

## Chika tahtoo huijata

Tehtävän nimi	Huijaus
Syötetiedosto	Interaktiivinen tehtävä
Tulostetiedosto	Interaktiivinen tehtävä
Aikaraja	2 sekuntia
Muistiraja	512 megabittia

Chikalla on korttipakka, jonka  $q$  korttia on numeroitu erilaisilla positiivisilla kokonaisluvuilla. Hän haluaa pelata pelejä Shuchi'in Academy -koulun oppilaskunnasta tuntemiensa ystävien kanssa, mutta hän haluaa myös voittaa, joten hän päättää tehdä salaa merkintöjä pakkansa korttien selkäpuolelle.

Kortit ovat kaikki neliön muotoisia ja kokoa  $2 \times 2$ , missä ala-vasen kulma on koordinaatti  $(0, 0)$  ja ylä-oikea kulma  $(2, 2)$ . Chika piirtää tietynlaisen kuvion jokaisen kortin selkäpuolelle, jotta hän myöhemmin kuviota katsomalla muistaa, mikä numero on kortin oikealla puolella. Hän piirtää kuvion käyttämällä seuraavaa menetelmää: Niin monta kertaa kuin häntä huvittaa (mahdollisesti 0 kertaa) hän valitsee kaksi erillistä pistettä  $A$  ja  $B$ , joiden kokonaislukukoordinaatit vasemman alakulman suhteen tiedetään, ja piirtää **suoran viivanpätjän** niiden välille.

Kuitenkin Chika piirtää vain **kelvollisia** viivoja, eli sellaisia viivoja pisteiden  $A$  ja  $B$  välille, että ei löydy vielä kolmatta ( $A$ :sta ja  $B$ :stä erillinen) pistettä  $C$ , jonka kokonaislukukoordinaatit ovat myös viivalla. Esimerkiksi viiva  $(0, 0)$ :n ja  $(2, 2)$ :n välillä on epäkelpo, koska sillä on myös piste  $(1, 1)$ , mutta viivat  $(0, 0)$ :n ja  $(1, 1)$ :n välillä ja  $(1, 1)$ :n ja  $(2, 2)$ :n välillä ovat **kelvollisia**, ja Chika voi piirtää vaikka molemmat samaan kuvioon. Ota myös huomioon että viivoilla ei ole suuntaa: viiva  $A$ :sta  $B$ :hen on **identtinen** paitsi itsensä, myös viivan  $B$ :stä  $A$ :han kanssa.

On tärkeää, että Chika tunnistaa korttinsa riippumatta siitä, miten niitä on kierretty. Korttia voi kiertää  $0$ ,  $90$ ,  $180$  tai  $270$  astetta vastapäivään niiden alkuperäiseen suuntaan nähden.

Tehtäväsi on auttaa Chikaa suunnittelemaan kuviot pakkansa  $q$ :lle kortille ja tunnistamaan kortit jälkeenpäin.

# Toteutus

Tämä on interaktiivinen kaksivaiheinen tehtävä **ja sinun täytyy ajaa ohjelmasi kummassakin vaiheessa erikseen**. Sinun täytyy toteuttaa kaksi funktiota:

- `BuildPattern`-funktio (eli "tee kuvio"), joka palauttaa kuvion joka piirretään annetun kortin selkäpuolelle. Tätä funktiota kutsutaan  $q$  kertaa ensimmäisessä vaiheessa.
- `GetCardNumber`-funktio (eli "etsi kortin numero") joka palauttaa (mahdollisesti kierretyn) kortin numeron edellisessä vaiheessa siihen piirretyn kuvion perusteella. Tätä funktiota kutsutaan  $q$  kertaa toisessa vaiheessa.

## Ensimmäinen funktio

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

ottaa syötteenä yhden parametrin  $n$ , eli numeron joka on kirjoitettu kortin etupuolelle. Sinun täytyy palauttaa `std::vector`-vektori, joka sisältää viivat, joista muodostuu Chikan kortin selkäpuolelle piirtämä tunnistekuvio. Viiva kuvataan `std::pair`-parina pisteitä, ja piste on esitetty `std::pair`  $(x, y)$ -parina kortin vasemmasta alakulmasta lähteviä kokonaislukukoordinaatteja, missä  $0 \leq x, y \leq 2$ . Jokaisen Chikan piirtämän viivan täytyy olla kelvollinen ja niiden on oltava keskenään epäidenttisiä. On taattu että jokainen  $q$ :sta `BuildPattern`-kutsusta saa eri arvot parametrille  $n$ .

Vastaanotettuaan kuviot kaikille  $q$  kortille, arvioija (grader) voi tehdä mitä vain seuraavista operaatioista mielivaltaisen monta kertaa jokaiselle kuviolle:

- Kiertää koko kuviota 0, 90, 180 tai 270 astetta vastapäivään.
- Muuttaa viivojen keskinäistä järjestystä kuvion `std::vector`-esityksessä
- Vaihtaa viivojen päätepisteiden keskinäistä järjestystä. (Eli viiva  $A$ :sta  $B$ :hen muuttuu identtiseksi viivaksi  $B$ :stä  $A$ :han.)

## Toinen funktio

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

ottaa syötteenä yhden parametrin  $p$ , joka on `std::vector`-vektori, joka kuvaa Chikan kortin selkäpuolelle piirtämää kuviota edellisen `BuildPattern`-funktio-kutsun palautusarvon perusteella. Tämän funktion täytyy palauttaa numero  $n$  joka on kirjoitettu kortin etupuolelle. Muista että kuvio  $p$  ei välttämättä ole samaa muotoa kuin jossa se palautettaisiin funktiolta `BuildPattern`, sillä sitä on voitu muokata yllä olevien kolmen operaation kautta. On myös mahdollista, että korttien järjestys on eri kuin se, jossa ne annettiin ensimmäisessä vaiheessa, mutta on taattu että jokaista korttia käytetään tasan kerran.

