

# 2019 LGE Code Jam: Online Round 2

## Problem A. 배열 돌리기

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 3 초  
Memory limit: 512 MB

크기가  $n \times n$ 인 2차원 정수 배열  $X$ 가 있다. ( $n$ 은 홀수)

$X$ 를  $45^\circ$ 의 배수만큼 시계방향 혹은 반시계방향으로 돌리려고 한다.  $X$ 를 시계 방향으로  $45^\circ$  돌리면 아래와 같은 연산이 동시에  $X$ 에 적용되어야 한다:

- $X$ 의 주 대각선을  $((1,1), (2,2), \dots, (n, n))$  가운데 열  $((n+1)/2$  번째 열)로 옮긴다.
- $X$ 의 가운데 열을  $X$ 의 부 대각선으로  $((n, 1), (n-1, 2), \dots, (1, n))$  옮긴다.
- $X$ 의 부 대각선을  $X$ 의 가운데 행  $((n+1)/2$  번째 행)으로 옮긴다.
- $X$ 의 가운데 행을  $X$ 의 주 대각선으로 옮긴다. 위 네 가지 경우 모두 원소의 기존 순서는 유지되어야 한다.
- $X$ 의 다른 원소의 위치는 변하지 않는다.

반시계 방향으로  $45^\circ$  돌리는 경우도 위와 비슷하게 정의된다.

예를 들어, 아래 그림 중앙에  $5 \times 5$  배열  $X$ 가 있고, 이 배열을 시계방향 혹은 반시계방향으로  $45^\circ$  돌렸을 때의 결과가 우측 그리고 좌측에 있다. 붉은 원소는 주 대각선 / 중간 열 / 부 대각선 / 중간 행에 위치한 원소이다.

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20
21	22	23	24	25

Table 1: 배열  $X$  ( $5 \times 5$ )

3	2	5	4	15
6	8	9	14	10
1	7	13	19	25
16	12	17	18	20
11	22	21	24	23

Table 2:  $X$ 를 반시계 방향으로  $45^\circ$  회전한 경우

11	2	1	4	3
6	12	7	8	10
21	17	13	9	5
16	18	19	14	20
23	22	25	24	15

Table 3:  $X$ 를 시계 방향으로  $45^\circ$  회전한 경우

입력으로 2차원 배열  $X$ 와 어느 방향으로 몇 도 회전할지 입력 받아, 그 결과를 출력하는 프로그램을 작성하시오.

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스에 대해: 첫 줄에 배열의 크기를 나타내는  $n$  ( $1 \leq n < 500$ ,  $n$ 은 홀수) 그리고 각도  $d$ 가 주어진다.  $d$ 는  $0 \leq |d| \leq 360$  을 만족하며  $|d|$  는 45의 배수이다.  $d$ 가 양수이면 시계방향으로  $d^\circ$  돌려야 하고, 음수이면 반시계방향으로  $|d|^\circ$  돌려야 한다. 다음  $n$ 줄에 걸쳐 각 줄에  $n$ 개의 정수가 공백으로 구분되어 주어진다 ( $X$ 의 원소들을 나타낸다). 각 값은 1 이상 1,000,000 이하의 정수이다.

## Output

각 테스트 케이스에 대해 회전 연산을 마친 후 배열의 상태를 출력한다.  $n$ 줄에 걸쳐 각 줄에  $n$ 개의 정수를 공백으로 구분하여 출력한다.

## Examples

표준 입력	표준 출력
4	11 2 1 4 3
5 45	6 12 7 8 10
1 2 3 4 5	21 17 13 9 5
6 7 8 9 10	16 18 19 14 20
11 12 13 14 15	23 22 25 24 15
16 17 18 19 20	3 2 5 4 15
21 22 23 24 25	6 8 9 14 10
5 -45	1 7 13 19 25
1 2 3 4 5	16 12 17 18 20
6 7 8 9 10	11 22 21 24 23
11 12 13 14 15	23 2 21 4 11
16 17 18 19 20	6 18 17 12 10
21 22 23 24 25	25 19 13 7 1
5 135	16 14 9 8 20
1 2 3 4 5	15 22 5 24 3
6 7 8 9 10	1 2 3 4 5
11 12 13 14 15	6 7 8 9 10
16 17 18 19 20	11 12 13 14 15
21 22 23 24 25	16 17 18 19 20
5 360	21 22 23 24 25
1 2 3 4 5	
6 7 8 9 10	
11 12 13 14 15	
16 17 18 19 20	
21 22 23 24 25	

## Problem B. 홀수 부분열

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 3 초  
Memory limit: 512 MB

길이  $n$ 인 음이 아닌 정수로 이루어진 배열  $A$ 에 대해, 0개 이상의 원소를 지워서 얻을 수 있는 배열을 "부분열"이라고 한다.

