

Η Chika θέλει να κάνει ζαβολιά (Chika Wants to Cheat)

Όνομα Προβλήματος	Η Chika θέλει να κάνει ζαβολιά (Cheat)
Αρχείο εισόδου	Διαδραστικό πρόβλημα
Αρχείο εξόδου	Διαδραστικό πρόβλημα
Χρονικό όριο	2 δευτερόλεπτα
Όριο μνήμης	512 megabytes

Η Chika έχει μια τράπουλα με q κάρτες αριθμημένες με διάφορους θετικούς ακεραίους. Θέλει να παίξει κάποιες κάρτες με τους φίλους της από το συμβούλιο μαθητών της Ακαδημίας Shuchi'in χρησιμοποιώντας αυτές τις κάρτες, αλλά επίσης θέλει και να κερδίσει, οπότε αποφασίζει να σημαδέψει το πίσω μέρος των καρτών στην τράπουλα της.

Οι κάρτες έχουν όλες τετράγωνο σχήμα με διάσταση 2×2 όπου η κάτω αριστερή γωνία τους έχει συντεταγμένες $(0, 0)$ και η επάνω δεξιά $(2, 2)$. Η Chika σημαδεύει με ένα συγκεκριμένο μοτίβο την οπίσθια όψη (πλάτη) κάθε κάρτας, έτσι ώστε αργότερα να ξέρει, κοιτώντας το μοτίβο, ποιον αριθμό περιέχει στην εμπρόσθια όψη της. Ζωγραφίζει μοτίβα ακολουθώντας την ακόλουθη διαδικασία: Όσες φορές θέλει (πιθανόν 0 φορές), επιλέγει δυο μοναδικά σημεία A και B που έχουν ακέραιες συντεταγμένες σε σχέση με την κάτω αριστερή γωνία της κάρτας και σχεδιάζει το **ευθύγραμμο τμήμα** ανάμεσα στα δυο σημεία.

Η Chika θα σχεδιάσει μόνο **έγκυρα** ευθύγραμμα τμήματα, δηλαδή, τμήματα μεταξύ των σημείων A και B για τα οποία δεν υπάρχει κάποιο άλλο σημείο C (διαφορετικό από τα A και B) με ακέραιες συντεταγμένες που να περιέχεται στο ίδιο ευθύγραμμο τμήμα. Για παράδειγμα, το τμήμα ανάμεσα στα $(0, 0)$ και $(2, 2)$ είναι **άκυρο** καθώς περιέχει το σημείο $(1, 1)$, ενώ τα ευθύγραμμα τμήματα μεταξύ των $(0, 0)$ και $(1, 1)$ και μεταξύ των $(1, 1)$ και $(2, 2)$ είναι και τα δυο **έγκυρα**, και η Chika μπορεί ακόμα και να ζωγραφίσει και τα δύο αυτά τμήματα στο ίδιο μοτίβο. Επίσης, λάβετε υπόψη ότι τα ευθύγραμμα τμήματα δεν έχουν φορά: Ένα ευθύγραμμο τμήμα σχεδιασμένο από το A στο B είναι **ταυτόσημο** με το ευθύγραμμο τμήμα ανάποδης φοράς, δηλαδή από το B στο A .

Είναι επίσης σημαντικό, η Chika να μπορεί να αναγνωρίζει τις κάρτες της τράπουλας άσχετα με το πως αυτές έχουν περιστραφεί. Μια κάρτα μπορεί να περιστραφεί 0, 90, 180 ή 270 μοίρες αντίστροφα από τους δείκτες του ρολογιού σε σχέση με τον αρχικό της προσανατολισμό.

Η δουλειά σας είναι να βοηθήσετε την Chika να σχεδιάσει μοτίβα για τις q κάρτες στην τράπουλα της και μετά να τις αναγνωρίσει.

Υλοποίηση

Αυτό είναι ένα διαλογικό task σε δυο στάδια, **κάθε στάδιο περιλαμβάνει ξεχωριστή εκτέλεση του προγράμματος σας**. Πρέπει να υλοποιήσετε δυο συναρτήσεις:

- Μια συνάρτηση με όνομα `BuildPattern` που επιστρέφει το μοτίβο που πρέπει να σχεδιαστεί στην πίσω όψη μιας συγκεκριμένης κάρτας. Η συνάρτηση αυτή θα κληθεί q φορές κατά το πρώτο στάδιο του task.
- Μια συνάρτηση `GetCardNumber` που επιστρέφει τον αριθμό μιας (πιθανόν περιστραμμένης) κάρτας που έχει σχεδιασμένο ένα συγκεκριμένο μοτίβο σχεδιασμένο κατά το πρώτο στάδιο. Η συνάρτηση αυτή θα κληθεί q φορές κατά το δεύτερο στάδιο.

