

ENIGMA 0x00

18 Δεκεμβρίου 2022

Προβλήματα

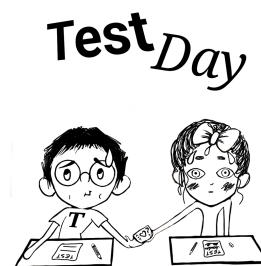
Πρόβλημα	Χρονικό Όριο	Όριο Μνήμης	Βαθμολογία
Αντιγραφή	1 sec	16MB	10
Τα πλεκτά της γιαγιάς	1 sec	16MB	25
Μαγικά τετράγωνα	1 sec	16MB	25
Μάντεψε και Κέρδισε!	1 sec	16MB	40
Σύνολο			100

Αντιγραφή

Είναι Κυριακή πρωί και ο Τοτός έχει αγχωθεί:

«Σε λίγο ξεκινάει ο ENIGMA και δε θυμάμαι καμία γλώσσα προγραμματισμού!»

Ευτυχώς η φίλη του η Αννούλα, που είναι η καλύτερη στον προγραμματισμό, κάθεται δίπλα του και σκοπεύει να τον βοηθήσει :).



Ο διαγωνισμός ξεκινάει και μέσα σε ελάχιστα λεπτά, η Αννούλα λύνει το πρώτο θέμα, το γράφει σε ένα σκονάκι και του το δίνει να το αντιγράψει.

Του γράφει μάλιστα σε δύο πασίγνωστες γλώσσες προγραμματισμού, τη **C** και την **Python**, ώστε να μπορεί να επιλέξει όποια τον βολεύει καλύτερα!

Ο Τοτός ξεκινά αμέσως την αντιγραφή, αλλά ευτυχώς, λίγο πριν πατήσει το κουμπί της υποβολής, αντιλαμβάνεται ότι κάτι δεν πάει καλά.

Τα θέματα είναι διαφορετικά για κάθε παιδί!

Σε εκείνον ζητήθηκε το εξής:

“Θέλουμε να γράψουμε ένα πρόγραμμα που θα διαβάζει από την είσοδο `STDIN` τρεις θετικούς ακέραιους αριθμούς N , S και T , και θα εκτυπώνει στην έξοδο `STDOUT` πλήθη χαρακτήρων '@' ή διαχωριστικών ' ' και '\n'.

- Ο N ($1 \leq N \leq 10$) συμβολίζει το **πλήθος** των χαρτακίων.
- Ο S ($1 \leq S \leq 10$) συμβολίζει το **μέγεθος** των χαρτακίων.
- Ο T ($0 \leq T \leq 1$) συμβολίζει **τον τρόπο τοποθέτησης** των χαρτακίων, `0` για οριζόντια, `1` για κάθετη.

Προσοχή! Κάθε γραμμή εισόδου ή εξόδου (θα πρέπει να) τερματίζει με το χαρακτήρα αλλαγής γραμμής '\n'.”

Παραδείγματα

1ο

`STDIN`

```
4 3 0
```

`STDOUT`

```
000 000 000 000
000 000 000 000
000 000 000 000
```

2ο

STDIN

```
2 4 1
```

STDOUT

```
0000  
0000  
0000  
0000  
  
0000  
0000  
0000  
0000
```

Ακολουθεί το σκονάκι ;)

C

```
#include <stdio.h>  
  
// read_number(), STDIN: "123 "  
// |123 : 0  
// 1|23 : 0 * 10 + 1 = 1  
// 12|3 : 1 * 10 + 2 = 12  
// 123| : 12 * 10 + 3 = 123  
// 123 | : 123  
int read_number() {  
    int number = 0;  
  
    char c = getchar();  
    // Xtizo ton arithmo  
    while((c >= '0') && (c <= '9')) {  
        int digit = c - '0';  
        number = number * 10 + digit;  
        c = getchar();  
    }  
  
    return number;  
}  
  
// squares_horizontal(2, 3)  
// 000 000  
// 000 000  
// 000 000
```

```
void squares_horizontal(int cnt, int size) {
    // Metritis grammon othonis
    for(int i = 0; i < size; i++) {

        // Metritis tegragonon
        for(int j = 0; j < cnt; j++) {

            // Metritis xaraktiron
            for(int k = 0; k < size; k++) {
                putchar('@');
            }

            putchar(' ');
        }

        putchar('\n');
    }
}

// squares_vertical(2, 3)
// @@@
// @@@
// @@@
//
// @@@
// @@@
// @@@

void squares_vertical(int cnt, int size) {
    // Metritis tegragonon
    for(int j = 0; j < cnt; j++) {

        // Metritis grammon othonis
        for(int i = 0; i < size; i++) {

            // Metritis xaraktiron
            for(int k = 0; k < size; k++) {
                putchar('@');
            }

            putchar('\n');
        }

        putchar('\n');
    }
}
```

```
// Η εκτέλεση του προγράμματος κσεκίνα εδο!  
int main() {  
    int cnt = read_number();  
    int size = read_number();  
  
    squares_horizontal(cnt, size);  
    squares_vertical(cnt, size);  
  
    return 0;  
}
```

