

2018  
서울대학교 프로그래밍 경시대회  
Division 1

주최 및 주관



후원



서울대학교 컴퓨터공학부  
Seoul National University  
Dept. of Computer Science and Engineering



2018년 9월 9일

---

## 참가자를 위한 도움말

### 주의 사항

- 대회 시간은 13:00부터 17:00까지입니다. 대회가 진행되는 동안 인터넷 검색 및 전자기기 사용 등을 하실 수 없습니다. 단, 아래의 문서에 한해 대회 진행 중에도 참고하실 수 있으며, 책과 노트 등을 가져오신 경우 역시 참고하실 수 있습니다.
  - C/C++ reference: <https://en.cppreference.com/w/>
  - Python 2 reference: <https://docs.python.org/2/>
  - Python 3 reference: <https://docs.python.org/3.6/>
  - Java documentation: <https://docs.oracle.com/javase/8/docs/api/>
- 대회는 Baekjoon Online Judge(<https://www.acmicpc.net/>) 플랫폼을 이용하여 진행됩니다. 별도로 제공되는 계정 정보를 이용하여 로그인하신 뒤 코드 제출 및 결과 확인 등을 하실 수 있습니다.
- 제출하실 답안 코드는 C, C11, C++, C++11, C++14, C++17, Java, Python 2, Python 3, PyPy2, PyPy3 로만 작성하셔야 합니다. 단, 출제자는 C++14와 C++17을 제외한 다른 언어로 주어진 시간 제한과 메모리 제한을 지키며 올바른 답을 내는 코드를 작성할 수 있다는 보장을 하지 않습니다.
- 모든 입력은 표준 입력으로 주어지며, 모든 출력은 표준 출력으로 합니다.
- 테스트 케이스가 존재하는 문제의 경우, 테스트 케이스에 대한 출력을 모아서 하실 필요 없이, 각 테스트 케이스를 처리할 때마다 출력해도 괜찮습니다.
- **중요!!** 리턴 코드와 표준 오류(standard error, stderr) 스트림 출력에 주의하십시오. 프로세스가 0이 아닌 리턴 코드를 되돌리는 경우나 표준 오류 스트림에 출력을 하는 경우 “런타임 에러”를 받게 됩니다.
- 문제에 대한 질의 사항은 대회 페이지의 질문 기능을 사용해 주시기 바랍니다. 이 때 대답해 드리기 어려운 질문에 대해서는 “답변을 드릴 수 없습니다”로 대답될 수 있으므로 유의하십시오.
- 이 문제지는 참고용입니다. 문제지에 적힌 문제와 플랫폼에 올라온 문제가 다를 경우, 플랫폼의 문제가 우선합니다. 특히 색이 들어간 이미지는 플랫폼에서 보시기 바랍니다.

## 채점 결과에 대하여

맞았습니다!! 제출하신 답안이 모든 테스트 데이터를 정해진 시간 안에 통과하여 정답으로 인정되었음을 의미합니다.

틀렸습니다 제출하신 답안 프로그램이 테스트 데이터에 대해 생성한 출력이 출제자의 정답과 일치하지 않음을 의미합니다.

컴파일 에러 제출하신 답안 프로그램을 컴파일하는 도중 오류가 발생하였음을 의미합니다.

런타임 에러 제출하신 답안 프로그램을 실행하는 도중 프로세스가 비정상적으로 종료되었음을 의미합니다.

시간 초과 제출하신 답안 프로그램이 정해진 시간 안에 종료되지 않았음을 의미합니다.

출력 형식이 잘못되었습니다 답을 올바르게 구했으나, 답안 프로그램의 출력 형식이 문제에 나와 있는 출력 형식과 다름을 의미합니다. 줄 뒤에 의미 없는 공백을 두 칸 이상 출력하거나 의미 없는 빈 줄을 출력할 경우 이 결과를 받을 수 있습니다.

출력 초과 답안 프로그램이 정답에 비해 너무 많은 출력을 했음을 의미합니다. 이 결과는 틀렸습니다와 같은 의미를 갖습니다.

만약 여러 가지의 원인으로 인해 “맞았습니다!!”가 아닌 다른 결과를 얻으셨다면, 그 중 어떤 것도 결과가 될 수 있습니다. 예를 들어 답도 잘못되었고 비정상적인 동작도 수행하는 코드를 제출하신 경우 대부분 “런타임 에러”를 받으시게 되지만, 경우에 따라서 “틀렸습니다”를 받을 수도 있습니다.

문제별 시간 제한은 다음과 같습니다.

- A : 1.0 s
- B : 2.0 s
- C : 1.0 s
- D : 2.0 s
- E : 1.0 s
- F : 2.0 s
- G : 3.0 s
- H : 1.0 s
- I : 1.0 s
- J : 2.0 s
- K : 2.0 s
- L : 0.5 s

문제별 메모리 제한은 모든 문제에 대해 512MB로 동일합니다.

## A. 달빛 여우

관악산 기슭에는 보름달을 기다리는 달빛 여우가 한 마리 살고 있다. 달빛 여우가 보름달의 달빛을 받으면 아름다운 구미호로 변신할 수 있다. 하지만 보름달을 기다리는 건 달빛 여우뿐만이 아니다. 달빛을 받아서 멋진 늑대인간이 되고 싶어하는 달빛 늑대도 한 마리 살고 있다.

관악산에는 1번부터  $N$ 번까지의 번호가 붙은  $N$ 개의 나무 그루터기가 있고, 그루터기들 사이에는  $M$ 개의 오솔길이 나 있다. 오솔길은 어떤 방향으로든 지나갈 수 있으며, 어떤 두 그루터기 사이에 두 개 이상의 오솔길이 나 있는 경우는 없다. 달빛 여우와 달빛 늑대는 1번 나무 그루터기에서 살고 있다.