Η πρώτη συνάρτηση

```
std::vector<std::pair<std::pair<int, int>, std::pair<int, int>>> BuildPattern(int n);
```

παίρνει μια μοναδική παράμετρο, τον ακέραιο n , ο οποίος απεικονίζεται στην εμπρόσθια όψη της. Πρέπει να επιστρέψετε ένα `std::vector` που να περιέχει τα ευθύγραμμα τμήματα που η Chika θα σχεδιάσει για το μοτίβο στην οπίσθια όψη της κάρτας ώστε να την αναγνωρίζει αργότερα. Ένα ευθύγραμμο τμήμα αναπαρίσταται από ένα `std::pair` σημείων, και ένα σημείο αναπαρίσταται από ένα `std::pair(x, y)` ακέραιων συντεταγμένων ορισμένων σε σχέση με την κάτω αριστερή γωνία της κάρτας, όπου $0 \leq x, y \leq 2$. Όλα τα ευθύγραμμο τμήματα που η Chika θα σχεδιάσει πρέπει να είναι έγκυρα και να μην είναι ταυτόσημα με άλλα ευθύγραμμο τμήματα του μοτίβου. Είναι εγγυημένο ότι όλες οι q κλησεις στη `BuildPattern` λαμβάνουν διαφορετικές τιμές για την παράμετρο n .

Αφού λάβει όλα τα μοτίβα για τις q κάρτες, ο βαθμολογητής (grader) μπορεί να κάνει οποιοδήποτε από τις παρακάτω ενέργειες, όσες φορές θέλει, σε κάθε ένα από τα μοτίβα:

- Να περιστρέψει ολόκληρο το μοτίβο κατά 0, 90, 180 ή 270 μοίρες αντίστροφα από τη φορά των δεικτών του ρολογιού.
- Να αλλάξει τη σειρά των ευθύγραμμων τμημάτων στην αναπαράσταση `std::vector` του μοτίβου.
- Να αλλάξει τη σειρά των σημείων ενός ευθύγραμμου τμήματος στο μοτίβο. (Δηλαδή το ευθύγραμμο τμήμα από το A στο B , να γίνει το ταυτόσημο του, από το B στο A).

Η δεύτερη συνάρτηση,

```
int GetCardNumber(std::vector<std::pair<std::pair<int, int>, std::pair<int, int>>> p);
```

λαμβάνει μια μοναδική παράμετρο p , τύπου `std::vector` από ευθύγραμμα τμήματα που περιγράφουν το μοτίβο που σχεδίασε η Chika στην οπίσθια όψη της κάρτας, βασισμένη στην απάντησή σας από προηγούμενη κλήση της συνάρτησης `BuildPattern`. Η συνάρτηση πρέπει να επιστρέψει τον αριθμό n της εμπρόσθιας όψης της κάρτας. Προσέξτε ότι το μοτίβο p δεν είναι απαραίτητα στην ίδια διάταξη που το επέστρεψε η `BuildPattern`; αλλά μπορεί να έχουν εφαρμοστεί σε αυτό οι τρεις ενέργειες που περιγράφηκαν προηγουμένως. Είναι επίσης πιθανό η σειρά των καρτών να είναι διαφορετική από τη σειρά που δόθηκαν στο πρώτο στάδιο.

Περιορισμοί

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$ για όλες τις κλήσεις στη συνάρτηση `BuildPattern`.
- Λάβετε υπόψη, ότι υπάρχουν αλγόριθμοι να δημιουργήσουν μοτίβα τέτοια ώστε κάποιος να αναγνωρίζει 67 000 000 διαφορετικές κάρτες.

Βαθμολόγηση

- Subtask 1 (2 βαθμοί): $n \leq 2$.
- Subtask 2 (9 βαθμοί): $n \leq 25$.
- Subtask 3 (15 βαθμοί): $n \leq 1\,000$ και ο βαθμολογητής (grader) **δεν θα περιστρέψει** τα μοτίβα ανάμεσα στο στάδιο 1 και 2. (Ο βαθμολογητής **μπορεί** να εκτελέσει τις υπόλοιπες δυο ενέργειες).
- Subtask 4 (3 βαθμοί): $n \leq 16\,000\,000$ και ο βαθμολογητής **δεν θα περιστρέψει** τα μοτίβα ανάμεσα στο στάδιο 1 και 2. (Ο βαθμολογητής **μπορεί** να εκτελέσει τις υπόλοιπες δυο ενέργειες).
- Subtask 5 (24 βαθμοί): $n \leq 16\,000\,000$.
- Subtask 6 (18 βαθμοί): $n \leq 40\,000\,000$.
- Subtask 7 (29 βαθμοί): Χωρίς επιπλέον περιορισμούς.

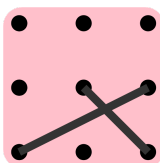
Διαδραστικό Παράδειγμα

Κλήση Συνάρτησης	Επιστρεφόμενη Τιμή	Επεξήγηση
Πρώτο στάδιο.	-	-
<code>BuildPattern(3)</code>	<code>{{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}</code>	Έχουμε να σχεδιάσουμε ένα μοτίβο για τον αριθμό 3 στην κάρτα διάστασης 2×2 . Αποφασίζουμε να σχεδιάσουμε 2 ευθύγραμμα τμήματα: <ul style="list-style-type: none">- μεταξύ (0, 0) και (2, 1),- μεταξύ (1, 1) και (2, 0).