Python

```
from sys import stdin, stdout, exit  
  
# read_number(), STDIN: "123 "  
# |123 : 0  
# 1|23 : 0 * 10 + 1 = 1  
# 12|3 : 1 * 10 + 2 = 12  
# 123| : 12 * 10 + 3 = 123  
# 123 | : 123  
def read_number():  
    number = 0  
  
    c = stdin.read(1)  
    # Xtizo ton arithmo  
    while((c >= '0') and (c <= '9')):  
        digit = int(c)  
        number = number * 10 + digit  
        c = stdin.read(1)  
  
    return number  
  
# squares_horizontal(2, 3)  
# @@@ @@@  
# @@@ @@@  
# @@@ @@@  
def squares_horizontal(cnt, size):  
    # Metritis grammon othonis  
    for i in range(0, size, 1):  
  
        # Metritis tegrakonon  
        for j in range(0, cnt, 1):  
  
            # Metritis xaraktiron  
            for k in range(0, size, 1):  
                stdout.write('@')
```

```
        stdout.write(' ')

        stdout.write('\n')

# squares_vertical(2, 3)
# @@@
# @@@
# @@@
#
# @@@
# @@@
# @@@
def squares_vertical(cnt, size):
    # Metritis tegrakonon
    for j in range(0, cnt, 1):

        # Metritis grammon othonis
        for i in range(0, size, 1):

            # Metritis xaraktiron
            for k in range(0, size, 1):
                stdout.write('@')

            stdout.write('\n')

        stdout.write('\n')

# H ektelesi tou programmatos ksekina edo!
if __name__ == "__main__":
    cnt = read_number()
    size = read_number()

    squares_horizontal(cnt, size)
    squares_vertical(cnt, size)

    exit(0)
```

Τα πλεκτά της γιαγιάς

Ο Τοτός αγαπάει πολύ τη γιαγιά του.

Προσπαθεί πάντα να περνάει όσο περισσότερο χρόνο μαζί της γίνεται, ειδικά τα Χριστούγεννα.

Κάθε χρόνο τέτοια εποχή, η γιαγιά του ετοιμάζει τα δώρα για τα παιδιά, τα εγγόνια και τα δισέγγονά της, που δεν είναι άλλα από χειροποίητα πλεκτά!

Η γιαγιά, βέβαια, γνωρίζει ότι δεν έχουν όλοι ούτε τις ίδιες προτιμήσεις, ούτε τις ίδιες ανάγκες.

Κάποιοι προτιμούν κασκόλ κοντά ή μακριά, κάποιοι άλλοι πουλοβέρ σε μέγεθος που να τους κάνει.



Φέτος, ο Τοτός αποφάσισε να βοηθήσει τη γιαγιά να προετοιμάσει τα σχέδια που θα πλέξει ανάλογα με την περίπτωση: σίγουρα ένα πρόγραμμα που υπολογίζει αυτόματα το σχέδιο που πρέπει να πλέξει θα της λύσει τα χέρια!

Θέλουμε να γράψουμε ένα πρόγραμμα που θα διαβάσει από την είσοδο `STDIN` ένα χαρακτήρα `C`, ακολουθούμενο από έναν θετικό ακέραιο αριθμό `N`, χωρισμένους από έναν χαρακτήρα κενού `' '`, και θα εκτυπώνει στην έξοδο `STDOUT` πλήθη χαρακτήρων `'x'` ή διαχωριστικών `'\n'` και `'\n'`.