보름달이 뜨면 나무 그루터기들 중 하나가 달빛을 받아 밝게 빛나게 된다. 그러면 달빛 여우와 달빛 늑대는 먼저 달빛을 독차지하기 위해 최대한 빨리 오솔길을 따라서 그 그루터기로 달려가야 한다. 이 때 달빛 여우는 늘 일정한 속도로 달려가는 반면, 지구력이 부족한 달빛 늑대는 오솔길 하나를 달빛 여우의 두 배의 속도로 달려간 뒤 다음 오솔길은 달빛 여우의 절반의 속도로 걸어가는 것을 반복한다. 달빛 여우와 달빛 늑대 모두 자신의 이동 방법을 잘 알고 있으며 각자 시간이 가장 적게 걸리는 경로로 이동한다. 따라서 둘의 이동 경로가 서로 다를 수도 있다.

출제자는 관악산의 모든 동물을 사랑하지만, 이번에는 달빛 여우를 조금 더 사랑해 주기로 했다. 그래서 달빛 여우가 달빛 늑대보다 먼저 도착할 수 있는 그루터기에 달빛을 비춰 주려고 한다. 이런 그루터기가 몇 개나 있는지 알아보자.

### 입력

첫 줄에 나무 그루터기의 개수와 오솔길의 개수를 의미하는 정수  $N, M$  ( $2 \leq N \leq 4,000, 1 \leq M \leq 100,000$ ) 이 주어진다.

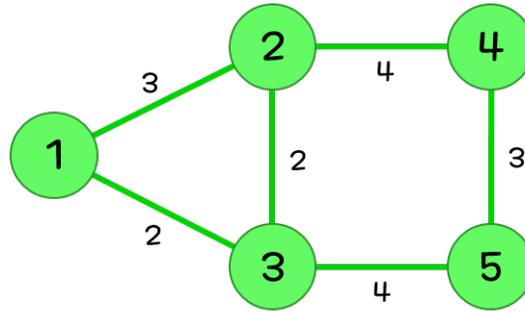
두 번째 줄부터  $M$ 개의 줄에 걸쳐 각 줄에 세 개의 정수  $a, b, d$  ( $1 \leq a, b \leq N, a \neq b, 1 \leq d \leq 100,000$ )가 주어진다. 이는  $a$ 번 그루터기와  $b$ 번 그루터기 사이에 길이가  $d$ 인 오솔길이 나 있음을 의미한다.

### 출력

첫 줄에 달빛 여우가 달빛 늑대보다 먼저 도착할 수 있는 나무 그루터기의 개수를 출력한다.

### 예제 입출력

standard input	standard output
5 6 1 2 3 1 3 2 2 3 2 2 4 4 3 5 4 4 5 3	1

**노트**

5번 그루터기에 달빛을 비추면 달빛 여우가 달빛 늑대보다 먼저 도착할 수 있다. 4번 그루터기에 달빛을 비추면 달빛 여우와 달빛 늑대가 동시에 도착한다.

## B. Cherrypick

SNUPC(SNU Patisserie Cafe)에서는 유명한 케이크를 팔고 있다. 이 케이크는  $N$ 개의 행과  $N$ 개의 열로 나눌 수 있는 정사각형 모양이다. 그리고 이 케이크를  $1 \times 1$  크기의 정사각형 조각  $N^2$ 개로 잘랐을 때, 각 조각 위에 체리가 하나씩 올려져 있다.  $i$ 행  $j$ 열 조각 위의 체리는  $C_{i,j}$ 만큼의 당도를 가지고 있다. 이 케이크는 한 번에 먹기에는 너무 크기 때문에, 각 변이 축에 평행하고 각 꼭지점의  $x$ 좌표와  $y$ 좌표가 정수인 정사각형 모양으로 잘라내어 판매하고 있는데, 잘라낸 정사각형의 한 변의 길이를  $X$ 라고 했을 때, 잘라낸 조각을  $X^2$ 원에 판매한다.

이 가게에 슈퍼 부자 Corea가 방문하였다. Corea는 이전 가게에서 이미 너무 많은 음식을 먹었기 때문에, 케이크를 산 다음 그 위에 있는 체리 중 가장 단 체리 한 조각만을 먹으려고 한다. 또, 이 케이크의 특정한 조각을 하나 골라서, 그 조각은 반드시 포함되도록 구매하려고 한다.

Corea는 돈을 낭비하는 것을 별로 좋아하지 않기 때문에, 자신이 먹게 될 체리의 당도에서 케이크를 사는데 내는 비용을 뺀 값을 최대화하려고 한다. 하지만 Corea는 그런 사소한 결정을 하는데 낭비할 시간이 없기 때문에, Corea가 케이크에 포함될 조각을 고를 동안 당신은 Corea가 선택할 수 있는 모든 경우의 수에 대해서 Corea가 먹게 될 체리의 당도에서 케이크의 비용을 뺀 값의 최댓값을 계산해야 한다.

### 입력

첫 번째 줄에 전체 케이크의 행과 열의 길이  $N$  ( $1 \leq N \leq 1,000$ )이 주어진다.

두 번째 줄부터 총  $N$ 개의 줄에 각각  $N$ 개의 정수  $C_{i,j}$  ( $1 \leq C_{i,j} \leq 1000$ )가 주어지는데, 이는  $i$ 행  $j$ 열 조각 위에 놓인 체리의 당도이다.

### 출력

공백으로 구분된  $N$ 개의 정수를  $N$ 줄에 걸쳐 출력한다.  $i$ 번째 줄의  $j$ 번째 정수로 출력하는 값은  $i$ 행  $j$ 열 조각을 포함하도록 케이크를 구매했을 때 Corea가 먹게 될 체리의 당도에서 케이크의 비용을 뺀 값의 최댓값이다.

