

## 문제 1. 길 뿐인 넥타이

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 1초  
메모리 제한: 256MB

당신은 Just Odd Inventions라는 회사를 아는가? 이 회사는 “그저 기묘한 발명”을 하는 회사다. 여기서는 줄여서 JOI 사라고 한다.

JOI 사는 신상품 “길 뿐인 넥타이”를 개발했다. 목걸이는  $N + 1$  종류가 있어서, 각 종류에는 1번부터  $N + 1$  번까지의 번호가 붙어있다.  $i$  번째 ( $1 \leq i \leq N + 1$ ) 종류 넥타이의 길이는  $A_i$ 이다.

JOI 사는 사원을 모아서 넥타이의 테스트를 하기로 했다. 테스트에는  $N$  명의 사원이 참가했고,  $j$  번째 ( $1 \leq j \leq N$ ) 사원이 처음에 매고 있는 넥타이의 길이는  $B_j$ 이다.

테스트는 다음과 같은 순서로 진행될 예정이다.

1. 우선 테스트에 사용하지 않을 넥타이를 한 종류 골라 제외한다.
2. 각 사원은 제외된 넥타이를 뺀 넥타이 중 한 종류를 고른다. 단, 어떤 두 명도 같은 종류의 넥타이를 고를 수는 없다.
3. 마지막으로 각 사원은 매고 있던 넥타이를 벗어나서, 처음에 선택한 넥타이를 맨다.

길이  $b$ 의 넥타이를 매고 있던 사원이 길이  $a$ 의 넥타이를 맨 때  $\max\{a - b, 0\}$ 의 해당하는 기묘함을 느낀다 (여기서,  $\max\{a - b, 0\}$ 은  $a - b$ 와 0중 작지 않은 값을 의미한다.) 테스트에서 각 사원이 느낀 기묘함의 최댓값이 그 테스트의 기묘함이다.

테스트에서 제외된 넥타이가  $k$  번째 넥타이일 경우 테스트의 기묘함이 될 수 있는 값 중 최솟값을  $C_k$ 라 하자. 각 종류의 넥타이의 길이와 각 사원이 처음에 매고 있는 넥타이의 길이가 주어졌을 때,  $C_1, C_2, \dots, C_{N+1}$ 의 값을 구하는 프로그램을 작성하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 값은 정수이다.

$N$   
 $A_1 \cdots A_{N+1}$   
 $B_1 \cdots B_N$

### 출력 형식

$C_1, C_2, \dots, C_{N+1}$ 의 값을 공백으로 구분하여 표준 출력으로 첫째 줄에 출력하여라.

### 제한

- $1 \leq N \leq 200\,000$ .
- $1 \leq A_i \leq 1\,000\,000\,000$  ( $1 \leq i \leq N + 1$ ).
- $1 \leq B_j \leq 1\,000\,000\,000$  ( $1 \leq j \leq N$ ).

### 서브태스크 1 (1 점)

- $N \leq 10$ .

## 서브태스크 2 (8 점)

- $N \leq 2\,000$ .

## 서브태스크 3 (91 점)

추가 제한조건이 없다.

### 예제

standard input	standard output
3 4 3 7 6 2 6 4	2 2 1 1

예를 들어, 테스트가 다음과 같이 진행된다고 하자.

- 4번째 종류의 넥타이를 제외한다.
- 1번째 사원이 1번째 넥타이를, 2번째 사원이 2번째 넥타이를, 3번째 사원이 3번째 넥타이를 고른다.
- 각 사원이 넥타이를 맨다.

이때, 각 사원이 느낀 기묘함은 차례로 2, 0, 3이 된다. 이 테스트의 기묘함은 3이다.

사원이 고른 넥타이를 바꾸는 것으로 테스트의 기묘함을 1로 만들 수 있다. 예를 들어, 테스트가 다음과 같이 진행된다고 하자.

- 4번째 종류의 넥타이를 제외한다.
- 1번째 사원이 2번째 넥타이를, 2번째 사원이 3번째 넥타이를, 3번째 사원이 1번째 넥타이를 고른다.
- 각 사원이 넥타이를 맨다.

이때, 각 사원이 느낀 기묘함은 차례로 1, 1, 0이 된다. 이 테스트의 기묘함은 1이다.

