(p, q)$ . Každé políčko obsahuje nezáporné celé číslo, na začiatku hry sú tam samé nuly.

Hra funguje nasledovne. Usama môže v ľubovoľnej chvíli urobiť jeden z nasledovných ťahov:

- zmeniť hodnotu políčka  $(p, q)$
- opýtať sa Mišoľa na hodnotu najväčšieho spoločného deliteľa (NSD) všetkých čísel v obdĺžnikovej oblasti určenej protifaľými rohmi  $(p, q)$  a  $(u, v)$  (vrátane).

Usama vykoná  $N_U + N_Q$  ťahov ( $N_U$  krát zmení hodnotu a  $N_Q$  krát položí otázku), kým sa unaví a pôjde si vonku zahrať frisbee.

Vašou úlohou je správne odpovedať na všetky otázky za Mišoľa.

## Príklad

Nech  $R = 2$  a  $C = 3$ , Usama začne s nasledovnými ťahmi:

- zmena hodnoty políčka  $(0, 0)$  na 20
- zmena hodnoty políčka  $(0, 2)$  na 15
- zmena hodnoty políčka  $(1, 1)$  na 12

20	0	15
0	12	0

Výsledná mriežka je na obrázku vyššie. Usama sa môže potom opýtať na NSD v nasledujúcich odľžníkoch:

- protiľahlé rohy  $(0, 0)$  a  $(0, 2)$  : tri čísla 20, 0 a 15, ich NSD je 5.
- protiľahlé rohy  $(0, 0)$  a  $(1, 1)$  : štyri čísla 20, 0, 0 a 12, ich NSD je 4.

Usama následne urobí ďalšie dve zmeny:

- zmena hodnoty políčka  $(0, 1)$  na 6
- zmena hodnoty políčka  $(1, 1)$  na 14

20	6	15
0	14	0

Zmenená mriežka je zobrazená na obrázku vyššie. Usama sa môže potom opýtať na NSD v ďalších odľžníkoch:

- protiľahlé rohy  $(0, 0)$  a  $(0, 2)$  : tri čísla 20, 6 a 15, ich NSD je 1.
- protiľahlé rohy  $(0, 0)$  a  $(1, 1)$  : štyri čísla 20, 6, 0 a 12, ich NSD je 2.

Usama vykonal 9 ťahov,  $N_u = 5$  zmien a  $N_Q = 4$  otázky.

---

## Implementácia

Odovzdajte súbor, v ktorom implementujete procedúry `init()` a `update()` a funkciu `calculate()`, ktoré sú popísané nižšie.

Ako pomôcka pre vás, predloha riešenia na vašom počítači (`game.c`, `game.cpp` a `game.pas`) zahŕňa funkciu `gcd2(X, Y)` na výpočet NSD dvoch nezáporných celých čísel  $X$  a  $Y$ . V prípade  $X = Y = 0$  funkcia `gcd2(X, Y)` vráti tiež hodnotu 0.

Táto funkcia je dostatočne rýchla, aby s ňou bolo možné dosiahnuť plný počet bodov, konkrétne beží v čase  $O(\log(X + Y))$ .

### vaša procedúra: `init()`

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

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

#### Popis

Musíte implementovať túto procedúru.

Táto procedúra vám zadá rozmery mriežky a umožní vám inicializovať ľubovoľné globálne premenné a dátové štruktúry. Zavolaná bude práve raz, pred ľubovoľným zavolaním `update()` alebo `calculate()`.

#### Parametre

- `R` : počet riadkov.
- `C` : počet stĺpcov.

### vaša procedúra: `update()`

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

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

#### Popis

Musíte implementovať túto procedúru.

Táto procedúra bude zavolaná, keď Usama zmení hodnotu jedného políčka.

#### Parametre

- `P` : číslo riadku políčka ( $0 \leq P \leq R - 1$ ).
- `Q` : číslo stĺpca políčka ( $0 \leq Q \leq C - 1$ ).
- `K` : nová hodnota políčka ( $0 \leq K \leq 10^{18}$ ). Môže byť rovnaká ako aktuálna hodnota.

## Vaša funkcia: `calculate()`

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

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

### Popis

Musíte implementovať túto procedúru.

Funkcia má vypočítať najväčšieho spoločného deliteľa všetkých čísel, ktoré sa nachádzajú v obdĺžniku určeného protiľahlými rohmi `(P, Q)` a `(U, V)`. Tento rozsah je vrátane, a teda políčka `(P, Q)` a `(U, V)` sa nachádzajú v obdĺžniku.

Ak sú všetky čísla v tomto obdĺžniku s hodnotou nula, funkcia by mala vrátiť nulu.

### Parametre

- `P`: Riadok horného ľavého políčka mriežky v obdĺžniku ( $0 \leq P \leq R - 1$ ).
- `Q`: Stĺpec horného ľavého políčka mriežky v obdĺžniku ( $0 \leq Q \leq C - 1$ ).
- `U`: Riadok dolného pravého políčka mriežky v obdĺžniku ( $P \leq U \leq R - 1$ ).
- `V`: Stĺpec dolného pravého políčka mriežky v obdĺžniku ( $Q \leq V \leq C - 1$ ).
- *Vracia*: NSD všetkých čísel v danom obdĺžniku alebo `0`, ak sú všetky tieto čísla s hodnotou nula.

---

## Ukázkový beh

Nasledujúci beh popisuje príklad spomínaný vyššie:

volanie funkcie/procedúry	návratová hodnota
<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

---

## Obmedzenia

- Časový limit: viď podúlohy
- Pamäťový limit: viď podúlohy
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ , kde  $K$  je ľubovoľné celé číslo, ktoré Usama zmení v mriežke.

---

## Podúlohy

Hodnoty sú uvedené v anglickej verzii.

Podúloha	Body	R	C	$N_U$	$N_Q$	Časový limit	Pamäťový limit

---

## Experimentácia

Ukážkový testovač na vašom počítači bude čítať vstup zo súboru `game.in`, ktorý musí mať nasledovný formát:

- riadok 1: `R C N`
- nasledovných `N` riadkov: jeden ťah na riadok v poradí v akom nastávajú.

Riadok s každým ťahom musí byť v jednom z nasledujúcich formátov:

- zmena hodnoty `update(P, Q, K) : 1 P Q K`
- otázka `calculate(P, Q, U, V) : 2 P Q U V`

Napríklad príklad uvedený vyššie, by bol zapísaný nasledovne:

```
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
```

---

## Poznámky k programovacím jazykom

Keďže celé čísla v mriežke môžu byť veľmi veľké, je odporúčané používať dátové typy s dostatočne veľkým rozsahom.

- C/C++ Musíte použiť `#include "game.h"`. Používajte celočíselný typ `long long`.
- Pascal Musíte definovať `unit Game`. Všetky polia sú číslované od `0` (nie od `1`). Používajte celočíselný typ `Int64`.