### 예제 입출력

standard input	standard output
2	1 0
2 1	3 0
4 1	
3	1 2 2
1 2 3	4 5 5
4 5 6	6 7 8
7 8 9	

## C. PPAP

bryan은 PPAP를 좋아한다. bryan은 어떻게 하면 사람들에게 PPAP를 전파할 수 있을까 고민하던 중 PPAP 문자열이라는 것을 고안하게 되었다.

PPAP 문자열은 문자열 P에서 시작하여 문자열 내의 P를 PPAP로 바꾸는 과정을 반복하여 만들 수 있는 문자열로 정의된다. 예를 들어 PPAP는 PPAP 문자열이다. 또한 PPAP의 두 번째 P를 PPAP로 바꾼 PPPAPAP 역시 PPAP 문자열이다.

문자열이 주어졌을 때, 이 문자열이 PPAP 문자열인지 아닌지를 알려 주는 프로그램을 작성하여라.

### 입력

첫 번째 줄에 문자열이 주어진다. 문자열은 대문자 알파벳 P와 A로만 이루어져 있으며, 문자열의 길이는 1 이상 1,000,000 이하이다.

### 출력

첫 번째 줄에 주어진 문자열이 PPAP 문자열이면 PPAP를, 아닌 경우 NP를 출력한다.

### 예제 입출력

standard input	standard output
PPPAPAP	PPAP
PPAPAPP	NP

## D. 사무실 이전

SNUPC(SNU Printing Company)는 서울대학교 컴퓨터공학부 학생들의 자료 인쇄를 책임지는 대기업이다. 최근 SNUPC의 사장 ntopia는 사무실이 301동에 있어 출근이 불편하다는 사원들의 의견을 받아들여 사무실을 지하철역 주변으로 옮기기로 했다.

서울 지하철에는  $N$  개의 역이 있고, SNUPC의 직원들은 그 중 서로 다른  $M$  개의 역 근처에 한 명씩 살고 있다. 지하철의 모든 역은 트리 구조로 연결되어 있어, 임의의 역에서 출발해 다른 모든 역으로 가는 경로가 유일하게 존재한다. 또한 모든 인접한 역 사이의 거리는 1로 같다. 직원들은 출근 경로의 거리의 제곱에 비례하여 스트레스를 받기 때문에, 출근 경로의 길이가 어떻게 바뀔지는 중요한 정보이다.

ntopia는  $K$  개의 역 중 한 곳으로 사무실을 옮기기로 했으나, 오랜 시간 동안 결정을 하지 못했다. 그래서 키파는 대신 모든 후보에 대해 전 직원이 출근하는 경로의 길이의 제곱의 평균을 계산해서 미리 공지해 주기로 했다.

가능한 모든 사무실 후보에 대해 모든 직원의 출근 경로의 길이의 제곱을 모두 합한 값을 구하여라. 모든 직원은 가능한 가장 짧은 경로로 출근한다.

### 입력

첫 번째 줄에 지하철역의 개수  $N(1 \leq N \leq 300,000)$ 이 주어진다. 이는 1번부터  $N$ 번까지의 번호가 붙은  $N$  개의 역이 있음을 의미한다.

이후  $N - 1$  개의 줄에 걸쳐 서로 인접한 지하철역의 정보가 주어진다. 각 줄에 두 정수  $a, b(1 \leq a, b \leq N, a \neq b)$ 가 주어지며, 이는  $a$  번 역과  $b$  번 역이 서로 인접해 있음을 의미한다. 주어지는 정보는 트리 구조임이 보장된다.

그 다음 줄에 직원들이 사는 지하철역의 수  $M(1 \leq M \leq N)$ 이 주어진다. 그 다음 줄에  $M$  개의 서로 다른 지하철역의 번호가 공백을 사이에 두고 주어진다.

그 다음 줄에 사무실을 옮기려는 후보 지하철역의 수  $K(1 \leq K \leq N)$ 가 주어진다. 그 다음 줄에  $K$  개의 서로 다른 지하철역의 번호가 공백을 사이에 두고 주어진다.

### 출력

첫 번째 줄에 모든 사무실 후보에 대한 모든 직원의 출근 경로 길이의 제곱을 모두 합한 값을 998,244,353으로 나눈 나머지를 출력한다.

## 예제 입출력

standard input	standard output
5 1 2 1 3 3 4 3 5 2 1 4 2 2 3	12
9 1 2 2 3 2 6 3 4 4 5 6 7 6 8 6 9 4 3 5 7 8 3 6 8 9	102

## E. Unary

많은 프로그래밍 언어에서는 정수에 대해 다음과 같은 두 개의 단항 연산을 지원한다.

- $-x$ :  $x$ 의 부호를 바꾼다.  $x$ 의 값이 0인 경우는 값이 변하지 않는다.
- $\sim x$ :  $x$ 의 모든 비트를 뒤집는다. 예를 들어  $x$ 의 2진수 표현이 0000 1010일 경우  $\sim x$ 는 1111 0101이 된다.

편의상 이 문제에서 다루는 프로그래밍 언어에서는 정수를 무한히 많은 비트로 표현하며, 음수를 나타낼 때 2의 보수 표현법(음수를 표현할 때 무한히 큰 2의 거듭제곱에서 그 수를 뺀 값으로 나타내는 표현법)을 쓴다고 가정한다. 이때  $\sim x$ 의 연산 결과는  $-x-1$ 과 같다.