이것이 4번째 종류의 넥타이를 제외했을 때 테스트의 기묘함의 최솟값이므로,  $C_4 = 1$ 이다.

standard input	standard output
5 4 7 9 10 11 12 3 5 7 9 11	4 4 3 2 2 2

## 문제 2. JJOOII 2

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 2초  
메모리 제한: 256MB

비타로는 비바코에게 생일 선물로 J, O, I 세 종류의 문자로 이루어진 길이  $N$ 의 문자열  $S$ 를 받았다.

$K$ 를 1 이상의 정수라고 하자.  $K$  개의 문자 J,  $K$  개의 문자 O,  $K$  개의 문자 I를 차례로 나열한 문자열을 레벨  $K$ 의 **JOI 문자열**이라고 부른다. 예를 들어, JJOOII는 레벨 2의 JOI 문자열이다.

비타로는 레벨  $K$ 의 JOI 문자열을 좋아하므로 다음 세 종류의 조작을 원하는 순서로 원하는 만큼 해서 문자열  $S$ 를 레벨  $K$ 의 JOI 문자열로 바꾸려 한다.

조작 1 문자열  $S$ 의 가장 앞에 있는 문자를 지운다.

조작 2 문자열  $S$ 의 가장 뒤에 있는 문자를 지운다.

조작 3 문자열  $S$ 의 가장 앞에도 뒤에도 있지 않은 문자를 지운다.

조작 3이 제일 귀찮기 때문에, 조작 3을 하는 횟수를 최소한으로 하여 문자열  $S$ 를 레벨  $K$ 의 JOI 문자열로 바꾸고 싶다.

길이  $N$ 의 문자열  $S$ 와 1 이상의 정수  $K$ 가 주어질 때, 문자열  $S$ 를 레벨  $K$ 의 JOI 문자열로 바꾸는 데에 필요한 조작 3의 횟수의 최솟값을 출력하는 프로그램을 작성하여라. 단, 어떻게 조작해도 문자열  $S$ 를 레벨  $K$  JOI 문자열로 만드는 것이 불가능할 때에는, 대신  $-1$ 을 출력하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다.

$N$   $K$

$S$

### 출력 형식

문자열  $S$ 를 레벨  $K$ 의 JOI 문자열로 바꾸는 데에 필요한 조작 3의 횟수의 최솟값을 출력하는 프로그램을 작성하여라. 단, 어떻게 조작해도 문자열  $S$ 를 레벨  $K$  JOI 문자열로 만드는 것이 불가능할 때에는, 대신  $-1$ 을 출력하여라.

### 제한

- $3 \leq N \leq 200\,000$ .
- $1 \leq K \leq \frac{N}{3}$ .
- $S$ 는 J, O, I 세 종류의 문자로 이루어진 길이  $N$ 의 문자열이다.

### 서브태스크 1 (1 점)

- $N \leq 21$ .

### 서브태스크 2 (12 점)

- $N \leq 3\,000$ .

### 서브태스크 3 (87 점)

추가 제한조건이 없다.

#### 예제

standard input	standard output
10 2 OJIJOIOIIJ	2

다음과 같은 조작을 통해 문자열  $S$ 를 레벨  $K$ 의 JOI 문자열로 바꿀 수 있다.

1. 처음에 조작 1을 한다. 문자열  $S$ 는 JIJJOIOIIJ가 된다.
2. 다음에 조작 2를 한다. 문자열  $S$ 는 JIJJOIOII가 된다.
3. 다음에 조작 3을 한다. 앞에서부터 두 번째 문자를 지운다. 문자열  $S$ 는 JJJOIOII가 된다.
4. 다음에 조작 4를 한다. 앞에서부터 네 번째 문자를 지운다. 문자열  $S$ 는 JJOOII가 된다.

두 번 미만으로 조작 3을 해서 문자열을 바꾸는 것은 불가능하기 때문에 2를 출력한다.

standard input	standard output
9 3 JJJO00III	0

조작을 할 필요가 없다.

standard input	standard output
9 3 IIIIO00JJJ	-1

이 입력 예제에서는, 어떻게 조작을 해도 문자열  $S$ 를 레벨 1의 JOI 문자열로 바꿀 수 없다.

## 문제 3. 스탬프 수집 3

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 2초  
메모리 제한: 1024MB

JOI 군이 사는 IOI 나라는 거대한 호수가 있는 것으로 유명하다. 오늘 호수 둘레에서 스탬프 수집 대회가 개최된다.