배열  $A$ 의 부분열  $S$ 에 대해,  $S$ 가 다음 조건을 만족하면  $S$ 를  $A$ 의 "홀수 부분열"이라고 정의한다:

- $S$ 의 원소의 합의 자릿수 중 홀수인 것들이 홀수이다.

음이 아닌 정수 배열  $A$ 를 입력받아  $A$ 의 서로 다른 홀수 부분열의 개수를 출력하시오. 단, 부분열의 원소들을 정렬했을 시 그 결과가 같다면 같은 부분열로 취급한다.

$A = [3, 3, 6, 8, 6]$  경우 정답은 8이다.

- 길이가 0인 홀수 부분열: X
- 길이가 1인 홀수 부분열: [3]
  - 합이 3, 자릿수 중 홀수가 3으로 하나
- 길이가 2인 홀수 부분열: [3, 6], [6, 6], [6, 8]
  - 합이 각각 9, 12, 14로 자릿수 중 홀수가 9, 1, 1로 하나씩
- 길이가 3인 홀수 부분열: [3, 3, 6], [3, 3, 8]
- 길이가 4인 홀수 부분열: [3, 3, 6, 6], [3, 6, 6, 8]
- 길이가 5인 홀수 부분열: X

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스에 대해 첫 줄에  $A$ 의 길이  $n$ 이 주어진다. 둘째 줄에  $n$ 개의 0 이상의 정수가 공백으로 구분되어 주어진다.  $A$ 의 각 원소는 0 이상 2,000 이하이다.

### Output

각 테스트 케이스에 대해 홀수 부분열의 수를 한 줄에 출력한다.

#### Subtask 1 (3 points)

- $1 \leq n \leq 10$

#### Subtask 2 (11 points)

- $1 \leq n \leq 60$

## Examples

표준 입력	표준 출력
3	3
3	8
3 3 6	8
4	
0 1 2 3	
5	
3 3 6 8 6	

## Notes

- 테스트 케이스 1: 3, 3,6, 3, 3, 6 세 종류의 홀수-부분열이 있다.
- 테스트 케이스 2: 1, 3, 0,1, 0,3, 1,2, 2,3, 0,1,2, 0,2,3 여덟 종류의 홀수-부분열이 있다.
- 테스트 케이스 3: 문제에서 서술된대로 여덟 종류의 홀수-부분열이 있다.

## Problem C. 주식

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 1 초  
Memory limit: 512 MB

당신은 주식을 사고 팔아 이윤을 남기고 싶다. 주식 투자에 대한 법이 최근 바뀌어 아래와 같은 제한이 있다.

- 시장은 1일차부터  $n$ 일차까지,  $n$ 일 동안만 개장된다.
- 당신은 그 중에 연속된  $k$ 일 ( $k \geq 1$ ) 기간 동안에만 참가할 수 있다. 참가하는 날에는 무조건 매일 주식을 사야 한다.
- 당신이 주식을 살 때의 가격은, 매일  $y$ 원으로, 언제나 일정하다.
- 당신이 주식을 팔 때의 가격은,  $i$ 일차에 산 경우  $V[i]$ 원으로, 산 날짜에 따라 다르다.  $V[i]$ 는 음수일 수 있다.
- 당신은 초기 자금  $X$ 원을 가지고 참가한다.
- 당신은 시장 개장 기간인  $n$ 일차가 지나기 전까지는 산 주식을 팔 수 없다.
- 당신은 최소  $Z$ 원 이상의 이윤을 남겨야 한다. ( $Z \geq 0$ )

즉,  $1 \leq i \leq j \leq n$ 을 만족하는  $i$ 일차부터  $j$ 일차까지 주식을 샀다면,

$(V[i] + V[i+1] + \dots + V[j-1] + V[j]) - y \times (j - i + 1) \geq Z$  및  $y \times (j - i + 1) \leq X$  을 만족해야 한다.

이 때, 당신의 목표는  $Z$ 원 이상의 이윤을 남길 수 있는 구매 방법 중에서 가장 짧은 기간을 찾는 것이다.

