

<CPC 2018

고려대학교 프로그래밍 교내대회

- ADVANCED DIVISION -

※ 대회가 시작되기 전까지 절대 표지를 넘기지 마세요.

< HOSTED BY >

ALPS AIKor

< SPONSORED BY >

 **STARTLINK**

 **sooho.**

 **wishket**

NAVER D²
FOR DEVELOPERS, BY DEVELOPERS


VCNC value creators & company

 **SAMSUNG
SOFTWARE
MEMBERSHIP**



정보대학 SW중심대학
정보보호학부·정보보호대학원



대회는 총 8문제이며, 구성은 다음과 같습니다. 총 문제지가 표지를 제외하고 15쪽인지 확인하시길 바랍니다.

A	근우의 다이어리 꾸미기	300
B	Back to the Bones	400 (100)
C	등산	500
D	하노삼의 탑	600 (150)
E	XOR 포커	700
F	종이 자르기	700 (150)
G	실시간 내비게이션	700 (250)
H	기묘한 여행계획	900

문제 이름 옆에 있는 수는 문제의 전체 점수를, 괄호 안의 점수는 서브태스크의 점수를 의미합니다.

문제지는 참가자의 편의를 위해 제공되었을 뿐, 웹사이트의 내용이 공식 문제입니다. 인쇄본과 사이트의 내용 중 상이한 부분이 있을 경우, 사이트에 기재된 설명을 참고하시길 바랍니다.

기타 대회 관련 설명은 대회 사이트의 ‘유의 사항’을 참고하시길 바랍니다.

A. 근우의 다이어리 꾸미기 (300 점)

시간 제한: 1 초
메모리 제한: 256 MB

곧 2018년이 끝나고, 2019년이 온다. 근우는 2019년에는 꼭 다이어리를 쓰기로 했다. 하지만, 처음 써보는 다이어리에 쓸 내용이 없어 고민하던 중 자신의 목표 연봉을 다이어리 앞에 쓰기로 했다.

다이어리를 쓰는 사람은 알겠지만 예쁜 다이어리의 핵심은 스티커다. 그렇기 때문에 근우는 목표 연봉을 손으로 쓰지 않고, 스티커로 붙이려고 한다. 목표연봉이 100이라면 [1] [0] [0]과 같이 붙이는 것이다. [1]이란 1이 써져있는 스티커로, 다른 숫자에 대해서도 동일한 규칙이 적용된다.

근우는 자신의 연봉 최댓값이 N 임을 안다. 그렇기에 근우는 0부터 N 까지의 수를 하나씩 스티커를 통해 모두 표현하고자 한다. 최댓값 N 이 10이면 만드는 과정은 다음과 같다.

- 스티커 더미에서 [0] 하나를 가져와 0을 표현하고, ([0]), 사용한 스티커를 스티커 더미로 되돌린다.
- 스티커 더미에서 [1] 하나를 가져와 1을 표현하고, ([1]), 사용한 스티커를 스티커 더미로 되돌린다.
- 9까지 마찬가지로 방법으로 표현할 수 있다.
- 스티커 더미에서 [0] 하나와 [1] 하나를 가져와 10을 표현한다 ([1] [0]). 이후 사용한 스티커 [0]과 [1]을 스티커 더미로 되돌린다.

그러므로 N 이 10 이면 스티커가 [0]부터 [9]까지 1개씩만 있으면 모두 표현할 수 있다.

필요한 스티커를 사러 고려대 하스퀘어 유니스토어에 도착한 근우는 고민이 생겼다. 스티커 팩에는 [0]부터 [9]까지 스티커가 한 장씩 밖에 없으면서 생각보다 너무 비싼 것이다! 그렇기에 근우는 0부터 N 까지 모든 수를 하나씩 표현할 수 있게 최소한의 스티커 팩만 사려고 한다.

근우는 매우 똑똑하지만, 스티커 팩 가격에 충격을 받아 계산할 수 없는 상태가 돼버렸다. 여러분이 근우의 최대 목표액 N 이 주어졌을 때, 근우가 필요한 최소 스티커 팩의 개수를 구해주자.

입력 형식

첫 번째 줄에 근우의 연봉 최댓값을 의미하는 정수 N 이 주어진다. ($0 \leq N \leq 1,000,000,000$)

출력 형식

첫 번째 줄에 근우가 0부터 N 까지 스티커로 표현하기 위해 구매해야 하는 스티커 팩의 최소 개수를 출력한다.