호수 둘레에는  $N$  개의 도장이 있어서, 시계방향으로 1번부터  $N$ 번까지의 번호가 붙어있다. 호수 둘레의 길이는  $Lm$ 이고,  $i$ 번 ( $1 \leq i \leq N$ ) 스탬프는 출발 지점으로부터 시계방향으로  $X_i m$  앞에 있다.

스탬프 수집의 각 참가자는 스탬프 수집 대회를 시작할 때 출발 지점에 모이고, 시작한 후에는 호수 둘레를 시계방향 혹은 반시계방향으로 돌 수 있다. 참가자는 도장이 있는 위치에 도착하면 그 도장을 찍어서 스탬프를 남길 수 있다. 같은 도장을 여러 번 찍을 수는 없다. 하지만 스탬프 수집 대회가 시작한 지  $T_i$ 초가 지나기 전에 도착해야 한다(대회가 시작한 지 정확히  $T_i$ 초 후에 도장이 있는 위치에 도착해도 도장을 찍을 수 있다.)

JOI 군은 이 스탬프 수집 대회의 참가자이다. JOI 군은  $1m$ 를 걷는 데  $1$ 초가 걸린다. 또한 JOI는 도장 찍기의 숙련자이기 때문에, 도장을 찍는 데 걸리는 시간은 무시해도 좋다.

도장의 개수, 호수 둘레의 길이, 각 도장이 있는 지점과 각 도장을 찍을 수 있는 시간이 주어졌을 때, JOI군이 찍을 수 있는 스탬프 개수의 최댓값을 구하는 프로그램을 작성하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 값은 정수이다.

$N$   $L$

$X_1 \cdots X_N$

$T_1 \cdots T_N$

### 출력 형식

JOI군이 찍을 수 있는 스탬프 개수의 최댓값을 표준 출력으로 첫째 줄에 출력하여라.

### 제한

- $1 \leq N \leq 200$ .
- $2 \leq L \leq 1\,000\,000\,000$ .
- $1 \leq X_i < L$  ( $1 \leq i \leq N$ ).
- $X_i < X_{i+1}$  ( $1 \leq i \leq N-1$ ).
- $0 \leq T_i \leq 1\,000\,000\,000$  ( $1 \leq i \leq N$ ).

### 서브태스크 1 (5 점)

- $N \leq 12$ .
- $L \leq 200$ .
- $T_i \leq 200$  ( $1 \leq i \leq N$ ).

### 서브태스크 2 (10 점)

- $N \leq 15$ .

### 서브태스크 3 (10 점)

- $L \leq 200$ .
- $T_i \leq 200$  ( $1 \leq i \leq N$ ).

### 서브태스크 4 (75 점)

추가 제한조건이 없다.

#### 예제

standard input	standard output
6 25 3 4 7 17 21 23 11 7 17 10 8 10	4

다음 방법으로 JOI 군은 4개의 스탬프를 모을 수 있다.

1. 반시계방향으로 2m 간다. 대회 시작 2초 후이기 때문에, 6번 도장을 찍을 수 있다.
2. 또 반시계방향으로 2m 간다. 대회 시작 4초 후이기 때문에, 5번 도장을 찍을 수 있다.
3. 시계방향으로 7m 간다. 대회 시작 11초 후이기 때문에, 1번 도장을 찍을 수 있다.
4. 또 시계방향으로 1m 간다. 대회 시작 12초 후이기 때문에, 2번 도장을 찍을 수 없다.
5. 또 시계방향으로 3m 간다. 대회 시작 15초 후이기 때문에, 3번 도장을 찍을 수 없다.

어떤 방법으로도 JOI 군이 5개 이상의 스탬프를 모을 수는 없기 때문에, 4를 출력한다.

standard input	standard output
5 20 4 5 8 13 17 18 23 15 7 10	5

JOI 군은 스탬프 수집이 시작한 후, 호수 둘레를 반시계방향으로만 돌면서 모든 스탬프를 모을 수 있다.

standard input	standard output
4 19 3 7 12 14 2 0 5 4	0

유감이지만, JOI 군은 어떻게 이동해도 스탬프를 하나도 모을 수 없다.

standard input	standard output
10 87 9 23 33 38 42 44 45 62 67 78 15 91 7 27 31 53 12 91 89 46	5