연산	10진법	2진법
$x$	10	0000 1010
$-x$	-10	1111 0110
$\sim x$	-11	1111 0101

$x$ 의 값이 10인 경우의 예시

정수 0에 위의 두 연산을 총  $N$ 번 적용해서 정수  $M$ 을 만들고자 할 때, 가능한 연산 순서의 가짓수를 구하여라.

### 입력

첫 줄에 연산 횟수를 의미하는 정수  $N$ 과 만들고자 하는 정수  $M$  ( $0 \leq N \leq 300,000$ ,  $-300,000 \leq M \leq 300,000$ )이 주어진다.

### 출력

첫 줄에 정수 0에  $N$ 번의 연산을 적용하여  $M$ 을 만드는 방법의 가짓수를 998,244,353으로 나눈 나머지를 출력한다.

### 예제 입출력

standard input	standard output
4 0	6
7 -3	7

### 노트

1번 예시의 경우 ----0, ---~0, --~~0, ~~~~0, ~~~~0, ~~~~0의 6가지 경우가 존재한다.

2번 예시의 경우 -----0, -----0, -----0, -----0, -----0, -----0, -----0의 7가지 경우가 존재한다.

## F. 피타고라스 쌍

피타고라스 쌍은  $m > n$  인 1 이상의 자연수  $m, n$ 에 대해

$$a = m^2 - n^2$$

$$b = 2mn$$

$$c = m^2 + n^2$$

인  $(a, b, c)$ 를 말합니다. 이때  $(a, b, c)$ 는  $(m, n)$ 으로부터 생성되었다고 얘기합니다.

원시 피타고라스 쌍  $(a, b, c)$ 는 피타고라스 쌍이면서  $\gcd(a, b, c) = 1$ 인 것입니다. 당신은  $L$ 이 주어졌을 때,  $n < m \leq L$ 인  $(m, n)$ 으로부터 생성된 원시 피타고라스 쌍의 개수를 구해야 합니다.

### 입력

첫째 줄에 5,000,000,000보다 작거나 같은 양의 정수  $L$ 이 주어집니다.

### 출력

정답을 출력합니다.

### 예제 입출력

standard input	standard output
4	4

### 노트

다음과 같이 4개 있습니다:

- $(m, n) = (2, 1)$ 인 경우  $(a, b, c) = (3, 4, 5)$
- $(m, n) = (3, 2)$ 인 경우  $(a, b, c) = (5, 12, 13)$
- $(m, n) = (4, 3)$ 인 경우  $(a, b, c) = (7, 24, 25)$
- $(m, n) = (4, 1)$ 인 경우  $(a, b, c) = (15, 8, 17)$

## G. 나는 행복합니다

한국 프로야구단 이글스의 열렬한 팬인 아인따는 올해는 분명 이글스의 가을 야구를 볼 수 있을 거라는 희망에 부풀어 있다. 그의 바람대로 올해 이글스가 포스트시즌에 진출한다면 드디어 10자리의 비밀번호를 끊을 수 있게 된다!

한국 프로야구에서 비밀번호란 어떤 팀의 연도별 순위를 나열한 문자열을 말하는데, 오랜 기간 동안 포스트시즌에 진출하지 못한 팀들의 비밀번호는 놀림감이 되곤 한다. 아인따가 응원하는 이글스의 비밀번호는 5886899678이고, 그 외에도 트윈스의 6668587667, 자이언츠의 8888577 등이 유명하다. 한국 프로야구 리그는 10개의 팀으로 구성되어 있기 때문에 1위부터 10위까지 순위가 매겨진다. 이 때 10위를 기록한 해는 비밀번호에 0으로 표기한다.

현실 세계의 이글스는 고작 10자리의 비밀번호를 찍고 있지만, 이글스가 100만 자리의 비밀번호를 찍고 있는 미래를 상상하는 것은 그다지 어렵지 않다. 미래의 아인따는 모든 것에서 해탈해서 특별한 능력을 쓸 수 있게 되었다. 아인따는 어떤 기간 동안 이글스가  $A$ 등을 기록한 모든 해의 경기 내용을 조작해서 이글스를  $B$ 등으로 바꿀 수 있다. 이 때 순위가 너무 부자연스러우면 다른 팀의 팬들에게 의심을 살 수 있기 때문에, 능력을 쓰는 중간중간에 어떤 기간의 비밀번호가 얼마나 자연스러운지를 자신만의 방법으로 확인한다.

미래의 이글스의 비밀번호와 아인따가 비밀번호를 조작하는 과정이 주어질 때, 이에 따라 비밀번호를 조작하는 프로그램을 작성해 보자.

### 입력

첫 줄에 이글스의 비밀번호를 의미하는 문자열  $S$ 가 주어진다. 문자열은 숫자로만 이루어져 있으며, 문자열의 길이는 1 이상  $10^6$  이하이다.

두 번째 줄에 아인따가 비밀번호를 조작하면서 취한 행동의 수를 의미하는 정수  $Q$  ( $1 \leq Q \leq 100,000$ )가 주어진다.

세 번째 줄부터  $Q$ 개의 줄에 걸쳐 각 줄에 아인따의 행동이 한 줄에 하나씩 순서대로 주어진다. 아인따의 행동은 다음과 같은 두 가지 형식이 있다.

- $1 \ i \ j \ from \ to$  ( $1 \leq i \leq j \leq |S|$ ,  $from, to$ 는 숫자): 비밀번호의  $i$ 번째 위치부터  $j$ 번째 위치까지의 범위에 있는 숫자  $from$ 을 전부  $to$ 로 바꾼다.
- $2 \ i \ j$  ( $1 \leq i \leq j \leq |S|$ ): 비밀번호의  $i$ 번째 위치부터  $j$ 번째 위치까지의 숫자를 이어 붙여 하나의 정수로 볼 때, 이 정수를 998,244,353으로 나눈 나머지를 구한다.