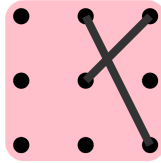
BuildPattern(1)	{{{0, 1}, {0, 0}}}	Έχουμε να σχεδιάσουμε ένα μοτίβο για τον αριθμό 1 στην κάρτα διάστασης 2×2 . - μεταξύ (0,1) και (0,0).
Τέλος πρώτου σταδίου.	-	-
Αρχή δεύτερου σταδίου.	-	-
GetCardNumber ({{{0, 0}, {0, 1}}})	1	Λαμβάνουμε ένα μοτίβο αποτελούμενο από 1 ευθύγραμμο τμήμα: - μεταξύ (0,0) και (0,1) Αυτό είναι το ίδιο μοτίβο με το να σχεδιάζαμε το ευθύγραμμο τμήμα: - μεταξύ (0,1) και (0,0) το οποίο είναι ταυτόσημο με το μοτίβο με τον ίδιο προσανατολισμό (περιστραμμένο κατά 0 μοίρες) που δημιουργήσαμε και επιστρέψαμε κατά τη δεύτερη κλήση στη συνάρτηση BuildPattern. Επομένως, επιστρέφουμε 1.
GetCardNumber ({{{1, 1}, {2, 2}}, {{1, 2}, {2, 0}}})	3	Λαμβάνουμε ένα μοτίβο αποτελούμενο από 2 ευθύγραμμα τμήματα: - μεταξύ (1,1) και (2,2), - μεταξύ (1,2) και (2,0). Αυτό είναι το μοτίβο που δημιουργήσαμε και επιστρέψαμε στην πρώτη κλήση στη συνάρτηση BuildPattern, περιστραμμένο κατά 90 μοίρες αντίστροφα από τους δείκτες του ρολογιού. Επομένως, επιστρέφουμε 3.
Τέλος δεύτερου σταδίου.	-	-

Οι επόμενες τρεις φωτογραφίες αναπαριστούν με τη σειρά:

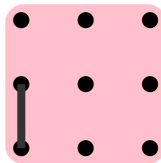
- Το μοτίβο που επιστραφικε από την πρώτη κλήση στη BuildPattern:



- Το μοτίβο που λήφθηκε ως παράμετρος στη δεύτερη κλήση στη `GetCardNumber`, το οποίο είναι το πρώτο μοτίβο περιστραμμένο κατά 90 μοίρες αντίστροφα με τους δείκτες του ρολογιού.



- Το μοτίβο που επιστράφηκε από τη `BuildPattern` κατά τη δεύτερη κλήση της, το οποίο είναι ίδιο με το μοτίβο που λήφθηκε σαν παράμετρος στην πρώτη κλήση της `GetCardNumber`.



Παράδειγμα Βαθμολογητής (Sample Grader)

Το παράδειγμα βαθμολογητή που παρέχεται, `grader.cpp`, στο επισυναπτόμενο `Cheat.zip`, διαβάζει έναν ακέραιο q από την τυπική είσοδο και θα εκτελέσει τα επόμενα βήματα q φορές:

- Διαβάσε ακέραιο αριθμό n από την τυπική είσοδο.
- Κάλυψε τη `BuildPattern(n)` και αποθήκευσε την επιστρεφόμενη τιμή στην μεταβλητή p .
- Κάλυψε τη `GetCardNumber(p)` και τύπωσε την επιστρεφόμενη τιμή στην τυπική έξοδο.

Μπορείτε να τροποποιήσετε τον βαθμολογητή τοπικά (στον υπολογιστή σας) αν το επιθυμείτε.

Για να μεταγλωττίσετε το παράδειγμα βαθμολογητή με το πρόγραμμά σας, μπορείτε να χρησιμοποιήσετε την ακόλουθη εντολή στη γραμμή εντολών του τερματικού (terminal prompt):

```
g++ -std=gnu++11 -O2 -o solution grader.cpp solution.cpp
```

όπου `solution.cpp` είναι το αρχείο με το πρόγραμμά σας που θα υποβάλετε στο CMS. Για να τρέξετε το πρόγραμμά σας με το παράδειγμα βαθμολογητή που παρέχεται σαν επισύναψη, γράψτε την παρακάτω εντολή στη γραμμή εντολών του τερματικού (terminal prompt):

```
./solution < input.txt
```

Παρακαλώ λάβετε υπόψη, ότι αντίθετα με το παράδειγμα βαθμολογητή, ο πραγματικός βαθμολογητής του CMS θα εκτελέσει το πρώτο και το δεύτερο στάδιο με δυο διαφορετικές εκτελέσεις του προγράμματός σας.