예제

표준 입력	표준 출력
88	2

B. Back to the Bones (400 점)

시간 제한: 1 초
메모리 제한: 256 MB

전설에 따르면 주사위야말로 진정한 실력을 판가름할 수 있는 도구라고 한다. 그중 1부터 6까지의 수가 적혀있는 정육면체 주사위가 그 중에서도 으뜸이라고 전해진다. 정육면체 주사위를 던졌을 때 각 면이 나올 확률은 $1/6$ 로 같다. 상현이는 주사위의 시험을 받게 되었다. 정육면체 주사위 N 개를 굴려, 나온 눈의 합이 K 이상이 되어야 한다.

그러나 상현이는 주사위 컨트롤 실력이 아직 부족하므로, 주사위 N 개를 굴린 다음 마음에 안 드는 눈을 가진 주사위들을 선택해서 다시 한 번 굴릴 수 있는 기회를 얻었다. 여기서 어느 주사위도 굴리지 않아도 괜찮다.

상현이는 고민이 든다. 어떤 주사위를 선택해야 모든 눈의 합이 K 이상이 될 확률이 가장 높을까? 상현이는 실력을 판가름하기 앞서 확률을 계산해보기 위해 여러분들에게 도움을 청했다.

입력 형식

입력의 첫 줄에는 테스트 케이스의 개수를 의미하는 정수 T 가 주어진다. ($1 \leq T \leq 1,000$)

각 테스트 케이스는 두 줄로 이루어져 있으며 테스트 케이스 사이에 빈 줄은 없다.

각 테스트 케이스의 첫째 줄에는 상현이가 가지고 있는 정육면체 주사위의 개수와 눈의 합을 최소 목표치를 의미하는 두 정수 N 과 K 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 20$, $N \leq K \leq 6N$)

각 테스트 케이스의 둘째 줄에는 N 개의 정수 a_1, a_2, \dots, a_N 이 공백으로 구분되어 주어진다. a_i 는 i 번째 정육면체 주사위를 던져서 나온 눈을 의미하며, 1 이상 6 이하의 자연수이다.

출력 형식

출력은 각 테스트 케이스별로 두 줄로 이루어진다. 그러므로 총 $2T$ 개의 줄에 걸쳐서 출력을 해야 한다. 각 테스트 케이스 별로 정육면체 주사위를 적절히 선택해서 다시 굴린 후, 눈의 합이 K 이상이 될 확률의 최댓값이 p 라고 하자.

각 테스트 케이스마다 첫째 줄에는 정수 $6^N \times p$ 를 출력한다. 이 값은 32비트 정수에 들어가기에는 매우 클 수 있다.

각 테스트 케이스 별로 둘째 줄에는 어떻게 선택해서 던져야 위의 확률 p 가 나오는지를 의미하는 N 개의 정수 x_1, x_2, \dots, x_N 을 공백으로 구분하여 출력한다. x_i 는 i 번째 주사위를 던져야 하는 경우 1이며 그렇지 않을 경우 0이다. 확률이 p 가 되게 정육면체 주사위를 선택할 수 있는 방법이 다양하다면 그중 아무 것이나 출력해도 좋다.

서브태스크 1 (100 점)

$1 \leq N \leq 3$ 이다.

서브태스크 2 (300 점)

문제 입력에서 주어진 조건 외에 추가적인 조건이 없다.

예제

표준 입력	표준 출력
3	2
1 5	1
3	18
2 10	0 1
6 1	216
3 8	1 0 0
2 5 4	

설명

첫 번째 테스트 케이스에서는 주사위를 다시 굴리면 눈의 합이 5 이상이 될 확률이 $1/3$ 이며 굴리지 않으면 0이다.

두 번째 테스트 케이스에서는 두 번째 주사위만 굴리는 것이 최적이며 이 때의 확률은 $1/2$ 이다.

세 번째 테스트 케이스에서는 이미 눈의 합이 11이다. 첫 번째 주사위를 굴린 결과와 상관없이 눈의 합이 8 이상이므로 확률은 1이다.

C. 등산 (500 점)

시간 제한: 1 초
메모리 제한: 256 MB

주환이는 요즘 등산에 빠졌다. 주환이는 등산을 위해 지도를 가지고 있는데, 그 지도에는 각 지점의 높이와 갈 수 있는 다른 지점까지의 거리가 표시되어 있다.