두 번째 행동이 한 번 이상 주어짐이 보장된다.

### 출력

각각의 두 번째 행동에 대한 결과를 순서대로 한 줄에 하나씩 출력한다.

**예제 입출력**

standard input	standard output
5886899678	9967
7	659
2 6 9	230110972
1 2 9 8 5	
1 1 3 5 0	
2 1 6	
1 1 10 0 2	
1 5 8 8 1	
2 1 10	

**노트**

- 쿼리 1: 입력된 수 중 [6, 9] 범위의 정수는 9,967이므로 이를 출력한다.
- 쿼리 2: [2, 9] 범위에 있는 8을 5로 바꾸므로 5556599678이 된다.
- 쿼리 3: [1, 3] 범위에 있는 5를 0으로 바꾸므로 0006599678이 된다.
- 쿼리 4: [1, 6] 범위의 정수인 659를 출력한다.
- 쿼리 5: [1, 10] 범위에 있는 0을 2로 바꾸므로 2226599678이 된다.
- 쿼리 6: [5, 8] 범위에 있는 8을 1로 바꾸는데, 이 범위에는 8이 없으므로 아무 일도 일어나지 않는다.
- 쿼리 7: [1, 10] 범위의 정수는 2,226,599,678이므로 이를 998,244,353으로 나눈 나머지인 230,110,972를 출력한다.

## H. 영점사격

논산훈련소에 간 도주는 영점사격 훈련을 받게 되었다. 이 훈련에서는 중앙에 원이 그려진 표적지에 총을 세 발 쏘는데, 세 총알 자국의 외심(세 점을 동시에 지나는 원의 중심)이 중앙의 원 안에 있을 경우 훈련을 통과할 수 있다. 정확히 원의 둘레 위에 있는 것도 허용되며, 만약 세 점이 삼각형을 이루지 못하는 경우는 불합격 처리된다. 이해하기 힘든 기준이지만 훈련소는 원래 그런 곳이다.

도주는 이미 총을 두 발 쏜 상태이다. 훈련을 통과하기 위해 남은 한 발로 맞춰야 하는 영역의 넓이를 구하여라. 편의상 표적지를 원의 중심을 원점으로 하는 좌표평면으로 생각하며, 표적지의 크기는 무한하기 때문에 총알이 표적지를 벗어나는 경우는 없다고 가정한다.

### 입력

첫 줄에 표적지의 중앙에 그려진 원의 반지름을 의미하는 정수  $R(1 \leq R \leq 500)$ 과 이미 쏜 두 발의 총알 자국의 위치를 의미하는 네 개의 정수  $x_1, y_1, x_2, y_2(-500 \leq x_1, y_1, x_2, y_2 \leq 500)$ 가 공백을 사이에 두고 주어진다. 두 점의 위치는 서로 다름이 보장된다.

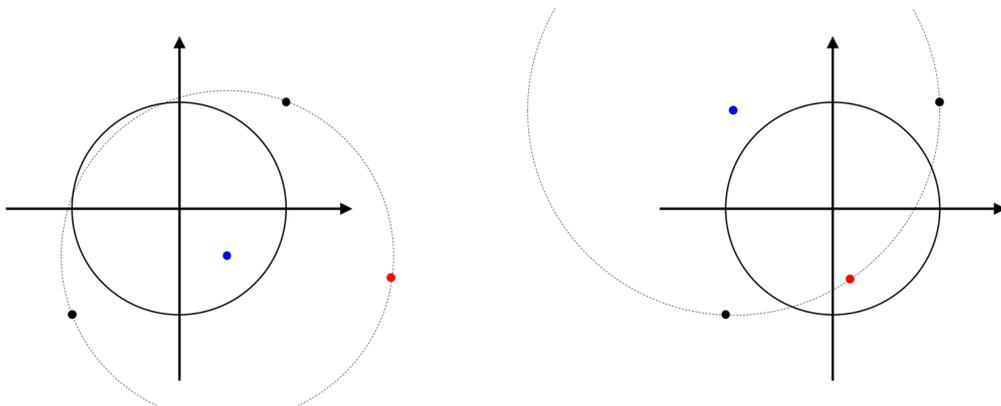
### 출력

첫 줄에 세 번째 총알로 맞춰야 하는 영역의 넓이를 출력한다. 답과  $10^{-4}$  이하의 절대 또는 상대오차가 있을 경우 정답으로 인정한다.

### 예제 입출력

standard input	standard output
1 -1 -1 1 1	13.04261075353702828801633
1 2 1 -3 3	0

### 노트



첫 번째 예시에서 훈련을 통과할 수 있는 경우와 불가능한 경우의 예시이다. 빨간 점은 세 번째 총알 자국의 위치, 파란 점은 세 점의 외심을 의미한다.

## I. 동아리방 확장

모 프로그래밍 대회에서 큰 상금을 거머쥔 상수는 좁은 동아리방에서 고통받는 SNUPS 회원들을 위해 새로운 동아리 공간을 지어 주려고  $H$ 행  $W$ 열의 격자가 그려진 직사각형 모양의 땅을 샀다.

상수는 땅을 여러 개의 방으로 나누기로 했다. 방은 크기별로 용도가 다른데, 개인 연습을 위한 한 칸짜리 방, 팀 연습을 위한 두 칸짜리 방, 스터디 및 세미나를 위한 세 칸짜리 방이 있다. 하나의 방에 속한 칸들은 상하좌우로 연결되어 있어야 하며, 하나의 칸이 여러 개의 방에 걸쳐 있을 수 없다.

상수는 어떤 방에도 속하지 않은 격자칸이 없도록 땅을 완전히 나누는 뒤, 모든 방의 둘레에 두꺼운 방음벽을 세웠다. 그리고 각각의 격자칸에 대해 상하좌우 중 몇 개의 방향이 벽으로 막혀 있는지를 기억해 두었다.

