Čika grib krāpties

Uzdevuma nosaukums	Krāpšanās
Ievaddati	Interaktīvs uzdevums
Izvaddati	Interaktīvs uzdevums
Laika limits	2 sekundes
Atmiņas limits	512 megabaiti

Čikai ir kāršu kava ar q kārtīm, kas ir sanumurētas ar dažādiem veseliem pozitīviem skaitļiem. Viņa vēlas uzspēlēt dažas spēles ar saviem draugiem no Shuchi'in akadēmijas studentu padomes, izmantojot šīs kārtis, taču viņa vēlas arī uzvarēt, tāpēc viņa nolemj savā kavā slepeni iezīmēt kāršu aizmugures.

Visas kārtis ir kvadrāta formā, un to izmērs ir 2×2 (apakšējā kreisā stūra koordinātas ir $(0, 0)$ un augšējā labā stūra koordinātas ir $(2, 2)$). Čika katras kartītes aizmugurē uzzīmē noteiktu rakstu, lai vēlāk, skatoties uz zīmējumu, zinātu, kurš skaitlis ir uzrakstīts kartītes priekšpusē. Viņa zīmē rakstu šādi: tik reižu, cik viņa vēlas (iespējams, 0 reizes), viņa izvēlas divus atšķirīgus punktus A un B, kuriem ir vesela skaitļa koordinātas attiecībā pret kārts apakšējo kreiso stūri, un starp tiem novelk **taisnas līnijas segmentu**.

Čika zīmēs tikai **derīgus** segmentus, tas ir, segmentus starp diviem punktiem A un B, kuriem nav cita punkta C (kas atšķiras no A un B) ar veselām koordinātēm, kas arī atrodas uz segmenta. Piemēram, segments starp $(0, 0)$ un $(2, 2)$ **nav derīgs**, jo tajā ir punkts $(1, 1)$, bet segmenti starp $(0, 0)$ un $(1, 1)$ un starp $(1, 1)$ un $(2, 2)$ ir **derīgi**, un Čika pat var zīmēt tos abus vienā rakstā. Jāņem vērā arī to, ka segmentiem nav virziena: segments, kas novilkts no A līdz B, ir **identisks** gan ar sevi, gan ar pretējā virzienā novilktu segmentu no B līdz A.

Svarīgi, ka Čika vēlas atpazīt savas kārtis neatkarīgi no tā, kā tās ir pagrieztas. Kārti var pagriezt par 0, 90, 180 vai 270 grādiem pretēji pulksteņrādītāja virzienam attiecībā pret sākotnējo orientāciju.

Uzdevums ir palīdzēt Čikai izveidot rakstus q kārtīm viņas kavā un pēc tam atpazīt šīs kārtis.

Realizācija

Šis ir interaktīvs uzdevums ar diviem posmiem, ***katrs posms ietver atsevišķu programmas izpildi***. Nepieciešams realizēt divas funkcijas:

- Funkcija `BuildPattern`, kas atgriež rakstu, kas jāzīmē konkrētās kārts aizmugurē. Pirmajā posmā šī funkcija tiks izsaukta q reizes.
- Funkcija `GetCardNumber`, kas atgriež skaitli, kurš ir uz (iespējams, pagrieztas) kārts, uz kuras ir pirmajā posmā uzzīmēts raksts. Otrajā posmā šī funkcija tiks izsaukta q reizes.

Pirmajai funkcijai

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

tiek padots viens parametrs n - skaitlis, kas ir rakstīts kārts priekšpusē. Jāatgriež datu struktūra `std::vector` ar segmentiem, ko Čika uzzīmē uz kārts aizmugures, lai to atpazītu vēlāk. Segmentu apzīmē punktu pāris (datu struktūra `std::pair`), un punktu arī apzīmē pāris (`std::pair`) – (x, y) koordinātas attiecībā pret kārts apakšējo kreiso stūri. $0 \leq x, y \leq 2$. Visiem segmentiem, ko Čika uzzīmē, jābūt derīgiem un pa pāriem atšķirīgiem. Ir garantēts, ka visi q funkcijas `BuildPattern` izsaukumi saņem atšķirīgas parametra n vērtības.

Pēc visu q kāršu rakstu saņemšanas vērtētājs var neierobežotu skaitu reižu ar katru no paraugiem veikt jebkuru no tālāk norādītajām darbībām:

- Pagriezt visu rakstu par 0, 90, 180 vai 270 grādiem pretēji pulksteņrādītāja virzienam.
- Mainīt segmentu secību raksta attēlojumā datu struktūrā `std::vector`.
- Mainīt segmenta galapunktu secību rakstā. (segments, kas zīmēts no A līdz B, var kļūt par identisku segmentu, kas zīmēts no B līdz A.)

Otrajai funkcijai

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

tiek padots viens parametrs p, datu struktūra `std::vector` ar segmentiem, kas attēlo rakstu, ko Čika uzzīmē uz kārts aizmugures, atbilstoši iepriekš veiktā funkcijas `BuildPattern` izsaukuma atgrieztajai vērtībai. Funkcijai jāatgriež skaitlis n, kas rakstīts kārts priekšpusē. Jāņem vērā, ka raksts p var nebūt tādā pašā formā, kā to atgrieza funkcija `BuildPattern`; ar to varētu būt veiktas iepriekš aprakstītās trīs operācijas. Tāpat ir iespējams, ka kāršu secība atšķiras no tās, kas tika dota pirmajā posmā, bet ir garantēts, ka katra kārts tiek izmantota tieši vienreiz.

