

Чикагийн хууралт

Бодлогын нэр	Cheat
Оролтын файл	Интерактив бодлого
Гаралтын файл	Интерактив бодлого
Хугацааны хязгаарлалт	2 секунд
Санах ойн хязгаарлалт	512 мегабайт

Чикад төрөл бүрийн эерэг бүхэл тоогоор дугаарласан q ширхэг хэзэртэй багц байгаа. Тэр Шучи Академийн оюутны зөвлөлийн найзуудтайгаа тоглохдоо тэдгээр хэзрийг хэрэглэхээр шийдсэн боловч хожих хүсэл ихтэй тул багц дахь хэзрүүдийнхээ ар талд нууц тэмдэглэгээ хийхээр шийдсэн.

Хэзрүүд бүгд 2×2 хэмжээтэй квадрат хэлбэртэй байна. Энд зүүн-доод булан $(0, 0)$ координаттай, баруун-дээд булан $(2, 2)$ координаттай байна. Чика хэзөр бүрийн ар дээр дүрс зурах ба ингэснээр дараа нь тэр дүрсээ хараад хэзрийн нүүрэн дээр ямар тоо байгааг мэдэж байх болно. Тэрээр дүрсээ дараах байдлаар зурдаг:

Тодорхой тооны удаа (магадгүй 0 удаа) тэрээр хэзрийн зүүн доод булантай харьцангуй бүхэл тоон координат бүхий A болон B гэсэн ялгаатай цэгүүдийг сонгон авч тэдгээрийг **хэрчмээр** холбоно.

Чика зөвхөн **зөв** хэрчмийг зурдаг. Зөв гэдэг нь A ба B -г холбосон хэрчим дээр, бүхэл тоон координат бүхий өөр ямар ч C цэг (A болон B -гээс ялгаатай) оршихгүй байхыг хэлнэ. Жишээ нь $(0, 0)$ ба $(2, 2)$ цэгүүдийн хоорондын хэрчим **буруу** байх ба учир нь тэр $(1, 1)$ цэгийг агуулна. Харин $(0, 0)$ ба $(1, 1)$ цэгүүдийг холбосон хэрчим болон $(1, 1)$ болон $(2, 2)$ цэгүүдийг холбосон хэрчмүүд хоёулаа **зөв** байх ба Чика тэр хоёрыг нэг дүрс дээр зурсан ч болно. Мөн хэрчим нь чиглэлгүй байна. Өөрөөр хэлбэл A -гаас B рүү татсан хэрчим нь өөртэйгээ болон B -гээс A руу татсан хэрчимтэй **ижилхэн** байна.

Мөн Чика хэзрүүдээ яаж ч эргүүлсэн таньдаг байхыг хүсч байгаа. Хэзрийг анхны байрлалаас нь цагийн зүүний эсрэг $0, 90, 180$ эсвэл 270 градусаар эргүүлж болно.

Таны даалгавар бол Чикагийн багцад байгаа q ширхэг хэзэрт зориулсан дүрснүүдийг зохиож, дараа нь тэдгээр хэзрийг таних явдал юм.

Хэрэгжүүлэлт

Уг интерактив бодлого нь хоёр шаттай ба шат бүр дээр таны програмыг тусад нь ажиллуулна. Та дараах хоёр функцийг хэрэгжүүлэх ёстой:

- `BuildPattern` функц нь өгөгдсөн хэзрийн ард зурах дүрсийг буцаана. Энэ функцийг эхний шатанд q удаа дуудна.
- `GetCardNumber` функц нь эхний шатанд зурагдсан дүрсийг агуулсан (эргүүлэгдсэн байж болох) хэзрүүдийн тоог буцаана. Энэ функц нь хоёр дахь шатанд q удаа дуудагдана.

