

1. 육각형 밥그릇

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용



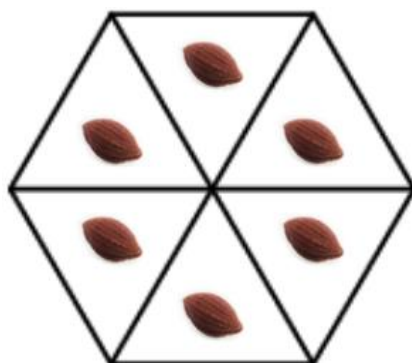
[그림 1]

대부분 사람들은 라쿤과 너구리, 레서팬더가 같은 동물인 줄 알고 있다. 하지만 셋은 서로 다른 종이다. 라쿤과 너구리를 구분할 수 있는 특징은 꼬리다. 너구리의 꼬리는 민무늬지만, 라쿤의 것은 가로 줄무늬다. 라쿤과 레서팬더를 구분할 수 있는 특징은 손발이다. 레서팬더의 손발은 납작하지만, 라쿤의 것은 손발가락이 길쭉하다. 라쿤과 너구리의 공통점은 잡식이다. 곡식, 개구리, 쥐 등 뭐든지 맛있게 먹는다. 특히 좋아하는 간식은 아몬드로, 라쿤 카페에 가면 항상 간식은 아몬드다.

오늘도 먹기 위해 사는 라쿤 "김병식"은 라쿤 카페 안에서 어슬렁거리며 손님들의 아몬드를 갈취한다. 원래 "김병식"은 아몬드를 받자마자 먹지만, 오늘만큼은 모아뒀다가 한꺼번에 폭식하고 싶어한다. 그래서 "김병식"은 아몬드를 받아서 모아둘 밥그릇을 똑딱 만들어낸다. 이 밥그릇은 다음과 같은 특징이 있다.

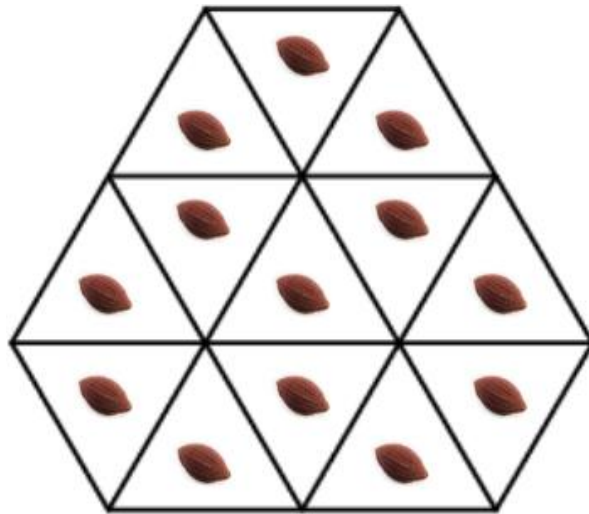
- 밥그릇의 모양은 6개의 내각이 각각 120° 인 육각형이고, 각 변의 길이는 cm 단위의 자연수이다. ([그림2] 및 [그림3] 참조)
- 밥그릇은 여러개의 칸으로 나뉘어져 있고, 각 칸은 한 변의 길이가 1cm인 정삼각형이다.
- 한 칸당 아몬드 한 개를 담을 수 있다.

예를 들어 시계 방향으로 육각형의 각 변의 길이가 1, 1, 1, 1, 1, 1 이라면 밥그릇의 모양은 다음과 같고 총 6개의 아몬드를 담을 수 있다.



[그림 2]

또다른 예로, 만약 시계 방향으로 육각형의 각 변의 길이가 1, 2, 1, 2, 1, 2 이라면 밥그릇의 모양은 다음과 같고 총 13개의 아몬드를 담을 수 있다.



[그림 3]

“김병식”은 이 밥그릇으로 손님들한테 아몬드를 받아왔다. 착한 손님들 덕분에 밥그릇의 모든 칸에 아몬드를 채웠다. 이 때 “김병식”이 얻을 수 있는 모든 아몬드의 개수를 구하라.