주환이는 아침에 집에서 출발하여 등산을 갔다가, 오후 수업을 듣기 위해 고려대학교로 돌아와야 한다.

- A. 주환이는 지도의 임의의 지점을 골라, 그 지점을 목표로 정한다. 집 또는 고려대학교는 목표로 선택할 수 없다.
- B. 주환이가 집에서 정한 목표에 도달할 때까지는 항상 높이가 증가하는 방향으로만 이동해야 한다.
- C. 주환이가 정한 목표에 도달한 후, 고려대학교로 갈 때에는 항상 높이가 감소하는 방향으로만 이동해야 한다.
- D. 주환이는 거리 1을 움직일 때 마다 D 의 체력이 소모된다.
- E. 주환이는 정한 목표에 도달하면 높이 1당 E 의 성취감을 얻는다. 즉 높이가 h 인 목표에 도달하면 hE 의 성취감을 얻는다.

주환이는 이 등산의 가치를 (얻은 성취감) - (소모한 체력) 으로 계산하기로 하였다. 주환이를 위해 가치가 가장 높은 등산 경로를 선택해주자.

입력 형식

첫 번째 줄에 지도에 표시된 지점의 개수, 지점을 잇는 경로의 개수, 주환이의 거리 비례 체력 소모량, 높이 비례 성취감 획득량을 나타내는 정수 N , M , D , E 가 공백을 사이에 두고 주어진다. ($2 \leq N \leq 100,000$, $1 \leq M \leq 200,000$, $1 \leq D \leq 100$, $1 \leq E \leq 100$)

두 번째 줄에 N 개의 정수 h_1, h_2, \dots, h_N 이 공백으로 구분되어 주어진다. h_i 는 i 번째 지점의 높이를 의미한다. ($1 \leq h_i \leq 1,000,000$, $1 \leq i \leq N$)

세 번째 줄부터 M 개의 줄에 걸쳐 세 정수 a, b, n 이 공백으로 구분되어 주어진다. 이는 a 번 지점과 b 번 지점을 잇는 거리 n 의 양방향 경로가 있음을 의미한다. ($1 \leq a, b \leq N$, $1 \leq n \leq 100,000$)

어떤 지점에서 다른 지점으로 가는 경로가 여러 개 있을 수도 있으며 (등산로는 여러 개가 있을 수 있다), 한 지점에서 출발해 그 지점으로 돌아가는 경로가 있을 수도 있다 (쉽터에서 몇 바퀴 돌며 쉴 수도 있다).

주환이의 집은 1번 지점에 위치하고, 고려대학교는 N 번 지점에 위치하며 주환이의 집과 고려대학교의 높이는 1임이 보장된다.

출력 형식

첫 번째 줄에 주환이가 얻을 수 있는 가치의 최댓값을 출력한다. 만약 조건을 만족하는 등산 경로를 선택할 수 없다면, "Impossible"을 쌍따옴표를 제외하고 출력한다. 답이 음수일 수 있음에 유의하여라.

예제

표준 입력	표준 출력
8 13 4 9 1 4 7 3 10 2 15 1 1 2 3 3 4 2 5 6 6 7 8 2 2 3 4 6 7 2 3 6 1 4 8 3 5 1 6 8 3 5 2 5 4 4 6 3 5 3 8	15
3 2 1 1 1 1 1 1 2 5 2 3 5	Impossible

D. 하노삼의 탑 (600 점)

시간 제한: 1 초
메모리 제한: 256 MB

COSE214 알고리즘 강의를 수강하는 이세정 군은 최근 강의에서 ‘하노이의 탑’ 문제를 해결하는 방법에 대해 배웠다. 하노이의 탑의 규칙은 아래와 같다.

- 기둥은 3개이다.
- N 개의 서로 다른 크기의 원판이 쌓여 있으며, 작은 원판 위에 큰 원판이 올라갈 수 없다.
- 각 원판은 1번부터 N 번까지 번호가 있으며, 이 번호가 클수록 원판의 크기가 크다.
- 한 번에 한 개의 원판만 옮길 수 있으며 한 번의 이동에는 1초가 소요된다.
- 1번 기둥에서 3번 기둥으로 모든 원판들을 최소 횟수로 옮겨야 한다.
- (1) 원판은 어느 기둥에서 어느 기둥으로든 자유롭게 옮길 수 있다.