- Ο `C` ($C \in \{ 'K', 'S', 'M', 'L' \}$) συμβολίζει το **είδος** του πλεκτού.
- Ο `N` ($1 \leq N \leq 4$) συμβολίζει το **πλήθος** των σχεδίων του πλεκτού.
- Όταν το είδος του πλεκτού είναι `'K'` **κασκόλ**, το σχέδιο επαναλαμβάνεται **κάθετα**.
- Όταν το είδος του πλεκτού είναι **πουλόβερ** με μέγεθος `'S'` ή `'M'` ή `'L'`, το σχέδιο επαναλαμβάνεται **οριζόντια**.

Προσοχή! Κάθε γραμμή εισόδου ή εξόδου (θα πρέπει να) τερματίζει με το χαρακτήρα αλλαγής γραμμής `'\n'`.

Παραδείγματα

1ο

`STDIN`

```
K 4
```

`STDOUT`

```
x   x   x   x
 x  x       x  x
  xx       xx
 x  x       x  x
 x   x   x   x
```

```
x   x   x   x
x  x   x  x
  xx   xx
x  x   x  x
x   x   x  x
x   x   x  x
  xx   xx
x  x   x  x
x   x   x  x
x   x   x  x
  xx   xx
x  x   x  x
x   x   x  x
```

2ο

STDIN

```
S 3
```

STDOUT

```
xx   xx   xx   xx   xx   xx
xx  xx   xx  xx   xx  xx
  xxxx   xxxx   xxxx
  xx     xx     xx
  xxxx   xxxx   xxxx
xx  xx   xx  xx   xx  xx
xx   xx   xx   xx   xx   xx
xx   xx   xx   xx   xx   xx
xx  xx   xx  xx   xx  xx
  xxxx   xxxx   xxxx
  xx     xx     xx
  xxxx   xxxx   xxxx
xx  xx   xx  xx   xx  xx
xx   xx   xx   xx   xx   xx
```

3ο

STDIN

```
M 1
```

STDOUT


```
xxxx  xxxx
xxxx  xxxx
xxxxxxx
xxxxxx
xxxxxxx
xxxx  xxxx
xxxx  xxxx
xxxx  xxxx
xxxx  xxxx
xxxxxxx
xxxxxx
xxxxxxx
xxxx  xxxx
xxxx  xxxx
```

4ο

STDIN

```
L 2
```

STDOUT

```
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxxxxxxxxx  xxxxxxxxxxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
xxxxxx  xxxxxx  xxxxxx  xxxxxx
```

Μαγικά τετράγωνα

Ο Τοτός πρέπει επιτέλους να κάνει την εργασία τριμήνου στην Πληροφορική. Το θέμα που του έτυχε ήταν τα μαγικά μετράγωνα. Διαβάζει στην Βικιπαίδεια:



«Το μαγικό τετράγωνο αποτελεί διάταξη αριθμών σε συστοιχία ίσου συνόλου γραμμών και στηλών, όπου η αριθμητική πράξη μεταξύ των αριθμών στην ίδια σειρά ή στήλη ή διαγώνιο του τετραγώνου επιστρέφει πάντα το ίδιο αποτέλεσμα. Το κοινό αποτέλεσμα ονομάζεται μαγική σταθερά του μαγικού τετραγώνου. Η πλέον συνηθής αριθμητική πράξη στα μαγικά τετράγωνα είναι η πρόσθεση μεταξύ των αριθμών, ενώ υπάρχουν και άλλες εκδοχές τους...»

2	7	6	→15	
9	5	1	→15	
4	3	8	→15	
↙15	↓15	↓15	↓15	↘15

Σχήμα 1: Παράδειγμα μαγικού τετραγώνου διαστάσεων 3x3 όπου το άθροισμα των αριθμών σε κάθε γραμμή, στήλη και διαγώνιο, ισούται με 15.

Ο αλγόριθμος κατασκευής του τού προκαλεί πονοκέφαλο όμως. Πιο εύκολο του φαίνεται να χρησιμοποιήσει μια μηχανή παραγωγής τυχαίων αριθμών και να επιλέγει αυτούς που σχηματίζουν μαγικό τετράγωνο.

Θέλουμε να γράψουμε ένα πρόγραμμα που θα διαβάζει από την είσοδο `STDIN` έναν θετικό ακέραιο αριθμό N , ακολουθούμενο από N^2 μοναδικούς θετικούς ακέραιους αριθμούς P_i , χωρισμένους από έναν χαρακτήρα κενού ' ' ή αλλαγής γραμμής '\n' και θα ελέγχει αν το δωσμένο τετράγωνο είναι μαγικό τετράγωνο.