## 문제 4. 올림픽 버스

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 1초  
메모리 제한: 256MB

JOI 나라에는  $N$  개의 도시가 있고, 1번부터  $N$ 번까지의 번호가 붙어있다. 또한 도시와 도시를 단방향으로 잇는  $M$  개의 버스노선이 있고, 1번부터  $M$ 번까지의 번호가 붙어있다.  $i$ 번 버스 노선은  $U_i$ 번 도시에서  $V_i$ 번 도시까지 운행하며, 운임은  $C_i$  엔이다.  $i$ 번 ( $1 \leq i \leq M$ ) 버스 노선은  $U_i$ 번 도시 이외에서 타거나,  $V_i$ 번 도시 이외에서 내리는 것이 불가능하다. 어떤 도시부터 어떤 도시를 향하는 버스 노선은 여러 개 있을 수도 있다.

JOI 나라에는 곧 올림픽이 개최된다. JOI 나라의 통행 장관인 K 이사장은 버스 노선을 **최대 한 개** 골라 올림픽 개최 중에 그 버스 노선을  $U_i$ 번 도시에서  $V_i$ 번 도시까지 운행하는 것이 아니라,  $V_i$ 번 도시에서  $U_i$ 번 도시로 운행하는 것으로 바꾸려고 한다.  $i$ 번 버스 노선을 반대로 바꾸는 데에 드는 비용은  $D_i$ 엔이 되며 이는 K 이사장 개인 돈에서 나간다. 단, 혼란을 막기 위해 올림픽 개최 중에 버스의 방향을 바꾸는 것은 할 수 없다.

통행 장관인 K 이사장은 올림픽 개최 중에 1번 도시와  $N$ 번 도시 사이를 버스 노선을 사용해 왕복할 예정이다. 버스 노선 운임과 운행 방향을 반대로 바꾸는 데에 드는 돈의 합을 최소화하고 싶다.

도시의 개수와 버스 노선의 정보가 주어졌을 때, 운행 방향을 반대로 바꾸는 것으로 1번 도시와  $N$ 번 도시 사이를 왕복하는 데에 드는 운임 합과 운행 방향을 반대로 바꾸는 데에 드는 돈의 합의 최솟값을 구하는 프로그램을 작성하여라. 단, 어떻게 버스 노선을 골라도 1번 도시와  $N$ 번 도시 사이를 왕복할 수 없다면 -1을 출력하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 값은 정수이다.

$N$   $M$

$U_1$   $V_1$   $C_1$   $D_1$

$\vdots$

$U_M$   $V_M$   $C_M$   $D_M$

### 출력 형식

1번 도시와  $N$ 번 도시 사이를 왕복하는 데에 드는 운임 합과 운행 방향을 반대로 바꾸는 데에 드는 돈의 합의 최솟값을 표준 출력으로 첫째 줄에 출력하여라. 단, 어떻게 버스 노선을 골라도 1번 도시와  $N$ 번 도시 사이를 왕복할 수 없다면 -1을 출력하여라.

### 제한

- $2 \leq N \leq 200$ .
- $1 \leq M \leq 50\,000$ .
- $1 \leq U_i \leq N$  ( $1 \leq i \leq M$ ).
- $1 \leq V_i \leq N$  ( $1 \leq i \leq M$ ).
- $U_i \neq V_i$  ( $1 \leq i \leq M$ ).
- $0 \leq C_i \leq 1\,000\,000$  ( $1 \leq i \leq M$ ).
- $0 \leq D_i \leq 1\,000\,000\,000$  ( $1 \leq i \leq M$ ).

### 서브태스크 1 (5 점)

- $M \leq 1\,000$ .

## 서브태스크 2 (11 점)

- $M$ 은 짝수이다.
- $U_{2i-1} = U_{2i}$  ( $1 \leq i \leq \frac{M}{2}$ ).
- $V_{2i-1} = V_{2i}$  ( $1 \leq i \leq \frac{M}{2}$ ).
- $C_{2i-1} = C_{2i}$  ( $1 \leq i \leq \frac{M}{2}$ ).

## 서브태스크 3 (21 점)

- $C_i = 0$  ( $1 \leq i \leq M$ ).

## 서브태스크 4 (63 점)

추가 제한조건이 없다.