3	2	3	3
3	4	3	2
3	3	3	4

어느 날, 항상 열심히 연습하는 SNUPS 회원들의 열정에 못 이겨 동아리 공간의 모든 벽이 녹아 버리고 말았다. 상수는 기억에 의존해서 벽을 다시 세우려고 했지만, 너무 오래 전 기억이라 자신이 기억하고 있는 정보로 다시 벽을 복원할 수 있을지 의심이 들었다.

어차피 최초의 방의 배치를 완벽히 기억하고 있지 않기 때문에, 세 가지 크기의 방으로 땅을 완전히 나눌 수 있거나 하면 상수는 자신의 기억이 맞다고 생각할 것이다. 상수의 의문을 해결해 주자.

### 입력

첫 줄에 땅에 그려진 격자의 행 수와 열 수를 의미하는 정수  $H, W(1 \leq H, W \leq 100)$ 가 주어진다.

두 번째 줄부터  $H$ 줄에 걸쳐 각 줄에 0 이상 4 이하의 정수가  $W$ 개씩 주어진다.  $i$ 번째 줄의  $j$ 번째 정수는 상수가 기억하고 있는  $i$ 행  $j$ 열 칸의 벽으로 막혀 있는 방향의 개수를 의미한다.

### 출력

첫 줄에 상수가 기억하고 있는 정보로 다시 벽을 복원할 수 있다면 HAPPY, 불가능하다면 HOMELESS를 출력한다.

### 예제 입출력

standard input	standard output
3 4 3 2 3 3 3 4 3 2 3 3 3 4	HAPPY
3 4 3 2 3 3 2 3 3 3 2 3 3 4	HOMELESS

### 노트

두 번째 예시에서 방을 복원하기 위해서는 네 칸짜리 방이 필요하다.

## J. 미생물 키우기

범수는 오늘도 방에서 미생물을 키운다. 미생물은 총  $N$ 가지가 있으며 이들은 1번 미생물부터  $N$ 번 미생물까지 번호가 매겨져 있다. 범수는 다양한 미생물들을 키우려 한다. 구체적으로,  $1 \leq i \leq N$ 를 만족하는 모든 정수  $i$ 에 대해  $i$ 번 미생물을  $x_i$ 개 만들려 한다.

미생물을 만드는 데는 다음 두 가지 방법이 있다.

- $1 \leq i \leq N$ 을 만족하는 임의의 정수  $i$ 에 대해,  $i$ 번 미생물 하나를 사서 방에 넣는다.  $i$ 번 미생물을 사는 데는  $y_i$ 만큼 비용이 든다.
- $1 \leq i, j \leq N$ 을 만족하는 임의의 두 정수  $i$ 와  $j$ 에 대해,  $i$ 번 미생물에게 특수한 약을 먹여  $j$ 번 미생물을 만들도록 시킨다. 이 연산은  $i$ 번 미생물이 이미 방에 있을 때만 할 수 있으며,  $i$ 번 미생물이  $j$ 번 미생물 하나를 만드는 데  $z_{i,j}$ 만큼 비용이 든다.  $i$ 와  $j$ 는 같을 수 있다.

범수는 처음에 아무런 미생물도 가지고 있지 않다. 범수가 두 방법을 적당히 사용하여 목적을 달성하는 데 드는 최소 비용을 출력한다.

### 입력

첫째 줄에 미생물의 수  $N$  ( $1 \leq N \leq 300$ )이 주어진다.

둘째 줄에  $N$ 개의 수가 공백을 사이에 두고 주어진다. 이 중  $i$ 번째 수는 범수가 만들고자 하는  $i$ 번 미생물의 수  $x_i$  ( $1 \leq x_i \leq 10^6$ )를 의미한다.

셋째 줄에  $N$ 개의 수가 공백을 사이에 두고 주어진다. 이 중  $i$ 번째 수는  $i$ 번 미생물을 사 오는 데 드는 비용  $y_i$  ( $1 \leq y_i \leq 10^9$ )를 의미한다.

넷째 줄부터  $N$ 개의 줄에 각각  $N$ 개의 수가 주어진다. 이 중  $i$ 번째 줄의  $j$ 번째 수는  $i$ 번째 미생물이  $j$ 번째 미생물을 만들 때 드는 비용  $z_{i,j}$  ( $1 \leq z_{i,j} \leq 10^9$ )를 의미한다.

### 출력

첫째 줄에 범수가 목적을 이루기 위한 최소 비용을 출력한다.

### 예제 입출력

standard input	standard output
2	5
2 2	
2 8	
4 1	
1 1	

## 노트

위 예시의 답은 아래와 같다.

1. 1번 미생물을 사 온다. 이 때 드는 비용은  $y_1 = 2$ 이다.
2. 1번 미생물에게 약을 먹여 2번 미생물을 하나 만들게 한다. 이 때 드는 비용은  $z_{1,2} = 1$ 이다.
3. 2번 미생물에게 약을 먹여 1번 미생물을 하나 만들게 한다. 이 때 드는 비용은  $z_{2,1} = 1$ 이다.
4. 2번 미생물에게 약을 먹여 2번 미생물을 하나 만들게 한다. 이 때 드는 비용은  $z_{2,2} = 1$ 이다.

이 때 비용의 합은 5이며 이 방법이 최적이다.

