Chika tahab sohki teha

Ülesande nimi	Cheat
Sisend	Interaktiivne ülesanne
Väljund	Interaktiivne ülesanne
Ajapiirang	2 sekundit
Mälupiirang	512 megabaiti

Chikal on kaardipakk q kaardiga, nummerdatud mingite positiivsete täisarvudega. Ta tahab oma sõpradega Shuchi'ni Akadeemia õpilasesindusest nende kaartidega mänge mängida, aga samuti tahab Chika võita ning seetõttu otsustab ta salaja oma pakis kaartide tagaküljed ära märgistada.

Kõik kaardid on ruudukujulised, suurusega 2×2, kus alumine vasak nurk on koordinaatidega $(0, 0)$ ja ülemine parem nurk $(2, 2)$. Chika joonistab iga kaardi tagaküljele mingi mustri, selleks, et hiljem mustrit vaadates teada, mis number on esiküljele kirjutatud. Ta joonistab mustrid järgmise protseduuriga: enda soovitud arv kordi (võimalik, et 0k korda) valib ta alumise vasaku nurga suhtes täisarvuliste koordinaatidega punktid A ja B ja joonistab nende vahele **sirglõigu**.

Chika joonistab vaid **lubatud** sirglõike, s.t. sirglõike punktide A ja B vahel, mille jaoks ei ole olemas veel üht täisarvuliste koordinaatidega punkti C (mis erinev punktidest A ja B), mis asub samuti sirglõigul. Näiteks sirglõik punktide $(0, 0)$ ja $(2, 2)$ vahel **ei ole lubatud**, sest sellel asub punkt $(1, 1)$, aga sirglõigud $(0, 0)$ ja $(1, 1)$ vahel ning $(1, 1)$ ja $(2, 2)$ vahel on mõlemad **lubatud** ja Chika võib isegi mõlemad joonistada samas mustris. Pane ka tähele, et sirglõikudel ei ole suunda: sirglõik punktist A punkti B on **võrdne** nii iseendaga kui ka sirglõiguga, mis on joonistatud punktist B punkti A.

Chika tahab kindlaks teha, et ta tunneb kaardid ära olenemata sellest, kuidas need pööratud on. Kaart võib esialgse asendiga võrreldes olla pööratud 0, 90, 180 või 270 kraadi vastupäeva.

Sinu ülesandeks on aidata Chikal paki q kaardi jaoks mustrid disainida ja aidata hiljem neid kaarte tuvastada.

Implementatsioon

*See on kaheetapiline interaktiivne ülesanne, kus **sinu koodi jooksutatakse eraldi mõlema etapi jaoks***. Pead implementeerima kaks funktsiooni:

- Funktsioon `BuildPattern`, mis tagastab mustri, mis joonistatakse antud kaardi tagaküljele. Seda funktsiooni kutsutakse esimeses etapis välja q korda.
- Funktsioon `GetCardNumber`, mis tagastab (võimalik, et pööratud) kaardi numbri, kusjuures kaardil on esimeses etapis joonistatud muster. Seda funktsiooni kutsutakse teises etapis välja q korda.

Esimene funktsioon

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

võtab ühe argumendi n, mis on arv kirjutatud kaardi esiküljele. Sinu ülesandeks on tagasta `std::vector`, mis sisaldab lõike, mida Chika joonistab mustrina kaardi tagaküljele selle hiljem tuvastamiseks. Sirglõik on esitatud kui `std::pair` punktidest ning punkt kui `std::pair` täisarvulisi koordinaate (x, y) kaardi alumise vasaku nurga suhtes, kus $0 \leq x, y \leq 2$. Kõik Chika joonistatud lõigud peavad olema lubatud ja paarikaupa mittevõrdsed. On garanteeritud, et kõik q funktsiooni BuildPattern väljakutset tehakse erinevate argumendi n väärtustega.

Pärast kõigi q kaardi jaoks mustrite leidmist võib hindaja iga mustri jaoks teha järgmiseid operatsioone ükskõik kui palju kordi:

- Keerata tervet mustrit 0, 90, 180 või 270 kraadi vastupäeva.
- Muuta lõikude järjekorda mustri `std::vector` esituses.
- Muuta mustris olevate lõikude otspunktide järjekorda. (Lõik punktist A punkti B võib selle operatsiooni järel muutuda sellega võrdseks lõiguks punktist B punkti A.)

Teine funktsioon

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

võtab ühe argumendi, p, mis on lõikude `std::vector` ja kirjeldab mustrit, mille Chika joonistas kaardi tagaküljele, sinu funktsiooni `BuildPattern` varasemalt tagastatud väärtuse põhjal. Funktsioon peab tagastama arvu n, mis on kirjas kaardi esiküljel. Pane tähele, et muster p ei ole tingimata samas vormis, nagu enne tagastatud funktsioonilt `BuildPattern`; seda võidi muuta kolme ülalkirjeldatud operatsiooniga. Samuti on võimalik, et kaartide järjekord erineb esimeses etapis kasutatust, kuid on garanteeritud, et igat kaarti kasutatakse endiselt vaid ühe korra.