### 예제

standard input	standard output
4 5 1 2 4 4 1 3 2 1 4 3 1 2 4 1 6 1 2 4 2 5	10

2번 버스 노선의 운행 방향을 비용 1로 반대로 바꾸면, 1번 도시부터 4번 도시까지 이동에 걸리는 운임의 최솟값은 6, 4번 도시부터 1번 도시까지 이동에 걸리는 운임의 최솟값은 3이 되고, 1번 도시와 4번 도시 사이의 왕복 비용과 운행 방향을 반대로 바꾸는 데에 드는 비용의 합이 10이 된다.

1번 도시와 4번 도시 사이를 왕복하는 데에 드는 비용과 운행 방향을 반대로 바꾸는 데에 드는 비용의 합을 10보다 작게 할 수는 없기 때문에, 10을 출력하여야.

standard input	standard output
4 10 1 2 4 4 1 2 4 4 1 3 2 1 1 3 2 1 4 3 1 2 4 3 1 2 4 1 6 1 4 1 6 1 2 4 2 5 2 4 2 5	10

이 입력 예제는 서브태스크 2의 조건을 만족한다.



standard input	standard output
4 4 1 2 0 4 1 3 0 1 4 3 0 2 4 1 0 1	2

이 입력 예제는 서브태스크 3의 조건을 만족한다.

standard input	standard output
4 5 1 2 4 4 1 3 2 4 4 3 1 5 4 1 6 1 2 4 2 5	12

버스 노선의 운행 방향을 바꾸지 않아도 된다.

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

이 입력 예제는, 도시 4에서 도시 3까지 가는 버스 노선이 두 개 존재한다.

## 문제 5. 화재

입력 파일: standard input  
출력 파일: standard output  
시간 제한: 1.5초  
메모리 제한: 256MB

JOI 마을에는  $N$  개의 구역이 있고, 1번부터  $N$ 번까지의 번호가 붙어있다. 이 구역은 번호순으로 일렬로 나열되어 있다. 지금 각 구역에는 화재가 발생했고 시각 0에  $i$ 번 ( $1 \leq i \leq N$ ) 구역에 일어난 화재의 세기는  $S_i$ 이다. ( $S_i > 0$ )

시각 0에 1번 구역에서  $N$ 번 구역까지의 방향으로 바람이 불기 시작했다. 시각  $t$ 에 ( $0 \leq t$ ) 인접한 두 구역에 대해 위쪽 구역의 화재가 아래쪽 구역의 화재보다 더 강할 경우, 시각  $t+1$ 에 아래쪽 구역의 화재의 세기는 시각  $t$ 의 위쪽 구역의 화재의 세기와 같아진다. 그렇지 않을 경우, 시각  $t+1$ 에 아래쪽 구역의 화재의 세기는 시각  $t$ 와 같다. 즉, 시각  $t$  ( $0 \leq t$ )의  $i$ 번 구역의 화재의 세기를  $S_i(t)$ 라고 하자.  $1 \leq t$ 일 경우,  $S_i(t) = \max\{S_{i-1}(t-1), S_i(t-1)\}$ 이 된다. 단, 임의의  $t$  ( $0 \leq t$ )에 대해  $S_0(t) = 0$ 이라고 하고 임의의  $i$  ( $1 \leq i \leq N$ )에 대해,  $S_i(0) = S_i$ 라고 하자.

소방관인 당신은  $Q$  개의 계획을 정했다.  $Q$  개의 계획 중 하나를 실행할 예정이다.  $j$  번째 ( $1 \leq j \leq Q$ ) 계획은 시각  $T_j$ 에  $L_j \leq k \leq R_j$ 를 만족하는 모든  $k$ 번 구역의 불을 끄는 것이다. 화재의 세기가  $s$ 인 구역을 소화하는데에는  $s\ell$ 의 물이 필요하다. 즉,  $j$  번째 계획에 사용되는 물의 양은  $S_{L_j}(T_j) + S_{L_j+1}(T_j) + \dots + S_{R_j}(T_j)\ell$ 이다.

어떤 계획을 실행할지 정하기 위하여 각 계획에 필요한 물의 양을 알고 싶다.

시각 0의 화재의 세기와 계획의 정보가 주어졌을 때, 각 계획에 필요한 물의 양을 구하는 프로그램을 작성하여라.

### 입력 형식

표준 입력에서 다음과 같은 형식으로 주어진다. 모든 값은 정수이다.

