



## International Olympiad in Informatics 2013

6-13 July 2013

Brisbane, Australia

Day 2 tasks

**game**

Greek — 1.0

Ο Bazza και ο Shazza παίζουν ένα παιχνίδι. Ο χώρος του παιχνιδιού είναι ένα πλέγμα από κελιά, με  $R$  γραμμές αριθμημένες  $0, \dots, R - 1$ , και  $C$  στήλες αριθμημένες  $0, \dots, C - 1$ . Έστω ότι το  $(p, q)$  συμβολίζει το κελί στη γραμμή  $p$  και την στήλη  $q$ . Κάθε κελί περιέχει έναν μη αρνητικό ακέραιο, και στην αρχή του παιχνιδιού όλοι αυτοί οι ακέραιοι είναι μηδέν.

Το παιχνίδι εξελίσσεται ως εξής. Οποιαδήποτε στιγμή, ο Bazza μπορεί:

- να ενημερώσει ένα κελί  $(p, q)$ , αλλάζοντας τον ακέραιο που περιέχει.
- να ζητήσει από τον Shazza να υπολογίσει τον μέγιστο κοινό διαιρέτη (GCD) όλων των ακεραίων μέσα σε ένα ορθογώνιο block από κελιά, με εκ διαμέτρου αντίθετες κορυφές  $(p, q)$  και  $(u, v)$  συμπεριλαμβανομένων.

Ο Bazza θα εκτελέσει το πολύ  $N$  ενέργειες (ενημέρωση κελιών και/ή υποβολή ερωτήσεων) πριν βαρεθεί και πάει στο ύπαιθρο για να παίξει cricket.

Το πρόβλημά σας είναι να βρείτε τις σωστές απαντήσεις.

## Παράδειγμα

Έστω  $R = 2$  και  $C = 3$ , και ο Bazza ξεκινά με τις παρακάτω ενημερώσεις:

- Ενημέρωση του κελιού  $(0, 0)$  σε 20;
- Ενημέρωση του κελιού  $(0, 2)$  σε 15;
- Ενημέρωση του κελιού  $(1, 1)$  σε 12.

|    |    |    |
|----|----|----|
| 20 | 0  | 15 |
| 0  | 12 | 0  |

Στην παραπάνω εικόνα φαίνεται το πλέγμα που προκύπτει. Ο Bazza στη συνέχεια μπορεί να ρωτήσει για τον μέγιστο κοινό διαιρέτη (ΜΚΔ) στα παρακάτω ορθογώνια:

- Αντιδιαμετρικές κορυφές  $(0, 0)$  και  $(0, 2)$ : Οι τρεις ακέραιοι σε αυτά τα ορθογώνια είναι 20, 0 και 15, και ο ΜΚΔ τους είναι 5.
- Αντιδιαμετρικές κορυφές  $(0, 0)$  και  $(1, 1)$ : Οι τέσσερις ακέραιοι σε αυτό το ορθογώνιο είναι 20, 0, 0 και 12, και ο ΜΚΔ τους είναι 4.

Υποθέστε τώρα ότι ο Bazza εκτελεί τις παρακάτω ενημερώσεις:

- Ενημέρωση του κελιού  $(0, 1)$  σε 6;
- Ενημέρωση του κελιού  $(1, 1)$  σε 14.

|    |    |    |
|----|----|----|
| 20 | 6  | 15 |
| 0  | 14 | 0  |

Η παραπάνω εικόνα δείχνει το νέο πλέγμα. Στη συνέχεια ο Bazza μπορεί να ρωτήσει ξανά για τον ΜΚΔ στα παρακάτω ορθογώνια:

- Αντιδιαμετρικές κορυφές  $(0, 0)$  και  $(0, 2)$ : Τώρα οι τρεις ακέραιοι σε αυτό το ορθογώνιο είναι 20, 6 και 15, και ο ΜΚΔ τους είναι 1.
- Αντιδιαμετρικές κορυφές  $(0, 0)$  και  $(1, 1)$ : Οι τέσσερις ακέραιοι σε αυτό το ορθογώνιο είναι τώρα 20, 6, 0 και 14, και ο ΜΚΔ τους είναι 2.

Εδώ ο Bazza έχει εκτελέσει  $N = 9$  πράξεις συνολικά (πέντε ενημερώσεις συν τέσσερις ερωτήσεις).

## Υλοποίηση

θα πρέπει να υποβάλλετε ένα αρχείο που θα υλοποιεί τις διαδικασίες `init()` και `update()` και την συνάρτηση `calculate()`, όπως περιγράφονται παρακάτω.

Για να σας βοηθήσουμε, οι υποδειγματικές λύσεις που θα παράγετε στον υπολογιστή σας (`game.c`, `game.cpp` και `game.pas`) κάθε μια θα περιλαμβάνει μια συνάρτηση `gcd2(X, Y)` που θα υπολογίζει τον μέγιστο κοινό διαιρέτη μεταξύ 2 μη αρνητικών ακεραίων  $X$  και  $Y$ . Εάν  $X = Y = 0$  τότε `gcd2(X, Y)` θα επιστρέψει 0 επίσης.

Αυτή η συνάρτηση αρκεί για να πετύχετε πλήρες σκόρ εάν πιο συγκεκριμένα, ο χρόνος εκτέλεσης είναι το χειρότερο ανάλογος με  $\log(X + Y)$ .