Эхний функц

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

функц нь n гэсэн ганц параметр авах ба энэ нь тухайн хэзрийн нүүрэн талд бичсэн тоо юм. Та Чикагийн хэзрийн ард зурж, дараа нь таних боломжтой болгох дүрсийн хэрчмүүдийг агуулсан `std::vector` векторыг буцаана. Хэрчим нь цэгүүдийн `std::pair` байдлаар дүрслэгдэх бол цэг нь бүхэл тоон (x, y) координатуудын `std::pair` байдлаар дүрслэгдэнэ. Энд $0 \leq x, y \leq 2$ байна. Чикагийн зурсан бүх хэрчмүүд нь зөв бөгөөд хос хосоороо ялгаатай байна. `BuildPattern`-ийн бүх q ширхэг дуудалтын үед n параметрийн ялгаатай утгуудыг хүлээн авах нь баталгаатай байна.

q ширхэг хэзрийн бүх дүрсийг хүлээж авсны дараа шалгагч нь доорх үйлдлүүдийн алийг нь ч хэдэн ч удаа дүрс бүр дээр гүйцэтгэж болно:

- Дүрсийг бүхлээр нь цагийн зүүний эсрэг $0, 90, 180$ эсвэл 270 градусаар эргүүлэх.
- `std::vector` хэлбэрээр дүрсэлсэн дүрсийн хэрчмүүдийн дарааллыг өөрчлөх.
- Дүрс дэх хэрчмийн төгсгөлийн цэгүүдийн дарааллыг өөрчлөх. (A -гаас B рүү зурагдсан хэрчим нь түүнтэй яг адилхан B -гээс A руу зурсан хэрчим болж болно.)

Хоёр дахь функц,

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

функц нь p гэсэн ганц параметр авах ба энэ нь тухайн хэзрийн ард Чикагийн зурсан дүрсний хэрчмүүдийг өмнөх `BuildPattern` функцийг дуудалт дээр суурилан тодорхойлох `std::vector` вектор байна. Энэ функц нь хэзрийн нүүрэн дээр бичсэн n тоог буцаана. p дүрс нь `BuildPattern` функцээс буцаагдсан анхны байдлаараа байх албагүй гэдгийг санаарай. Дээр өгүүлсэн гурван үйлдлүүдээр өөрчлөгдөж болно. Мөн хэзрүүдийн дараалал нь эхний шатанд өгөгдсөнөөсөө өөр байж болох боловч хэзэр тус бүрийг яг нэг удаа ашиглана гэдэг нь баталгаатай байна.

Хязгаарлалт

- $1 \leq q \leq 10\,000$.
- $1 \leq n \leq 67\,000\,000$, BuildPattern функцийн бүх дуудалтуудын хувьд.
- 67 000 000 ялгаатай хэзрийг таньж болох дүрсүүдийг байгуулах алгоритмууд байдаг гэдгийг анхаар.

Оноо

- Дэд бодлого 1 (2 оноо): $n \leq 2$.
- Дэд бодлого 2 (9 оноо): $n \leq 25$.
- Дэд бодлого 3 (15 оноо): $n \leq 1\,000$ ба шалгагч нь дүрснүүдийг 1-р болон 2-р шатны хооронд **эргүүлэхгүй**. (Шалгагч нь нөгөө хоёр үйлдлийг гүйцэтгэж **болно**.)
- Дэд бодлого 4 (3 оноо): $n \leq 16\,000\,000$ ба шалгагч нь дүрснүүдийг 1-р болон 2-р шатны хооронд **эргүүлэхгүй**. (Шалгагч нь нөгөө хоёр үйлдлийг гүйцэтгэж **болно**.)
- Дэд бодлого 5 (24 оноо): $n \leq 16\,000\,000$.
- Дэд бодлого 6 (18 оноо): $n \leq 40\,000\,000$.
- Дэд бодлого 7 (29 оноо): Ямар нэг хязгаарлалт байхгүй.