Ierobežojumi

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$ visiem funkcijas `BuildPattern` izsaukumiem.
- Ņemiet vērā, ka pastāv algoritmi, kas izveido rakstus tā, lai varētu atpazīt 67\,000\,00067000000 dažādas kārtis.

Vērtēšana

- 1.apakšuzdevums (2 punkti): $n \leq 2$.
- 2.apakšuzdevums (9 punkti): $n \leq 25$.
- 3.apakšuzdevums (15 punkti): $n \leq 1\,000$ un vērtētājs **negrozīs** rakstus starp 1. un 2. posmu (bet vērtētājs **var** izpildīt pārējās divas operācijas).
- 4.apakšuzdevums (3 punkti): $n \leq 16\,000\,000$ un vērtētājs **negrozīs** rakstus starp 1. un 2. posmu (bet vērtētājs **var** izpildīt pārējās divas operācijas).
- 5.apakšuzdevums (24 punkti): $n \leq 16\,000\,000$.
- 6.apakšuzdevums (18 punkti): $n \leq 40\,000\,000$.
- 7.apakšuzdevums (29 punkti): Bez papildu ierobežojumiem.

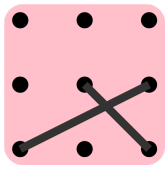
Interakcijas paraugs

Funkcijas izsaukums	Atgriežamā vērtība	Paskaidrojums
Sākas pirmais posms.	-	-
`BuildPattern(3)`	`{{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}`	Uz kārts ar izmēru 2×2 jāizveido raksts skaitlim 3. Var zīmēt 2 segmentus: - starp $(0, 0)$ un $(2, 1)$, - starp $(1, 1)$ un $(2, 0)$.
`BuildPattern(1)`	`{{{0, 1}, {0, 0}}}`	Uz kārts ar izmēru 2×2 jāizveido raksts skaitlim 1. Var zīmēt 1 segmentu: - starp $(0, 1)$ un $(0, 0)$.
Beidzas pirmais posms.	-	-
Sākas otrais posms.	-	-
`GetCardNumber( {{{0, 0}, {0, 1}}})`	`1`	Funkcijai tiek padots raksts, ko veido 1 segments: - starp (0, 0) un (0, 1). Šis ir tāds pats raksts, kādu iegūtu, zīmējot segmentu: - starp (0, 1) un (0, 0) kas ir tieši tas pats raksts ar tādu pašu orientāciju (tas ir pagriezts par 0 grādiem), kā raksts, kas tika atgriezts otrajā funkcijas `BuildPattern`

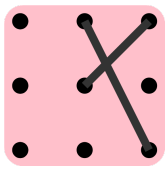
		izsaukumā. Tāpēc tiek atgriezts skaitlis 1.
`GetCardNumber( {{{1, 1}, {2, 2}}, {{1, 2}, {2, 0}}})`	`3`	Funkcijai tiek padots raksts, ko veido 2 segmenti: - starp $(1, 1)$ un $(2, 2)$, - starp $(1, 2)$ un $(2, 0)$. Šis ir raksts, ko iegūst, pirmā funkcijas `BuildPattern` izsaukuma atgriezto rakstu pagriežot par 90 grādiem pretpulksteņrādītājvirzienā. Tāpēc tiek atgriezts skaitlis 3.
Beidzas otrais posms.	-	-

Nākamajos trīs attēlos parādīts attiecīgi:

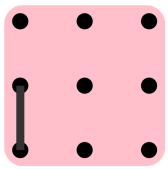
- Pirmā funkcijas `BuildPattern` izsaukuma atgrieztais raksts:



- Otrajam funkcijas `GetCardNumber` izsaukumam padotais raksts, kas sakrīt ar par 90 grādiem pretpulksteņrādītājvirzienā pagrieztu pirmo rakstu:



- Otrā funkcijas `BuildPattern` izsaukuma atgrieztais raksts, kas sakrīt ar funkcijas `GetCardNumber` pirmajam izsaukumam padoto rakstu:



Paraugvērtētājs

Uzdevuma pielikumā `Cheat.zip` esošais paraugvērtētājs `grader.cpp` ielasa veselu skaitli q no standarta ievades un izdara šādas operācijas q reizes:

- Ielasa veselu skaitli n no standarta ievades.
- Izsauc funkciju `BuildPattern(n)` un saglabā tās atgriezto vērtību mainīgajā p.
- Izsauc funkciju `GetCardNumber(p)` un izdrukā tās atgriezto vērtību standarta izvadē.

Pēc vēlēšanās paraugvērtētāju ir iespējams lokāli pārveidot.

Lai kompilētu paraugvērtētāju ar savu risinājumu, komandrindā jāizmanto šāda komanda:

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

kur `solution.cpp` ir risinājuma fails, kas būs jāiesniedz sacensību sistēmā (CMS). Lai darbinātu programmu ar pielikumā esošo ievaddatu paraugu, komandrindā jāizmanto šāda komanda:

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

Jāņem vērā, ka atšķirībā no paraugvērtētāja īstais vērtētājs CMS veiks pirmo un otro posmu atsevišķās programmas izpildēs.