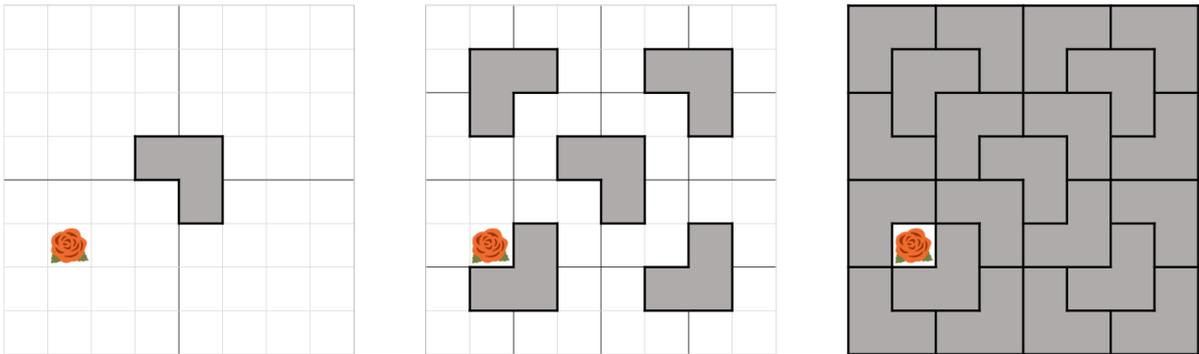
## K. 스눴시티

카젤은 도시 운영 게임을 플레이하고 있다. 이 게임에서는  $2^N \times 2^N$  크기의 땅에 다양한 오브젝트를 배치할 수 있다. 땅에는 격자가 그려져 있고, 모든 오브젝트는 격자에 맞춰서 배치해야 한다. 즉 한 오브젝트가 차지하는 영역이 어떤 격자칸과 일부만 겹쳐서는 안 된다. 오브젝트는 원하는 방향을 향하도록 회전시켜서 배치할 수도 있다.

이 게임에는 매 시간마다 금화를 생산하는 건물 오브젝트와 도시의 아름다움 수치를 올려 주는 꽃밭 오브젝트가 있다. 그 중 가장 효율이 좋은 건물 오브젝트는  $1 \times 1$  정사각형 세 개가 7자 모양으로 변을 맞대고 있는 모양의 영역을 차지하며, 가장 효율이 좋은 꽃밭 오브젝트는  $1 \times 1$  정사각형 모양의 영역을 차지한다. 카젤은 금화를 많이 벌고 싶어서 꽃밭 오브젝트는 한 개만 놓고 나머지 영역을 전부 건물 오브젝트로 채우기로 했다.

카젤은 분할정복 알고리즘을 공부한 적이 있기 때문에 꽃밭 오브젝트를 하나 배치했을 때 건물을 어떻게 배치해야 하는지도 잘 알고 있다.

1. 땅을  $2^{N-1} \times 2^{N-1}$  크기의 영역 4개로 분할한다. 이 때 한 개의 영역은 꽃밭 오브젝트가 한 칸을 차지하고 있고, 나머지 세 영역은 비어 있을 것이다.
2. 건물 오브젝트를 비어 있는 세 개의 영역에 한 칸씩 걸치도록 가운데에 배치한다.
3. 각각의 영역에 오브젝트가 차지하고 있는 칸이 한 칸씩 있게 되므로 각 영역에 대해 같은 방법으로 건물을 배치하면 된다.



8 × 8 크기의 땅의 6번째 줄 2번째 칸에 꽃밭을 배치하는 경우

하지만 도시에 건물만 가득 차 있으면 너무 삭막해 보이므로, 게임에서는 꽃밭의 사용을 권장하기 위해 일일 퀘스트를 도입했다. 매일 아침 격자칸을 한 칸 지정해 주고, 해당 격자칸에 꽃밭을 배치하면 큰 보상을 주기로 한 것이다. 카젤은 일일 퀘스트를 깨고 싶었지만, 건물을 없앴다가 다시 짓는 것은 시간이 아주 오래 걸리기 때문에 어려움에 처했다.

그나마 다행인 점은 건물을 제자리에서 회전시키는 데에는 시간이 걸리지 않는다는 점이다. 유저는 건물 하나를 선택하고 해당 건물을 포함하는  $2 \times 2$  정사각형 영역의 중심을 기준으로 시계 또는 반시계 방향으로 90도씩 회전시킬 수 있다. 물론 건물을 회전시켰을 때 다른 오브젝트가 차지하고 있는 영역과 겹치는 경우는 회전시킬 수 없다.

카젤은 일일 퀘스트가 주어지면 기존의 꽃밭을 삭제하고 건물들을 열심히 회전시켜서 지정된 위치를 빈 칸으로 만든 뒤, 그 위치에 새로 꽃밭을 배치해서 퀘스트를 완료하려고 한다. 비록 건물을 회전시키는 데에 시스템상으로 시간이 걸리지는 않지만 건물을 클릭하고 회전 버튼을 누르는 것은 너무 귀찮은 작업이기에, 카젤은 되도록 적은 횟수의 회전에 퀘스트를 완료하고 싶어한다.



건물 회전 예시

최초의 꽃밭의 위치와 이후  $M$  일간 일일 퀘스트로 지정된 격자칸들이 주어질 때, 카젤이 모든 일일 퀘스트를 순서대로 완료할 수 있는지, 가능하다면 최소 몇 번의 회전이 필요한지 구해 보자.

### 입력

첫 줄에 땅의 크기를 의미하는 정수  $N$  과 일일 퀘스트가 주어지는 날의 수를 의미하는 정수  $M$  ( $1 \leq N \leq 40$ ,  $1 \leq M \leq 100,000$ ) 이 주어진다. 땅의 크기는  $2^N \times 2^N$  임에 유의하라.

둘째 줄에 최초의 꽃밭의 위치를 의미하는 두 개의 정수  $a, b$  ( $1 \leq a, b \leq 2^N$ ) 가 주어진다. 이는 땅에 그려진 격자의  $a$  번째 줄  $b$  번째 칸에 꽃밭 오브젝트가 있음을 의미한다.