Piirangud

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$ kõigi funktsiooni `BuildPattern` väljakutsete jaoks.
- Pane tähele, et leiduvad algoritmid, millega saab luua mustrid, nii et võimalik on ära tunda $67\,000\,000$ erinevat kaarti.

Alamülesanded

1. (2 punkti): $n \leq 2$.
2. (9 punkti): $n \leq 25$.
3. (15 punkti): $n \leq 1\,000$ ja hindaja **ei pööra** mustreid etappide 1 ja 2 vahel. (Hindaja **võib** rakendada teisi operatsioone.)
4. (3 punkti): $n \leq 16\,000\,000$ ja hindaja **ei pööra** mustreid etappide 1 ja 2 vahel. (Hindaja **võib** rakendada teisi funktsioone.)
5. (24 punkti): $n \leq 16\,000\,000$.
6. (18 punkti): $n \leq 40\,000\,000$.
7. (29 punkti): Lisapiirangud puuduvad.

Näidissuhtlus

Funktsiooni väljakutse	Tagastatav väärtus	Selgitus
Algab esimene etapp.	-	-
`BuildPattern(3)`	`{{{0, 0}, {2, 1}}, {{1, 1}, {2, 0}}}`	Peame looma mustri arvu 3 jaoks kaardil suurusega 2×2. Otsustame joonistada kaks lõiku: - $(0,0)$ ja $(2,1)$ vahele, - $(1,1)$ ja $(2,0)$ vahele.
`BuildPattern(1)`	`{{{0, 1}, {0, 0}}}`	Peame looma mustri arvu 1 jaoks kaardil suurusega 2×2. Otsustame joonistada ühe lõigu: - $(0,1)$ ja $(0,0)$ vahele.
Lõppeb esimene etapp.	-	-
Algab teine etapp.	-	-
`GetCardNumber( {{{0, 0}, {0, 1}}})`	`1`	Saame 1 sirglõiguga mustri: - $(0,0)$ ja $(0,1)$ vahel. See on sama muster, mille saaksime, joonistades sirglõigu - $(0,1)$ ja $(0,0)$ vahele, mis on sama muster samas asendis, nagu see, mille tagastasime

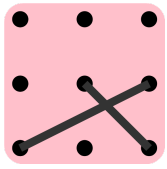
		funktsiooni `BuildPattern` teisel väljakutsel. Seega tagastame 1.

`GetCardNumber(`
`{{{1, 1}, {2, 2}},`
`{{1, 2}, {2, 0}}})`
	`3`	Saame 2 sirglõiguga mustri:
		- $(1,1)$ ja $(2,2)$ vahel,
		- $(1,2)$ ja $(2,0)$ vahel.
		See on muster, mille tagastasime esimesel funktsiooni `BuildPattern` väljakutsel, keeratud 90 kraadi vastupäeva. Seega tagastame 3.

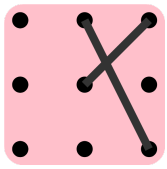
Lõppeb teine etapp.	-	-

Järgmised kolm pilti on:

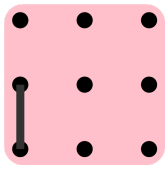
- Esimese `BuildPattern` väljakutse tagastatud muster:



- Muster, mille sai argumendina teine `GetCardNumber` väljakutse, mis on sama, mis esimene, aga 90 kraadi vastupäeva pööratud.



- Muster, mille tagastas teine `BuildPattern` väljakutse, mis on ka sama muster, mille sai teine `GetCardNumber` väljakutse argumendina sisendiks.



Näidishindaja

Ülesande manuses `Cheat.zip` antud näidishindaja `grader.cpp` loeb standardsisendist täisarvu q ja sooritab seejärel q korda järgmisi samme:

- Loe standardsisendist täisarv n.
- Kutsu välja `BuildPattern(n)` ja pane tagastatud väärtus muutujasse p.
- Kutsu välja `GetCardNumber(p)` ja kuva tagastatud väärtus standardväljundis.

Võid näidishindajat oma arvutis soovi korral muuta.

Näidishindaja oma lahendusega kompileerimiseks võid kasutada järgmist terminalikäsku:

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

kus `solution.cpp` on sinu lahendusfail, mille kavatsed esitada CMSis. Programmi käivitamiseks manuses antud näidissisendiga sisesta terminali järgmine käsk:

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

Palun pane tähele, et erinevalt näidishindajast käivitab pärishindaja CMSis sinu koodi kummagi etapi jaoks eraldi.