Το πρόγραμμα θα εκτυπώνει στην έξοδο `STDOUT` "OXI" αν το τετράγωνο της εισόδου δεν αντιστοιχεί σε μαγικό τετράγωνο ή θα εκτυπώνει στην πρώτη γραμμή εξόδου "NAI" και στην επόμενη γραμμή θα ακολουθεί ένας αριθμός S , αν το τετράγωνο της εισόδου αντιστοιχεί σε μαγικό τετράγωνο.

- Ο N ($1 \leq N \leq 100$) συμβολίζει το μέγεθος της πλευράς του τετραγώνου.
- Οι P_i ($1 \leq P_i \leq N^2$, $1 \leq i \leq N^2$) συμβολίζουν κάθε μοναδικό αριθμό του τετραγώνου.
- Ο S συμβολίζει το άθροισμα μιας σειράς, μιας στήλης ή μιας διαγωνίου του μαγικού τετραγώνου.

Προσοχή! Κάθε γραμμή εισόδου ή εξόδου (θα πρέπει να) τερματίζει με το χαρακτήρα αλλαγής γραμμής '\n' και οι λέξεις "NAI" και "OXI" είναι γραμμένες με λατινικούς χαρακτήρες.

Παραδείγματα

1ο

STDIN

```
3
2 7 6
9 5 1
4 3 8
```

STDOUT

```
NAI
15
```

2ο

STDIN

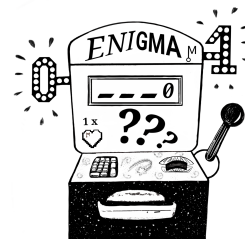
```
3
2 7 4
1 8 9
6 3 5
```

STDOUT

```
OXI
```

Μάντεψε και Κέρδισε!

Ο κόσμος έχει αλλάξει. Τα τυχερά παιχνίδια είναι η μόνη, πλέον, διέξοδος των ανθρώπων να γεμίσουν το χρόνο τους και να ψυχαγωγηθούν. Τα παιδιά δεν αποτελούν εξαίρεση και η νέα μόδα είναι να παίζουν το "Μάντεψε και Κέρδισε!".



Το "Μάντεψε και Κέρδισε!" είναι ένα παιχνίδι τύχης, ο σκοπός του οποίου είναι να μαντέψεις έναν θετικό ακέραιο αριθμό M .

Κάθε προσπάθεια κοστίζει 1 μονάδα νομίσματος. Δεν υπάρχει περιορισμός στον αριθμό προσπαθειών μέχρι να βρεις τον αριθμό, υπάρχει όμως περιορισμός στην τσέπη :).

Τα παιδιά ξοδεύουν πάντα όλο τους το χαρτζιλίκι στο παιχνίδι, και τώρα τα Χριστούγεννα που έχουν στα χέρια τους τις εισπράξεις από τα κάλαντα, μπορούν να παίξουν μέχρι 1000 φορές!

Ο Τοτός, εκτός από εθισμένος στο παιχνίδι, είναι και εξαιρετικά παρατηρητικός. Πρόσεξε ότι κάθε φορά που χάνει, έχοντας προβλέψει έναν αριθμό K μικρότερο από τον τυχερό αριθμό M , ανάβει ένα μικροσκοπικό λαμπάκι με ένδειξη '1'. Στην αντίθετη περίπτωση που έχει προβλέψει έναν αριθμό K μεγαλύτερο από τον M , ανάβει ένα λαμπάκι με ένδειξη '2'. Σε περίπτωση νίκης δεν ανάβει κανένα από τα μικρά λαμπάκια (ένδειξη '0').

Εμείς θέλουμε να ευχαριστήσουμε τον Τοτό για την πληροφορία που μας έδωσε, γράφοντας ένα πρόγραμμα που θα την αξιοποιεί για να μαντέψει **πάντα** τον M πριν εξαντληθούν τα χρήματά του. Το πρόγραμμα αυτό δοκιμάζει, έναν-έναν, αριθμούς K εκτυπώνοντάς τους στην έξοδο `STDOUT` σε **οποιαδήποτε** αποδεκτή αναπαράσταση, ακολουθούμενους από έναν χαρακτήρα αλλαγής γραμμής `'\n'`.

Ύστερα από κάθε δοκιμή, θα διαβάζει από το `STDIN` την απάντηση του παιχνιδιού, έναν ακέραιο L που θα συμβολίζει την παρατήρηση του Τοτού :).

Παράδειγμα

STDIN	STDOUT
	1
1	2
1	3
1	9
2	8
0	