입력

첫째 줄에 띄어쓰기를 기준으로 6개의 정수 $a_1, a_2, a_3, a_4, a_5, a_6$ 가 주어진다. ($1 \leq a_i \leq 1,000$)

a_i 는 육각형의 각 변의 길이이고, 시계 방향으로 주어진다. 또한 위 조건을 만족하는 육각형을 만들 수 있도록 값이 주어진다고 한다. (예: “1 1 1 1 1 2”와 같은 값은 주어지지 않는다.)

출력

“김병식”이 먹을 수 있는 모든 아몬드의 개수를 구하라.

예제 입력 1

1 1 1 1 1 1

예제 출력 1 ([그림2] 참조)

6

예제 입력 2

1 2 1 2 1 2

예제 출력 2 ([그림3] 참조)

13

2. 음수의 개수

시간 제한 : 1초 | 메모리 제한 : 256MB

문제 내용

“자작”이는 음수를 찾는 능력이 탁월하다고 한다. 이를 의심한 “마루”는 직접 이를 평가하려고 한다. “마루”가 작성한 문제는 다음과 같은 형식으로 작성되었다.

- 첫번째 줄에는 수열의 길이 $N(1 \leq N \leq 100,000)$ 과 명령어의 개수 $Q(1 \leq Q \leq 100,000)$ 가 주어진다.
- 두번째 줄에는 (공백으로 구분된) 길이가 N 이고 정수로 이루어진 초기수열이 다음과 같이 주어진다.

$$n_1 \quad n_2 \quad n_3 \quad \dots \quad n_{N-1} \quad n_N$$

각 원소는 0값을 갖지 않고, 절대값은 1,000보다 작다. ($0 < |n_i| < 1000, 1 \leq i \leq N$)

- 세번째 줄부터는 각 줄마다 하나의 명령어가 주어진다(전체 Q 개의 명령). 명령어에는 다음과 같이 두 가지 유형이 있다.
 - 1로 시작하는 “1 x y” 유형의 명령어: 이 명령어는 현재 수열의 x 번째부터 y 번째까지의 범위에 들어있는 음수의 개수를 한 줄에 출력하도록 하는 명령어이다. (수열의 첫번째 원소를 1번째 원소라고 한다.)
 - 2로 시작하는 “2 x” 유형의 명령어: 이 명령어는 현재 수열의 x 번째 원소의 부호를 바꾸어서 수열을 변경하도록 하는 명령어이다. 이 명령어에 대해서는 아무런 값도 출력하지 않는다.

“마루”가 제시하는 문제의 답을 출력하는 프로그램을 작성하자.

예를 들어 다음과 같은 문제가 주어진다고 하자.

```
4 3
1 2 -3 4
1 1 3
2 2
1 1 3
```

1. 첫번째 줄에서 수열의 길이가 4로 주어졌고, 해당 길이의 초기수열은 두번째 줄에 주어졌다. 명령어의 총 개수는 3개로 주어졌고 이는 세번째 줄부터 다섯번째 줄까지 주어졌다.
2. 첫번째 명령어에 의해 1번째부터 3번째 범위, 즉 $\{1, 2, -3\}$ 에 들어있는 음수의 개수를 구하면 1개이므로 한 줄에 1을 출력한다.
3. 두번째 명령어에 의해 2번째 원소의 부호를 바꾸므로 수열은 $\{1, -2, -3, 4\}$ 로 변경된다. 이 명령어에 대해서는 아무런 값도 출력하지 않는다.
4. 세번째 명령어에 의해 1번째부터 3번째 범위, 즉 $\{1, -2, -3\}$ 에 들어있는 음수의 개수를 구하면 2개이므로 한 줄에 2를 출력한다.
5. 따라서 최종 결과는 다음과 같다.

```
1
2
```

그 외 다른 예제는 다음과 같다.

예제 입력

```
10 4
-1 -2 -3 -4 -5 -6 -7 -8 -9 -10
1 1 10
2 5
2 1
1 1 10
```

예제 출력

```
10
8
```

3. 우서 더 라이트브링어

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용

시립대 가로등의 특징은 다음과 같다.