$N$   $Q$

$S_1 \dots S_N$

$T_1$   $L_1$   $R_1$

$\vdots$

$T_Q$   $L_Q$   $R_Q$

### 출력 형식

표준 출력으로  $Q$ 개의 줄을 출력하여라.  $j$  번째 ( $1 \leq j \leq Q$ ) 줄에는  $j$  번째 계획에 필요한 물의 양을 출력하여라.

### 제한

- $1 \leq N \leq 200\,000$ .
- $1 \leq Q \leq 200\,000$ .
- $1 \leq S_i \leq 1\,000\,000\,000$  ( $1 \leq i \leq N$ ).
- $1 \leq T_j \leq N$  ( $1 \leq j \leq Q$ ).
- $1 \leq L_j \leq R_j \leq N$  ( $1 \leq j \leq Q$ ).

### 서브태스크 1 (1 점)

- $N \leq 200$ .
- $Q \leq 200$ .

## 서브태스크 2 (6 점)

- $T_1 = T_2 = \dots = T_Q$ .

## 서브태스크 3 (7 점)

- $L_j = R_j$  ( $1 \leq j \leq Q$ ).

## 서브태스크 4 (6 점)

- $S_i \leq 2$  ( $1 \leq i \leq N$ ).

## 서브태스크 5 (80 점)

추가 제한조건이 없다.

### 예제

standard input	standard output
5 5	21
9 3 2 6 5	39
1 1 3	33
2 1 5	9
3 2 5	27
4 3 3	
5 3 5	

- 시각 0의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 3, 2, 6, 5이다.
- 시각 1의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 9, 3, 6, 6이다. 또한, 첫 번째의 계획에 필요한 물의 양은  $9 + 9 + 3 = 21\ell$ 이다.
- 시각 2의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 9, 9, 6, 6이다. 또한, 첫 번째의 계획에 필요한 물의 양은  $9 + 9 + 9 + 6 + 6 = 39\ell$ 이다.
- 시각 3의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 9, 9, 9, 6이다. 또한, 첫 번째의 계획에 필요한 물의 양은  $9 + 9 + 9 + 6 = 33\ell$ 이다.
- 시각 4의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 9, 9, 9, 9이다. 또한, 첫 번째의 계획에 필요한 물의 양은  $9\ell$ 이다.
- 시각 5의 각 구역의 화재의 세기는 1번 구역부터 차례로 9, 9, 9, 9, 9이다. 또한, 첫 번째의 계획에 필요한 물의 양은  $9 + 9 + 9 = 27\ell$ 이다.

이 입력 예제는 서브태스크 1, 5의 조건을 만족한다.

standard input	standard output
10 10	28
3 1 4 1 5 9 2 6 5 3	21
1 1 6	34
2 8 10	4
4 2 7	64
8 3 3	43
6 1 10	55
3 2 8	9
5 1 9	27
7 4 5	9
9 7 9	
10 10 10	

이 입력 예제는 서브태스크 1, 5의 조건을 만족한다.

standard input	standard output
10 10	9
3 1 4 1 5 9 2 6 5 3	9
1 6 6	3
2 8 8	4
4 2 2	3
8 3 3	4
6 1 1	5
3 4 4	9
5 5 5	9
7 10 10	9
9 8 8	
10 7 7	

이 입력 예제는 서브태스크 1, 3, 5의 조건을 만족한다.

standard input	standard output
10 10	28
3 1 4 1 5 9 2 6 5 3	27
7 1 6	34
7 8 10	4
7 2 7	64
7 3 3	43
7 1 10	55
7 2 8	9
7 1 9	27
7 4 5	9
7 7 9	
7 10 10	

이 입력 예제는 서브태스크 1, 2, 5의 조건을 만족한다.

standard input	standard output
20 20	25
2 1 2 2 1 1 1 1 2 2 2 1 2 1 1 2 1 2 1	30
1	12
1 1 14	32
2 3 18	2
4 10 15	24
8 2 17	38
9 20 20	10
4 8 19	14
7 2 20	40
11 1 5	8
13 2 8	28
20 1 20	24
2 12 15	32
7 1 14	4
12 7 18	2
14 2 17	28
9 19 20	28
12 12 12	12
6 2 15	40
11 2 15	
19 12 17	
4 1 20	

이 입력 예제는 서브태스크 1, 4, 5의 조건을 만족한다.