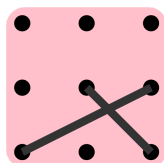
Жишээ харилцаа

Функцийн дуудалт	Буцаах утга	Тайлбар
Эхний шат эхлэх.	-	-
BuildPattern(3)	<code>{{{0, 0}, {2, 1}}, {1, 1}, {2, 0}}</code>	Бид 3 гэсэн тоонд зориулан 2×2 хэмжээтэй хөзөр дээр дүрс үүсгэх хэрэгтэй. Бид 2 хэрчим зурахаар шийдсэн: - (0,0) ба (2,1) цэгүүдийн хооронд, - (1,1) ба (2,0) цэгүүдийн хооронд.
BuildPattern(1)	<code>{{{0, 1}, {0, 0}}}</code>	Бид 1 гэсэн тоонд зориулан 2×2 хэмжээтэй хөзөр дээр дүрс үүсгэх хэрэгтэй. Бид 1 хэрчим зурахаар шийдсэн: - (0,1) ба (0,0) цэгүүдийн хооронд.
Эхний шат дуусах.	-	-
Хоёр дахь шат эхлэх.	-	-
GetCardNumber(<code>{{{0, 0}, {0, 1}}}</code>)	1	Бид 1 хэрчмээс тогтох дүрсийг хүлээн авна: - (0,0) ба (0,1) цэгүүдийн хооронд. Энэ нь доорх хэрчмийг зурахад гарах дүрстэй адилхан юм: - (0,1) ба (0,0) цэгүүдийн хооронд

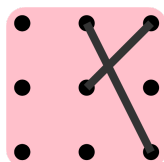
		Энэ нь эргүүлэлтгүйгээр (0 градус эргүүлсэн) BuildPattern-ын хоёр дахь дуудалтаар буцаасан дүрстэй яг ижил дүрс болох нь харагдана. Иймд бид 1-ийг буцаана.
<pre>GetCardNumber({{{1, 1}, {2, 2}}, {{1, 2}, {2, 0}}})</pre>	3	Бид 2 хэрчмээс тогтох дүрсийг хүлээн авна: - (1,1) ба (2,2) цэгүүдийн хооронд, - (1,2) ба (2,0) цэгүүдийн хооронд. Энэ нь BuildPattern функцийг эхний дуудалтаар буцаасан дүрсийг цагийн зүүний эсрэг 90 градус эргүүлэхэд гарах дүрс юм. Иймд бид 3-ыг буцаана.
Хоёрдугаар шат дуусах.	-	-

Доорх гурван зураг эдгээрийг дарааллаар нь үзүүлнэ:

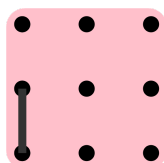
- BuildPattern-ийн эхний дуудалтын буцаасан дүрс



- Уг дүрс нь GetCardNumber функцийг хоёр дахь дуудалтын үед параметр байдлаар дамжуулагдсан ба эхний дүрсийг цагийн эсрэг 90 градус эргүүлсэнтэй ижил байна.



- Уг дүрс нь BuildPattern функцийг хоёр дахь дуудалтын дараа буцаж ирсэн ба GetCardNumber функцийг эхний дуудалтын үед аргументаар хэрэглэгдсэн байна.



Жишээ шалгагч

Бодлогын хавсралт Cheat.zip дотор байгаа grader.cpp жишээ шалгагч нь стандарт оролтоос q бүхэл тоог уншиж аваад доорх алхмуудыг q удаа хийнэ:

- n бүхэл тоог стандарт оролтоос унших.
- `BuildPattern(n)` функцийг дуудаж, буцаасан утгыг нь p хувьсагчид хадгалах.
- `GetCardNumber(p)` функцийг дуудаж, буцаасан утгыг нь стандарт гаралт руу хэвлэх.

Та хүсвэл шалгагчийг локалаар өөрчилж болно.

Жишээ шалгагчийг өөрийн кодтой хам хөрвүүлэхийн тулд та доорх командыг терминал дээр ажиллуулж болно:

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

энд `solution.cpp` нь таны CMS руу явуулах бодолт юм. Програмаа хавсралтанд байгаа жишээ оролт дээр ажиллуулах бол доорх командыг терминал дээр бичиж ажиллуулна:

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

Жишээ шалгагчаас ялгаатай нь CMS дээрх жинхэнэ шалгагч нь эхний болон хоёр дахь шатуудыг таны програмыг хоёр удаа салгаж ажиллуулж байж гүйцэтгэнэ гэдгийг анхаараарай.