- 시립대 가로등의 총 개수는 N 개다. i 번째 가로등의 밝기를 x_i 라고 하자. ($1 \leq i \leq N$)
- 각 가로등은 0부터 9의 밝기값을 갖는다. (0은 가로등이 꺼졌음을 의미한다.) 즉, $0 \leq x_i \leq 9$ 이다.
- 단, 첫번째 가로등은 끌 수 없다. 즉, $0 < x_1 \leq 9$ 이다.
- 모든 가로등의 밝기 합은 k 보다 작지 않다. 즉 $\sum_{j=1}^N x_j \geq k$ 이다.

시립대 빛 환경 지키미인 당신은 매일 밤마다 시립대를 한바퀴 돌면서 가로등들이 정상인지, 즉 위 조건을 모두 만족시키는지 확인한다. 어느날 누군가 가로등을 건드렸다는 제보를 접수받는 당신은 모든 가로등들을 확인한다. 다행히 가로등의 개수는 변함이 없었지만, 몇몇 가로등의 밝기가 달라져 있었다. (밝기가 달라지면 위 조건을 만족시킬 수도, 그렇지 않을 수도 있다.) 빛의 수호자-우서 더 라이트브링어-인 당신은 가로등의 밝기가 바뀌기 전의 상태를 모른채, 현재 가로등의 상태만 보고 최소 몇 개의 가로등의 밝기가 바뀌었는지 알아내야 한다.

입력

첫째 줄에 자연수 k 가 주어진다. ($1 \leq k \leq 10^9$) k 는 모든 가로등의 밝기 합 최대값이다.

둘째 줄에 N 자리 자연수 n 이 주어진다. ($1 \leq n \leq 10^{100000}$) n 의 각 자리의 숫자는 각각 가로등 하나의 밝기를 나타낸다. 예를 들어 $n = 103$ 이라면 $N = 3$ 이고, $x_1 = 1, x_2 = 0, x_3 = 3$ 이다.

출력

바뀐 가로등의 최소 개수를 구하시오.

예제 입력 1

3
11

예제 출력 1 ([그림2] 참조)

현재 상태가 조건을 만족하지 못하므로 1개 이상의 가로등이 바뀌었다는 것을 알 수 있다. 현재 상태 초기 가로등의 밝기 수열이 12,13,...,19 혹은 21,31,...,91일 수 있다. 고로, 한 개만 바꾸면 11이 된다. 그 외의 경우는 두 개를 모두 바꿔야 한다.

1

예제 입력 2

3 99

예제 출력 2

현재 가로등이 조건을 모두 만족한다. 고로, 초기 가로등의 밝기 수열이 n 과 그대로 같을 수 있으므로 아무것도 바꾸지 않는다.

13

4. 우서 더 우드커터

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용



빛의 수호자 우서는 하룻밤 사이에 전설의 나무꾼이 되어있었다. 적을 심판했던 망치는 나무를 베는 도끼가 되어 있었고, 그의 갑옷은 체크무늬 와이셔츠와 멜빵 청바지로 변해있었다. 거울 속의 우서에게서 더 이상 은빛 성기사단의 단장의 모습을 찾아볼 수 없었다. 그저 수수한 복장을 한 아저씨가 된 것이다. 그러나 우서는 이 또한 빛의 뜻이라 생각하고 자기가 할 수 있는 일을 찾아나선다. 목수가 되어서도 봉사밖에 모르는 성기사 우서는 집이 없는 아이들에게 통나무집을 선물해야겠다 마음먹는다. 우서는 다음과 같은 상황에 처해있다.

- 우서가 지어야 할 통나무집의 개수는 T 개이다. ($1 \leq T \leq 5$)
- 각 통나무집의 지을 위치가 다르고, 필요한 나무도 집을 지을 위치에서만 구해야 하고 다른 곳에서 가져올 수 없다.
 - j 번째 통나무집을 짓기 위해, 우서는 길이가 모두 같은 M_j 개의 통나무들이 필요하다. ($1 \leq M_j \leq 1,000,000$)
 - j 번째 통나무집을 짓기 위해 우서에게 N_j 개의 나무가 주어진다. ($1 \leq N_j \leq 10,000$)
 - 각 나무의 길이는 $[1, 2^{31} - 1]$ 범위에 있다. 즉, i 번째 나무의 길이를 x_i 라고 할 때, $1 \leq x_i \leq 2^{31} - 1$ ($1 \leq i \leq N_j$)이다.
 - 통나무들을 이어 붙여서 길게 만드는 것은 불가능하다.
 - 주어진 N_j 개의 나무를 모두 사용하지 않아도 된다.