**Η διαδικασία: `init()`**

Template:Ενδεικτικά

## Περιγραφή

Η υποβολή σας θα πρέπει να υλοποιεί την διαδικασία:

Η διαδικασία επιστρέφει το αρχικό μέγεθος του πλέγματος, και σου επιτρέπει να αρχικοποιήσεις όλες τις καθολικές μεταβλητές και τα δεδομένα. Θα κληθεί μόνο μια φορά και πριν από οποιαδήποτε κλήση `update()` ή `calculate()`.

## Παράμετροι

- `R`: Ο αριθμός των γραμμών.
- `C`: Ο αριθμός των στηλών.

**Η διαδικασία: `update()`**

Template:Ενδεικτικά

## Περιγραφή

Η υποβολή σας θα πρέπει να υλοποιεί την διαδικασία:

Η διαδικασία θα καλείται όταν η Bazza εισάγει ένα νούμερο σε κάποιο κελί.

## Παράμετροι

- `P`: Η γραμμή του κελιού στο πλέγμα ( $0 \leq P \leq R - 1$ ).
- `Q`: Η στήλη του κελιού στο πλέγμα ( $0 \leq Q \leq C - 1$ ).
- `K`: Ο καινούργιος ακέραιος στο κελί ( $0 \leq K \leq 10^{18}$ ). που μπορεί να είναι και ο ίδιος με την τιμή που υπάρχει ήδη σε αυτό.

## Η συνάρτηση: `putaway()`

Template: Ενδεικτικά

### Περιγραφή

Η υποβολή σας θα πρέπει να υλοποιεί την συνάρτηση:

Αυτή η συνάρτηση θα πρέπει να υπολογίζει τον μέγιστο κοινό διαιρέτη από όλους τους ακέραιους του στο πολύγωνο με εκ διαμέτρου αντίθετες κορυφές  $(P, Q)$  και  $(U, V)$ . Το διάστημα αυτό είναι κλειστό, π.χ., τα κελιά  $(P, Q)$  and  $(U, V)$  περιλαμβάνονται στο πολύγωνο.

Εαν όλοι οι ακέραιοι στο πολύγωνο είναι 0, τότε αυτή η συνάρτηση θα πρέπει να επιστρέψει 0 επίσης.

### Παράμετροι

- $P$ : Η γραμμή του επάνω αριστερά κελιού του πολυγώνου ( $0 \leq P \leq R - 1$ ).
- $Q$ : Η στήλη του επάνω αριστερά κελιού του πολυγώνου ( $0 \leq Q \leq C - 1$ ).
- $U$ : Η γραμμή του κάτω δεξιά κελιού του πολυγώνου ( $P \leq U \leq R - 1$ ).
- $V$ : Η στήλη του κάτω δεξιά κελιού του πολυγώνου ( $Q \leq V \leq C - 1$ ).
- *Επιστρέφει*: Τον ΜΚΔ όλων των ακεραίων στο πολύγωνο, ή 0 εάν όλοι αυτοί ακέραιοι στο πολύγωνο είναι 0.

## Παράδειγμα

Παρακάτω περιγράφεται το παράδειγμα:

| κλήση συνάρτησης                   | Επιστροφή |
|------------------------------------|-----------|
| <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         |

## Περιορισμοί

- Οριο χρόνου: Δες τα υποπροβλήματα.
- Οριο μνήμης: Δες τα υποπροβλήματα.
- $1 \leq R, C \leq 10^9$
- $0 \leq K \leq 10^{18}$ , όπου  $K$  είναι οποιοσδήποτε ακέραιος όπου η Bazza τοποθετεί σε ένα κελί του πλέγματος.

## Υποπροβλήματα

Δες την αγγλική έκδοση για τις παραμέτρους στα υποπροβλήματα.

| Υποπροβλήματα | Μονάδες | R | C | $N_U$ | $N_Q$ | Οριο χρόνου | Οριο μνήμης |
|---------------|---------|---|---|-------|-------|-------------|-------------|
|               |         |   |   |       |       |             |             |
|               |         |   |   |       |       |             |             |
|               |         |   |   |       |       |             |             |
|               |         |   |   |       |       |             |             |

## Πειραματισμός

Το πρόγραμμα αξιολόγησης (grader) στον υπολογιστή σας θα πάρει εισαγωγή από το αρχείο `game.in`. Αυτό το αρχείο θα πρέπει να συμφωνεί με την ακόλουθη μορφή:

- γραμμή 1: `R C N`
- επόμενες `N` γραμμές: ένα γεγονός σε κάθε γραμμή, και ανάλογα με την σειρά που καθένα προκύπτει.

Η γραμμή για κάθε γεγονός θα πρέπει να είναι σε μια από τις ακόλουθες μορφές:

- η υπόδειξη `update(P, Q, K) : 1 P Q K`
- η υπόδειξη `calculate(P, Q, U, V) : 2 P Q U V`

Έτσι, το πιο πάνω παράδειγμα θα πρέπει να παράξει την παρακάτω μορφή (output) :

```
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
```

## Οδηγίες για τη γλώσσα

**C/C++** Στον κώδικα χρησιμοποιήστε την οδηγία `#include "game.h"`.

**Pascal** Στον κώδικα θα πρέπει να οριστεί `unit Game`. Η αρίθμηση των στοιχείων των πινάκων ξεκινά από `0` (όχι `1`).

Επειδή οι ακέραιοι στα κελιά του πλέγματος μπορεί να είναι ιδιαίτερα μεγάλοι, οι C/C++ χρήστες καλό είναι να χρησιμοποιήσουν `long long` τύπο, και οι χρήστες της Pascal καλό είναι να χρησιμοποιήσουν `Int64` τύπο.