그런데 갑자기 옆에 앉아 있던 삼세정 군이 자신만의 규칙을 만들고 싶다면, (1) 대신 아래와 같은 조건을 적용하면 어떻게 될지 궁금해 했다.

- (2) 원판을 인접한 기둥으로만 옮길 수 있다. (1번 \leftrightarrow 2번 \leftrightarrow 3번)

반대쪽에 있던 사세정 군은 다음과 같은 규칙을 제안했다.

- (3) 원판을 다음 기둥으로만 옮길 수 있다. (1번 \rightarrow 2번 \rightarrow 3번 \rightarrow 1번)

그들은 신난다며 이 문제를 ‘하노삼의 탑’으로 명명했다. 이세정 군은 솔직히 이런 추가 조건들이 끌리지 않았지만, 그래도 인싸가 되기 위해 K 초 후에 각 원판이 어디에 배치되어 있을지 구해보기로 했다. 그를 도와주자.

입력 형식

첫째 줄에 세 정수 M , N , K 가 공백으로 구분되어 주어진다.

M 은 하노삼의 탑에 대해 적용될 규칙의 번호를 나타낸다. $1 \leq M \leq 3$ 이며, 1이라면 원래의 문제, 2라면 삼세정 군의 규칙, 3이라면 사세정 군의 규칙을 적용한다.

N 과 K 는 M 의 값에 따라 범위가 달라진다. 상세한 범위는 다음과 같다.

- $M = 1$ 인 경우: $1 \leq N \leq 60$, $0 \leq K < 2^N - 1$
- $M = 2$ 인 경우: $1 \leq N \leq 40$, $0 \leq K < 3^N - 1$
- $M = 3$ 인 경우: $1 \leq N \leq 30$, $0 \leq K < \frac{3+2\sqrt{3}}{6}(1+\sqrt{3})^N + \frac{3-2\sqrt{3}}{6}(1-\sqrt{3})^N$

출력 형식

첫 번째 줄에 N 개의 정수 a_1, a_2, \dots, a_N 을 공백으로 구분하여 출력한다. a_i 는 i 번 원판이 K 초 후 위치한 기둥의 번호이다.

서브태스크 1 (150 점)

$K \leq 1,000$ 이다.

서브태스크 2 (450 점)

문제 입력에서 주어진 조건 외에 추가적인 조건이 없다.

예제

표준 입력	표준 출력
1 3 6	1 3 3
1 4 1	2 1 1 1
2 3 6	1 3 1
2 4 1	2 1 1 1
3 3 6	2 3 1
3 4 1	2 1 1 1

설명

첫 번째 예제의 모습은 다음과 같다.

```
| | |
| | 2
1 | 3
-----
```

세 번째 예제의 모습은 다음과 같다.

```
| | |
1 | |
3 | 2
-----
```

다섯 번째 예제의 모습은 다음과 같다.

```
| | |
| | |
3 1 2
-----
```

E. XOR 포커 (700 점)

시간 제한: 2 초
메모리 제한: 512 MB

진벽이와 미야는 XOR 포커라는 게임을 하고 있다. 서로 정수가 적힌 카드를 N 장 받고, 받은 카드 중 일부를 적절히 골라 점수를 계산하여 점수가 더 높은 쪽이 이기는 게임이다.

점수를 계산하는 방법은 다음과 같다.

- 주어진 카드 중 짝수 개의 카드를 고른다. (0개는 고를 수 없다)
- 고른 카드에 적힌 수들의 XOR 값을 점수로 한다. (여러 정수의 XOR 값의 정의는 예제 밑의 ‘참고 사항’에 나와 있다.)

예를 들어서, 미야가 현재 $\{1, 2, 3, 3, 5\}$ 가 적힌 카드를 가지고 있다고 하자. $\{2, 3, 3, 5\}$ 을 고르면 $2 \oplus 3 \oplus 3 \oplus 5 = 7$ 이 점수가 된다. 똑같이 7이 되는 $\{1, 3, 5\}$ 는 원소의 개수가 홀수이기 때문에 고를 수 없다.

미야는 승부욕이 강해서 진벽이를 꼭 이기고 싶다. 미야가 승부에서 이길 수 있도록 도와주자.