## Rajat

- $1 \leq q \leq 10\,000$ .
- $1 \leq n \leq 67\,000\,000$  kaikille `BuildPattern`-funktiokutsuille
- Ota huomioon, että on olemassa algoritmeja, joiden avulla voi muodostaa 67 000 000 kortille erilaiset tunnistuskuviot

## Pisteytys

- Alitehtävä 1 (2 pistettä):  $n \leq 2$ .
- Alitehtävä 2 (9 pistettä):  $n \leq 25$ .
- Alitehtävä 3 (15 pistettä):  $n \leq 1\,000$  ja arvioija **ei kierrä** kuvioita vaiheiden 1 ja 2 välillä. (Arvioija **voi** toteuttaa kahta muuta operaatiota.)
- Alitehtävä 4 (3 pistettä):  $n \leq 16\,000\,000$  ja arvioija **ei kierrä** kuvioita vaiheiden 1 ja 2 välillä. (Arvioija **voi** toteuttaa kahta muuta operaatiota.)
- Alitehtävä 5 (24 pistettä):  $n \leq 16\,000\,000$ .
- Alitehtävä 6 (18 pistettä):  $n \leq 40\,000\,000$ .
- Alitehtävä 7 (29 pistettä): Ei muita rajoitteita.

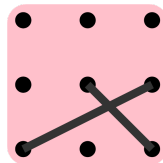
## Esimerkki-interaktio

Funktiokutsu	Palautusarvo	Selitys
Ensimmäinen kierros alkaa.	-	-
<code>BuildPattern(3)</code>	<code>{{{0, 0}, {2, 1}}, {1, 1}, {2, 0}}}</code>	Meidän täytyy muodostaa kuvio numerolle 3 kortille, jonka koko on $2 \times 2$ . Päättämme piirtää 2 viivaa: - (0,0):stä (2,1):een, - (1,1) :stä (2,0):een.
<code>BuildPattern(1)</code>	<code>{{{0, 1}, {0, 0}}}</code>	Meidän täytyy muodostaa kuvio numerolle 1 kortille, jonka koko on $2 \times 2$ . Päättämme piirtää 1 viivan: - (0,1):stä (0,0) : een.
Ensimmäinen kierros päättyy.	-	-
Toinen kierros alkaa.	-	-
<code>GetCardNumber( {{{0, 0}, {0, 1}}})</code>	1	Meille on annettu kuvio joka muodostuu 1 viivasta: - (0,0):stä (0,1):een. Saisimme saman kuvion jos piirtäisimme seuraavan viivan: - (0,1):stä (0,0):een

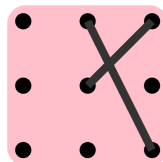
		joka on täsmälleen sama kuvio, ja samansuuntainen (kierretty 0 astetta), kuin se jonka saimme palautusarvona kutsuessamme funktiota <code>BuildPattern</code> toisen kerran. Siis palautamme numeron 1.
<pre>GetCardNumber(   {{{1, 1}, {2, 2}},   {{1, 2}, {2, 0}}})</pre>	3	Meille on annettu kuvio joka muodostuu 2 viivasta: - (1,1):sta (2,2):een, - (1,2):sta (2,0):een. Tämä on sama kuvio kuin se, jonka saimme ensimmäisen <code>BuildPattern</code> -kutsun palautusarvona, kierrettynä 90 astetta vastapäivään. Siis palautamme numeron 3.
Toinen kierros päättyy.	-	-

Nämä kolme kuvaa kuvaavat seuraavia tilanteita:

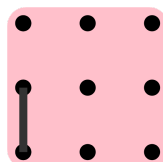
- Kuvio sellaisena kuin se palautetaan ensimmäisestä `BuildPattern`-kutsusta:



- Kuvio annettuna parametriksi toiselle `GetCardNumber`-kutsulle, joka on sama kuin ensimmäisen kuvio 90 asteen vastapäivään kiertämisen jälkeen.



- Kuvio sellaisena kuin se palautetaan toisesta `BuildPattern`-kutsusta, eli sama kuvio kuin se joka on annettu argumentiksi ensimmäisessä `GetCardNumber`-kutsussa.



## Esimerkkiarvioija

Tarjolla oleva esimerkkiarvioija, `grader.cpp`, joka löytyy tehtävän `Cheat.zip`-liitteestä, ottaa sisään kokonaisluvun  $q$  standardisyötteestä ja tekee sitten seuraavat vaiheet  $q$  kertaa:

- Lukee kokonaisluvun  $n$  standardisyötteestä.
- Kutsuu funktiota `BuildPattern(n)` ja tallentaa sen palautusarvon muuttujaan  $p$ .
- Kutsuu funktiota `GetCardNumber(p)` ja tulostaa sen palautusarvon standarditulosteena.

Saat muokata arvioijaasi lokaalisti jos haluat.

Kääntääksesi esimerkkiarvioijan koodillesi, voit käyttää komentoriviltä seuraavaa komentoa:

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

missä `solution.cpp` on ratkaisutiedosto, jonka aiot palauttaa CMS:ään. Ajaaksesi ohjelman liitteenä olevalla esimerkisyötteellä, kirjoita seuraava komento komentoriville:

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

**Ota huomioon, että toisin kuin esimerkkiarvioija, oikea CMS:n arvioija toteuttaa ensimmäisen ja toisen vaiheen erillisinä ohjelmasi ajokertoina.**