각 통나무집을 지을 위치에서 최대로 자를 수 있는 통나무의 길이를 구해서 우서를 돕자.

입력

첫 줄에는 우서가 지을 집의 개수가 테스트 케이스 수 $T(1 \leq T \leq 5)$ 가 주어진다.

j 번째 테스트 케이스의 첫 줄에는 나무의 개수 $N_j(1 \leq N_j \leq 10,000)$ 와 필요한 통나무의 개수 $M_j(1 \leq M_j \leq 1,000,000)$ 가 주어진다.

그 다음 N_j 개의 줄에는 각 줄마다 나무의 길이인 정수 $x_i(1 \leq x_i \leq 2^{31} - 1)$ 가 주어진다.

출력

각 테스트 케이스마다 최대로 자를 수 있는 통나무의 길이를 정수의 형태로 출력하고, 반드시 **Test #i (1 ≤ i ≤ T)와 답을 출력한다. (아래 예제 출력 반드시 참고할 것)**

예제 입력

```
3
4 4
10
11
12
13
2 1
3
4
2 5
10
10
```

예제 출력

```
Test #1
10
Test #2
4
Test #3
3
```


5. 소수정에 덕배

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용

작년에 이어서 올해 여름도 폭폭 찌는 듯한 찜통 더위였다.

이열치열로 뉴욕에서 가장 인구 밀도가 높은 맨해튼으로 화끈한 여행을 가고 싶었으나, 학점이 부족한 덕배는 여행을 대신해 여름 계절학기를 수강하기로 한다. 덕배가 수강하려는 과목은 미래관에서 열린다. 수업 교재가 매우 무겁지만 덕배는 제본할 돈이 없다. 그러므로 수업 시간마다 정보기술관 사물함에서 꺼내 오기로 결심한다.

오래전부터 덕배는 "소수(素數, prime number : 1과 그 자신으로만 나누어지는 자연수) 정예"였다. 소수 정예는 특별한 이동 규칙을 가지고 있는데, 현재 위치에서 목표 위치까지의 거리가 소수일 때만 이동할 수 있다. 현재 덕배는 시립대 정문에 위치한다. 그리고 시립대 정문과 정보기술관 사이에는 경유할 수 있는 건물들이 존재하고 (존재하지 않을 수도 있다), 모든 건물들 사이의 거리를 알고있다. 가장 빠른 경로로 이동하고 싶은 덕배에게 시립대 정문부터 정보기술관 사이의 최단 거리를 미리 알려주자.

시립대 정문과 정보기술관, 그리고 경유할 수 있는 건물들의 좌표가 데카르트 좌표계(카르테시안 좌표계)로 주어질 때, 덕배의 규칙을 준수하며 정문에서 정기관으로 갈 수 있는 최단 경로의 길이를 구해주자. 두 점 사이의 거리는 맨해튼 거리로 구한다.

- 두 점 (x_i, y_i) 과 (x_j, y_j) 사이의 맨해튼 거리는 $|x_i - x_j| + |y_i - y_j|$ 로 정의된다.

입력

첫번째 줄에 시립대 정문의 위치 (x_{start}, y_{start}) 정보기술관의 위치 (x_{end}, y_{end}) 가 주어진다.

두번째 줄에 경유할 수 있는 건물의 개수 $N(0 \leq N \leq 4,000)$ 이 주어진다.

세번째 줄부터 N 개의 경유할 수 있는 건물의 위치 (x_i, y_i) 이 주어진다. $(1 \leq i \leq N)$

단, 주어지는 모든 좌표들은 0 이상의 자연수이고, 절대값이 5,000을 넘지 않은 정수다. 즉, $0 \leq |x_i|, |y_i| < 5,000$ 이다.