입력 형식

첫 번째 줄에는 미야가 가진 카드의 개수 N 이 주어진다. ($2 \leq N \leq 100,000$)

두 번째 줄부터 N 개의 줄에 걸쳐 미야의 i 번째 카드에 적힌 수 a_i 가 주어진다. ($0 \leq a_i \leq 10^{18}$)

출력 형식

첫 번째 줄에 미야가 주어진 카드 XOR 포커를 할 때 만들 수 있는 점수의 최대값을 출력한다.

예제

표준 입력	표준 출력
5 1 2 3 3 5	7
4 8 2 4 1	15
7 765 876 961 315 346 825 283	1010

참고 사항

양의 정수들 x_1, x_2, \dots, x_N 의 XOR 값은 다음과 같이 정의된다.

XOR 값을 X 를 이진법으로 나타낼 때, x_1, x_2, \dots, x_N 중 2^k 의 자리가 1인 수가 홀수 개 있으면 X 의 2^k 의 자리는 1이며, 짝수 개 있으면 0이다.

F. 종이 자르기 (700 점)

시간 제한: 3 초
메모리 제한: 128 MB

고려대학교의 대표아싸 중 한명인 인구는 혼자놀기의 달인이다. 인구는 가로와 세로의 길이가 무한대인 종이를 무한대로 가지고 있다. 인구가 가진 종이는 너무 커서 필요한 만큼 잘라 쓴다. 인구는 이 특별한 종이를 자를 수 있는 유일한 도구인 직선칼을 가지고 있는데, 이 칼을 써서 종이를 자르면 자른 선분을 무한히 연장한 직선으로 종이가 잘린다. 인구는 다각형 모양으로 종이를 자르려고 하는데, 다음과 같은 과정을 거친다.

- 아직 자르지 않은 변 하나를 고른다.
- 그 변을 포함하는 직선을 자른다. 더 이상 자를 변이 없을 경우 종료한다.
- 종이를 그대로 놔둔 채로 1번으로 돌아간다.

위의 과정을 거쳐 자를 경우 다각형의 일부에 속하는 종이 조각과 다각형에 속하지 않는 종이 조각으로 나눌 수 있는데, 인구는 문득 그 숫자가 궁금해졌다. 그 두 수를 각각 구하시오.

단, 직선칼은 이미 나누어진 종이라도 직선상에 있으면 끝까지 자른다. 또 각 변을 연장해서 생기는 직선들과 이들이 만나 생기는 교점들에 대해, 어떤 세 점도 일직선상에 있지 않으며 어떤 세 직선도 한점에서 만나지 않음이 보장된다. 또 임의의 두 교점 사이의 거리는 10^{-6} 이상임이 보장된다.

입력 형식

첫 번째 줄에 인구가 가지고 놀 다각형의 변의 개수 N 이 주어진다. ($3 \leq N \leq 1,000$)

두 번째 줄부터 N 개의 줄에 걸쳐 다각형의 꼭짓점의 좌표가 시계 방향으로 주어지며, 각 줄마다 두 정수 x_i, y_i 가 공백으로 구분하여 주어진다. ($-10^6 \leq x_i, y_i \leq 10^6$)

입력으로 주어진 다각형의 서로 다른 두 변은 교차하지 않음이 보장된다.

출력 형식

첫 번째 줄에 다각형의 일부에 속하는 종이조각의 개수를 출력한다.

두 번째 줄에 다각형의 일부에 속하지 않는 종이조각의 개수를 출력한다.

서브태스크 1 (150 점)

주어지는 다각형의 모든 변은 x 축 또는 y 축에 평행하다.

서브태스크 2 (550 점)

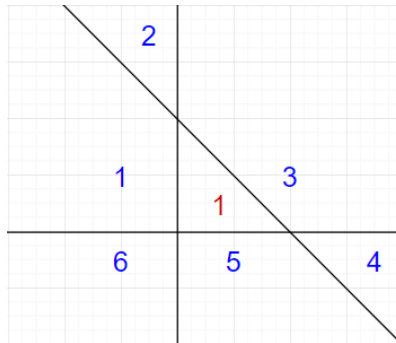
문제 입력에서 주어진 조건 외에 추가적인 조건이 없다.

예제

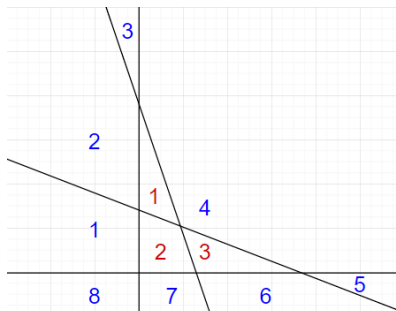
표준 입력	표준 출력
3 0 0 0 2 2 0	1 6
4 4 0 0 0 0 4 1 1	3 8

설명

첫 번째 예제의 모습은 다음과 같다.