세 번째 줄부터  $M$  개의 줄에 걸쳐 각 줄에 각 날짜마다 일일 퀘스트로 주어진 칸의 위치를 의미하는 두 개의 정수가 둘째 줄과 같은 형식으로 주어진다.

### 출력

카젤이  $M$  일간의 일일 퀘스트를 모두 순서대로 완료할 수 있다면 첫 줄에 모든 퀘스트를 완료하기 위해 필요한 최소 회전 횟수를 출력한다. 만약 불가능하다면 첫 줄에 WA Machine을 출력한다.

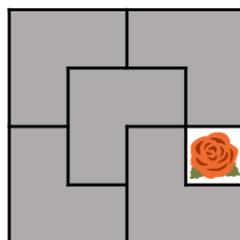
### 예제 입출력

standard input	standard output
1 2 1 1 1 2 2 1	3
2 2 3 4 1 3 2 4	5

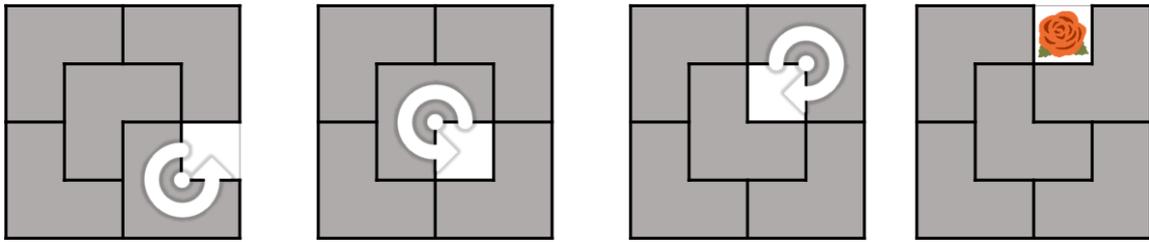
### 노트

첫 번째 예시는 건물이 하나뿐이며, 첫 날에는 시계 방향으로 한 번, 둘째 날에는 시계 또는 반시계 방향으로 두 번 회전시키면 된다.

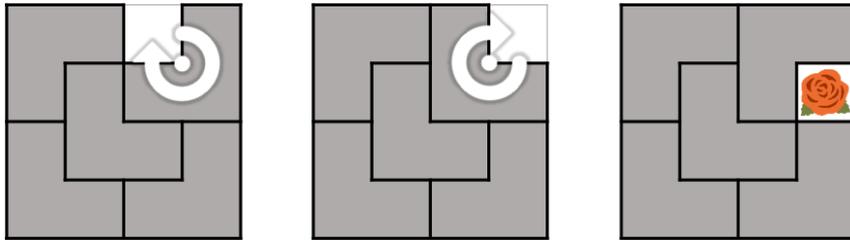
두 번째 예시의 최초의 건물과 꽃밭의 배치는 다음과 같다.



1일차 일일 퀘스트는 아래와 같은 방법으로 완료할 수 있다.



2일차 일일 퀘스트는 아래와 같은 방법으로 완료할 수 있다.



## L. 뚜루루 뚜루

최근 아기 석환이라는 노래가 큰 인기를 끌고 있다. 이 노래는 귀여운 아기 석환 캐릭터가 등장하는 동영상과 중독성 있는 뚜루루 뚜루 후렴구로 전국 각지의 학생들과 직장인들의 마음을 사로잡았다.

강남의 직장인 gs12117 역시 이 노래에 흠뻑 빠졌다. 특히 이 노래의 후렴구에 중독된 gs12117은  $R$  줄  $C$  칸으로 나뉘어진 종이에 뚜루루 뚜루 후렴구를 계속해서 적어 나가기 시작했다. 후렴구를 적을 때는 종이의 첫 줄 가장 왼쪽 칸에서부터 오른쪽으로 한 칸에 한 글자씩 뚜, 루, 루, 뚜, 루, 루, 뚜를 순서대로 적어 나가며, 한 줄의 가장 오른쪽 칸에 도달하면 다음 줄의 가장 왼쪽 칸으로 넘어간다.

뚜	루	루	뚜	루	뚜	루
루	뚜	루	뚜	루	루	뚜
루	뚜	루	루	뚜	루	뚜
루	루	뚜	루	뚜	루	루
뚜	루	뚜	루	루	뚜	루
뚜	루	루	뚜	루	뚜	루
루	뚜	루	뚜	루	루	뚜

7 × 7 종이에에서의 예시

gs12117은 이 종이에 뚜루루 뚜루 경로를 많이 찾으려고 한다. 뚜루루 뚜루 경로란 임의의 칸에서 출발해서 이미 방문한 칸을 다시 방문하지 않도록 상하좌우로 이동하면서 각 칸에 적힌 글자를 순서대로 읽었을 때, 그 결과가 정확히 뚜루루 뚜루가 되는 경로를 말한다.

종이의 크기가 주어졌을 때, gs12117이 찾을 수 있는 뚜루루 뚜루 경로의 개수를 구하여라. 어떤 두 경로가 같은 순서에 다른 칸을 방문할 경우 두 경로는 서로 다른 경로이다.

### 입력

첫 줄에 종이의 줄 수와 칸 수를 의미하는 정수  $R$ 과  $C$  ( $1 \leq R, C \leq 12, 117$ )가 주어진다.

### 출력

첫 줄에 gs12117이 찾을 수 있는 뚜루루 뚜루 경로의 개수를 출력한다.

### 예제 입출력

standard input	standard output
3 4	23
5 7	162

**노트**

첫 번째 예시에서는 다음과 같은 경로들이 가능하다.