출력

덕배의 규칙을 준수하여 시립대 정문에서 정보기술관까지 갈 수 있는 방법 중 가장 짧은 거리로 갈 수 있는 길의 거리를 출력한다. 만약 덕배의 규칙을 지킬 수 없는 경우(시립대 정문에서 정보기술관까지 도달할 수 없는 경우), -1을 출력한다.

예제 입력 1

```

1 2 5 4
2
3 5
6 2

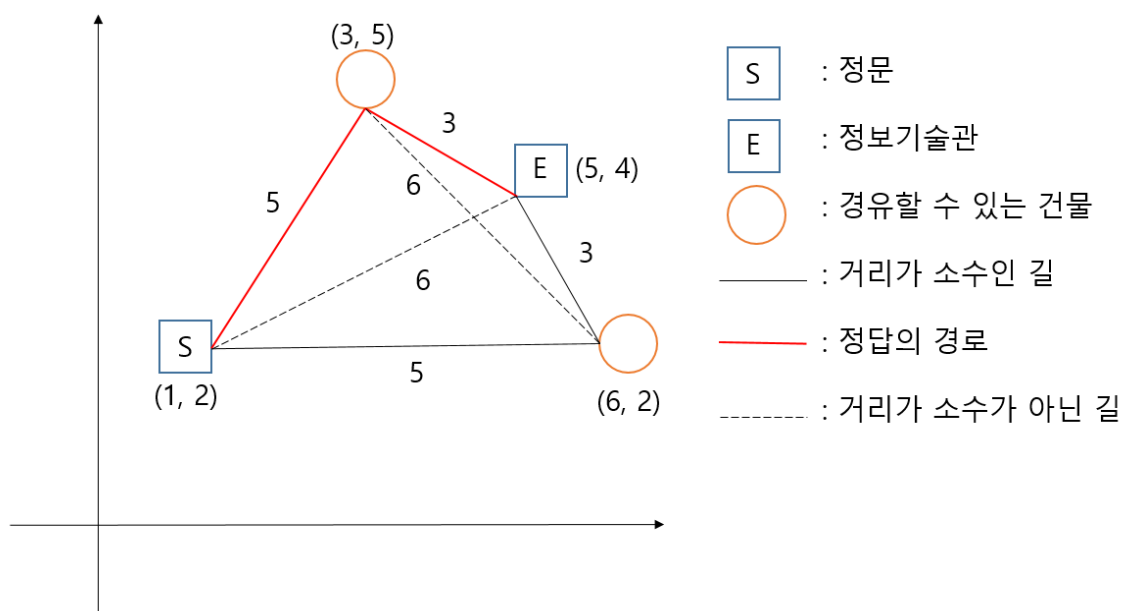
```

예제 출력 1 ([그림1] 참조)

```

8

```



[그림 1]

예제 입력 2

```

1 2 5 4
0

```

예제 출력 2

```

-1

```

6. 정보기술관의 이상한 엘리베이터

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용

0층부터 H층까지 있는 서울시립대 정보기술관에는 이상한 엘리베이터가 있다. 이 엘리베이터는 다음과 같은 특징이 있다.

- 일반적인 엘리베이터처럼 이동하기를 원하는 층이 적힌 버튼이 있지 않고, N개의 자연수가 적혀 있고 각 수마다 위/아래를 선택할 수 있는 스위치가 달려있다.
- 승객이 각 수마다 위/아래를 선택하고 "운행" 버튼을 누르면, 엘리베이터는 순서대로 N번 움직인 후 문이 열린다. 이 때, i번째 자연수를 n_i 라고 한다면 승객이 선택한 방향으로 n_i 만큼 움직인다.
- 만약 0층보다 아래로 혹은 H층보다 위로 이동하려 한다면 엘리베이터에 갇히게 된다.

예를 들어, 정보기술관이 0~5층까지 있고 3층에서 다음과 같이 승객이 방향을 설정하였다고 하면,

2	3	1	5	2
아래	위	위	아래	위

3->1->4->5->0->2

이므로, 엘리베이터는 2층에서 문이 열리게 된다.

컴퓨터과학부 11학번 김화석은 엘리베이터 중독자다. 그는 시도때도 없이 엘리베이터를 이용하는데, 수학을 못하는 화석이는 엘리베이터에 갇히는 일이 허다하다. 때마침 화석이 옆을 지나가는 당신! 수학을 못하는 화석이가 엘리베이터에 갇히지 않게 도와주자.