두 번째 예제의 모습은 다음과 같다.



G. 실시간 내비게이션 (700 점)

시간 제한: 2.5 초
메모리 제한: 512 MB

우리는 Edsger W. Dijkstra가 고안한 알고리즘 덕분에 길을 쉽게 찾을 수 있지만, 그러다 보니 사람들은 몇몇 도로만 지나치게 되어 교통체증이 심해졌습니다. 이에 따라 한국도로공사는 도로의 통행시간 정보를 수정하여 내비게이션이 다른 경로를 추천하도록 하였습니다. 그러자 다른 도로들이 새롭게 막히기 시작했습니다.

새로운 도로들이 계속 막히다 보니 한국도로공사는 도로 정보를 더욱 자주 바꾸기 시작했습니다. 허나, dijkstra의 알고리즘은 도로 정보를 바꿀 때마다 $O(E \log V)$ 만큼 연산을 수행하기에 내비게이션 서버가 다운이 되어버렸습니다.

큰 문제가 생기자 내비게이션 회사는 도로 정보가 바뀔 때 빨리 최단경로를 구해주는 소프트웨어를 새로 짜기 시작했습니다. 회사는 우선 수요가 많은 강변북로 및 올림픽대로 구간만 모델링하여 최단경로를 구하려고 합니다.

한강 북쪽에 강변북로, 남쪽에 올림픽대로가 있으며 강변북로와 올림픽대로에 각각 N 개의 나들목이 있습니다. 강변북로에 있는 나들목의 코드는 가장 서쪽에 있는 것부터 순서대로 $N-1, N-2, \dots, N-N$ 이고, 올림픽대로에 있는 나들목의 코드는 서쪽부터 순서대로 $S-1, S-2, \dots, S-N$ 입니다. -가 빼기 연산자가 아님에 유의하세요. 강변북로 i 번째 도로는 나들목 $N-i$ 와 $N-(i+1)$ 를 이으며, 올림픽대로 i 번째 도로는 나들목 $S-i$ 와 $S-(i+1)$ 를 잇습니다. 한강에 있는 i 번째 다리는 나들목 $N-i$ 와 $S-i$ 을 잇습니다. 각 도로 및 다리는 드나드는 데 일정한 시간이 걸립니다.

내비게이션 서버는 맨 처음에 $3N - 2$ 개 도로의 통행 시간 정보를 갖고 있습니다. 거기에 더해, 내비게이션 서버는 한국도로공사 서버로부터 ‘특정 도로의 통행 시간이 바뀌었다’라는 정보를 받습니다. 이 정보를 토대로 내비게이션 서버는 수시로 운전자가 ‘여기에서 저기로 가는 가장 빠른 길을 알려주세요’라고 물을 때마다 재빨리 답을 구하여 보내야 합니다. 빠르게 두 종류의 쿼리를 처리하는 프로그램을 작성하세요. 단, 다리를 여러 번 건너는 경로도 답이 될 수 있습니다.

입력 형식

첫 번째 줄에 다리의 수 N 이 주어집니다. ($2 \leq N \leq 300,000$)

두 번째 줄에는 강변북로에 있는 $N - 1$ 개의 길을 지나는데 걸리는 시간 N_i 가 서쪽 도로부터 순서대로 주어집니다. ($1 \leq N_i \leq 1,000,000,000$)

세 번째 줄에는 올림픽대로에 있는 $N - 1$ 개의 길을 지나는데 걸리는 시간 S_i 가 서쪽 도로부터 순서대로 주어집니다. ($1 \leq S_i \leq 1,000,000,000$)

네 번째 줄에는 한강에 있는 N 개의 다리를 지나는데 걸리는 시간 B_i 가 서쪽 다리부터 순서대로 주어집니다. ($1 \leq B_i \leq 1,000,000,000$)

다섯 번째 줄에는 쿼리의 수 Q 가 주어집니다. ($1 \leq Q \leq 300,000$)

여섯 번째 줄부터 Q 개의 줄에는 아래와 같은 형식으로 쿼리가 주어집니다.

