

CCC-05 (2005) - S2 (Move)

Time limit: 1.0s Memory limit: 256M

Mouse Move

Πιθανότατα, θα παρατηρήσετε ότι έχετε ένα ποντίκι συνδεδεμένο στον υπολογιστή σας, το οποίο σας επιτρέπει να μετακινήσετε τον κέρσορα γύρω από την οθόνη. Η δουλειά σας είναι να μπειτε μεταξύ του ποντικιού και του κέρσορα.

Ας υποθέσουμε ότι η κάτω αριστερή γωνία της οθόνης σας είναι η $(0, 0)$ και όλα τα σημεία στην οθόνη δίνονται από ακέραιες συντεταγμένες (x, y) όπου $0 \leq x \leq c$ και $0 \leq y \leq r$. Έτσι, η επάνω δεξιά γωνία της οθόνης είναι το σημείο (c, r) , η κάτω δεξιά γωνία είναι το σημείο $(c, 0)$ και η επάνω αριστερή γωνία είναι το σημείο $(0, r)$.

Όταν μετακινείται ένα ποντίκι, στέλνει ένα ζεύγος ακεραίων αριθμών (a, b) , υποδεικνύοντας ότι ο κέρσορας πρέπει να μετακινηθεί a μονάδες στην κατεύθυνση x και b μονάδες στην κατεύθυνση y . Αξίζει να σημειωθεί ότι πρόκειται για *σχετική κίνηση* (δηλαδή πόσο μακριά πρέπει να κινηθεί), παρά για *απόλυτη κίνηση* (δηλαδή πού να κινηθεί). Αξίζει επίσης να σημειωθεί ότι τα a και b μπορεί να είναι θετικά, αρνητικά ή μηδενικά.

Μπορείτε να υποθέσετε ότι το ποντίκι ξεκινά στη θέση $(0, 0)$. Η δουλειά σας είναι να διαβάζετε μηνύματα εισόδου (δηλαδή, σχετικές θέσεις κίνησης που αποστέλλονται από το ποντίκι) και να ενημερώνετε τον κέρσορα στη νέα θέση της οθόνης. Η έξοδος σας (στην οθόνη) θα είναι η θέση του ποντικιού μετά από κάθε κίνηση.

Εάν το ποντίκι χτυπήσει το όριο της οθόνης, σταματά να κινείται προς αυτή την κατεύθυνση. Για παράδειγμα, εάν το ποντίκι υποτίθεται ότι κινείται προς τη θέση $(-100, -10)$ από την τρέχουσα θέση του $(30, 40)$, οι τελικές συντεταγμένες θα είναι $(0, 30)$: το ποντίκι θα χτυπήσει το όριο της αριστερής πλευράς, αλλά μπορεί ακόμα να κινηθεί προς τα κάτω.

Η είσοδος παρατίθεται σε ζεύγη, το πρώτο ζεύγος είναι το (c, r) , ακολουθούμενο από τα σχετικά ζεύγη κίνησης (x, y) . Η είσοδος τερματίζεται όταν το ποντίκι κινηθεί στο $(0, 0)$.

Παραδείγματα

input

```
100 200
10 40
-5 15
30 -30
0 0
```

output

```
10 40
5 55
35 25
```

input

```
30 40  
30 40  
-100 -10  
0 0
```

output

```
30 40  
0 30
```