자연수의 개수 N과 현재 화석이 있는 층 S, 그리고 건물의 층 수 H가 주어졌을 때, 화석이가 엘리베이터를 타고 갈 수 있는 가장 높은 층을 구하는 프로그램을 작성하시오.

(화석이는 엘리베이터에 갇히지만 않는다면 현재 층보다 낮은 층으로 갈 수 밖에 없더라도 엘리베이터를 이용한다)

입력

첫째 줄에 자연수의 개수 N, 현재 층 S, 건물의 층수 H가 주어진다. ($1 \leq N \leq 500, 1 \leq H \leq 1000, 0 \leq S \leq H$)

둘째 줄에는 자연수 N개가 빈칸을 사이에 두고 차례대로 주어진다.

주어진 자연수 n_i 는 1보다 크거나 같고, H보다 작거나 같다. ($1 \leq n_i \leq H$)

출력

첫째 줄에 엘리베이터를 타고 갈 수 있는 최대 층 수를 출력한다. 만약 N번의 이동을 실행할 수 없다면(엘리베이터에 갇힌다면) -1을 출력한다.

예제 입력 1

```
3 10 14
1 2 3
```

예제 출력 1

```
14
```

예제 입력 2

```
5 3 10
4 7 2 5 8
```

예제 출력 2

```
-1
```

예제 설명

예제 1의 경우 10층 -> 9층 -> 11층 -> 14층의 경우가 가장 높은 층으로 가는 방법이다.

예제 2의 경우 3층 -> 7층 -> 0층 -> 2층 -> 7층 -> (갇힘) 의 경우만 가능하므로, -1을 출력한다.

7. 광물

시간 제한 : 2초 | 메모리 제한 : 256MB

문제 내용

아주 깊은 숲 속 깊은 어느 곳에 광물이 묻혀있다. 그 광물을 녹여 CPU에 코팅을 하게 되면 빛의 속도로 연산을 수행할 수 있게 되는데, 아직까지 아무도 그 광물을 찾지 못했다는 이야기가 전해져 내려온다. 컴퓨터 부품 판매를 업으로 하고 있는 김도매는 누구보다 빠르게 광물을 구해서 떼돈을 벌고자 한다. 하지만 광물을 구하는 일은 순탄치 않다 ...

김도매는 일정한 에너지를 가지고 있다. 마을과 마을 사이의 길을 지나갈 때마다 에너지 소비가 생기는데, 소모량은 길마다 다를 수 있다. 남은 에너지가 소모값보다 적다면, 그 길을 지나갈 수 없다. 김도매가 광물이 있는 곳까지 이동할 수 있는 가장 빠른 시간을 구하자.

입력

첫 줄에 마을의 개수 N ($2 \leq N \leq 500$), 김도매의 에너지 M ($0 \leq M \leq 10,000$), 길의 개수 K ($1 \leq K \leq 10,000$)가 공백을 사이에 두고 주어진다.

둘째 줄부터 K 줄 만큼 a, b, t, d ($1 \leq a, b \leq N, 1 \leq t \leq 100, 0 \leq d \leq 100$)가 공백을 사이에 두고 주어진다. a 와 b 는 도시의 번호, t 는 a, b 사이의 길(a, b 사이의 길은 양방향으로 이동 가능)을 이동할 때 걸리는 시간, d 는 a, b 사이의 길을 지날 때 소모되는 에너지다.

출력

김도매가 광물이 있는 곳까지 도달할 수 있는 가장 빠른 시간을 출력한다. 만약 도달할 수 없을 시, -1 을 출력한다. 단, 시작 마을의 번호는 1이고 광물이 있는 곳의 번호는 N 이다.

예제 입력 1

```
6 100 6
1 5 1 2
1 3 1 100
2 3 1 1
3 4 1 4
4 5 1 3
2 6 1 6
```

예제 출력 1

```
5
```

예제 입력 2

```
4 100 5
1 2 100 2
1 3 100 3
1 4 5 101
2 4 101 5
3 4 100 5
```

예제 출력 2

200