- 1 $a \ b$: a 번 나들목에서 b 번 나들목까지 가는데 걸리는 최소 시간을 구해라. a 와 b 는 서로 다르며, 강변북로 x 번째 나들목은 Nx 꼴로, 올림픽대로 x 번째 나들목은 Sx 꼴로 주어진다.
- 2 $a \ N_a$: 강변북로 a 번째 도로를 지나는데 걸리는 시간이 N_a 로 바뀐다. ($1 \leq a < N, 1 \leq N_a \leq 10^9$)
- 3 $b \ S_b$: 올림픽대로 b 번째 도로를 지나는데 걸리는 시간이 S_b 로 바뀐다. ($1 \leq b < N, 1 \leq S_b \leq 10^9$)
- 4 $c \ B_c$: 한강에 있는 c 번째 다리를 지나는데 걸리는 시간이 B_c 로 바뀐다. ($1 \leq c \leq N, 1 \leq B_c \leq 10^9$)

출력 형식

첫 번째 형식의 쿼리에 대하여, 쿼리의 답을 한 줄에 하나씩 출력합니다. 첫 번째 형식의 쿼리가 적어도 한 번 나오는 것은 보장됩니다.

서브태스크 1 (250 점)

$1 \leq N, Q \leq 3,000$ 을 만족한다.

서브태스크 2 (450 점)

문제 입력에서 주어진 조건 외에 추가적인 조건이 없다.

예제

표준 입력	표준 출력
7 1 2 1 1 1 2 1 1 1 3 3 1 10 9 7 12 11 8 10 6 1 N2 S4 4 6 2 1 N3 S5 3 3 8 2 4 2 1 N2 S4	10 8 14
4 1 1000000000 1 1000000000 1 1000000000 1000000000 1 1 1000000000 1 1 N1 N4	5

H. 기묘한 여행계획 (900 점)

시간 제한: 3 초
메모리 제한: 512 MB

휴학을 한 창수는 열심히 알바를 해서 B 원을 모았다. 창수는 돈도 쌓였겠다, 홍준랜드로 여행을 가기로 결정했다.

홍준랜드는 N 개의 여행지로 구성되어있으며, 각 여행지들은 2차원 격자판상에 존재한다. i 번째 여행지는 점 (x_i, y_i) 로 표현된다.

홍준랜드의 특별한 규칙이 있는데, 여행지끼리 이동할 때 꼭 택시를 타야만 한다. 택시요금은 택시거리로 책정된다. 즉 s 번 여행지에서 e 번 여행지로 이동하는 데에는 $|x_s - x_e| + |y_s - y_e|$ 만큼의 비용이 든다. 또한 홍준랜드의 각 여행지 $(i, i+1)$ 에 대해 $x_i \leq x_{i+1}$, $y_i \leq y_{i+1}$ 이 성립한다. ($1 \leq i < N$) 단, 같은 점에 두 개 이상의 여행지가 있는 경우는 없다. 그리고 홍준랜드의 택시는 출발지에서 택시에 탑승하자마자 경로상의 여행지를 볼 새도 없이 도착지에 도착한다.

창수는 B 원 이하의 비용을 써서 N 개의 여행지를 단 한 번씩 모두 방문하는 여행을 하고 싶어한다.

문득 창수는 이런 계획이 몇 가지나 존재하는지 궁금해졌다. 창수가 세울 수 있는 여행계획의 경우의 수를 1,000,000,007로 나눈 나머지를 구하시오.

입력 형식

첫 번째 줄에는 홍준랜드에 있는 여행지의 개수와 창수가 희망하는 최대 비용을 의미하는 두 정수 N 과 B 가 공백으로 구분되어 주어진다. ($1 \leq N \leq 60$, $0 \leq B \leq 2,000$)

그 이후 N 개의 줄에 걸쳐 여행지의 위치가 주어진다. i 번째 줄에는 i 번째 여행지의 정보인 두 정수 x_i, y_i 가 공백으로 구분되어 주어진다. ($|x_i| \leq B$, $|y_i| \leq B$)

출력 형식

가능한 여행 계획의 경우의 수를 1,000,000,007로 나눈 나머지를 출력한다.

예제

표준 입력	표준 출력
3 20 1 1 3 5 9 7	4

가능한 여행지 경로는 $(1, 2, 3)$, $(2, 1, 3)$, $(3, 1, 2)$, $(3, 2, 1)$ 이다.