다시 말해서, 이윤은  $Z$ 원 이상만 남길 수 있다면 아무래도 상관없고, 최대한 짧은 기간 동안에  $Z$ 원 이상의 이윤을 남기는 방법을 찾고 싶다. 그러한 기간을 나타내는 시작일/종료일  $i, j$ 를 출력하는 프로그램을 만드시오.

동일한 최단 기간 구매법이 복수 존재한다면, 그 중  $i$ 가 가장 큰 (즉, 시장에 가장 늦게 진입하는) 기간을 출력하시오.

$Z$ 원 이상의 이윤을 남길 수 있는 방법이 존재하지 않는다면, -1을 출력하시오.

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스는 두 줄에 걸쳐서 주어진다.

첫째 줄에  $n, X, y, Z$ 가 공백으로 구분되어 주어진다.  $0 \leq X, y, Z \leq 10^9$  를 만족한다.

둘째 줄에  $n$ 개의 정수  $V$ 가 공백으로 구분되어 주어진다. 각  $V[i]$  는  $|V[i]| \leq 10^6$  를 만족한다.

### Output

각 테스트 케이스에 대해 한 줄에 답을 출력한다. 문제에서 언급된 조건들을 만족하는  $i, j$ 가 존재하지 않을 경우 -1을 출력하고, 아닌 경우 묘사된대로  $(j - i + 1)$ 을 최소화 하는  $(i, j)$  중  $i$ 값이 최대가 되는  $i, j$  를 공백으로 구분하여 출력한다.

### Subtask 1 (5 points)

- $1 \leq n \leq 5,000$

### Subtask 2 (11 points)

- $1 \leq n \leq 100,000$

### Examples

표준 입력	표준 출력
5	1 1
1 0 0 1	-1
1	1 3
2 0 0 4	3 3
1 2	3 3
3 0 0 3	
2 -1 2	
3 0 0 2	
2 -1 2	
3 0 0 1	
2 -1 2	

## Problem D. 부족 전쟁

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 3 초  
Memory limit: 512 MB

어느 행성에  $N$ 개의 작은 부족들이 살고 있다. 편의상 1부터  $N$ 까지 번호를 부여하자.

- 임의의 서로 다른 두 부족은 서로 동맹 관계이거나 서로 적대 관계이다. 이 관계는 대칭적이다. 즉, 부족 A는 부족 B를 동맹이라 생각하나 B는 A를 적이라 생각하는 경우는 없다.
- 부족 A와 부족 B가 동맹이고, 부족 B와 부족 C가 동맹이라도 부족 A와 부족 C가 적대 관계일 수 있다.
- 임의의 서로 다른 세 부족에 대하여, 세 부족이 서로 동맹 관계인 경우를 “삼자 동맹 관계”라 하고, 세 부족이 서로 적대 관계인 경우를 “삼자 적대 관계”라 하자.

이 때,  $N$ 개의 부족 내의 서로 다른 “삼자 동맹 관계”와 “삼자 적대 관계”들을 전부 합한 개수를 구하라.

예를 들어  $N = 5$ 이고 부족1-부족2, 부족1-부족3, 부족4-부족5가 서로 동맹 관계, 나머지 부족 쌍은 모두 적대 관계라고 가정하자. 이 경우 (2, 3, 4)와 (2, 3, 5)가 “삼자 적대 관계”를 만족하고, “삼자 동맹 관계”는 없기 때문에 답은 2가 된다.

다른 예로  $N = 5$ 이고 부족1-부족2, 부족2-부족3, 부족1-부족3이 동맹인 경우 (1, 2, 3)은 “삼자 동맹 관계”이고, (1, 4, 5), (2, 4, 5), (3, 4, 5)는 “삼자 적대 관계”이기 때문에 답은 4가 된다.

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스 첫 줄에  $N, M$ 이 공백으로 구분 되어 주어진다.

다음  $M$ 줄에 걸쳐서 한 줄에 두 개의 정수  $x, y$ 가 주어지는데 이는  $x$ 와  $y$ 가 동맹임을 나타낸다.

입력으로 주어지는 동맹 관계는 언제나  $1 \leq x < y \leq N$ 을 만족하고, 같은 부족 쌍이 여러 번 주어지는 경우는 없다.

입력으로 주어지지 않은 부족 쌍은 적대관계라고 가정한다.

### Output

각 테스트 케이스에 대해 한 줄에 동맹-반동맹 계수를 출력한다.

#### Subtask 1 (4 points)

- $3 \leq N \leq 200$
- $1 \leq M \leq \min(1,000, N \times (N - 1)/2)$

#### Subtask 2 (13 points)

- $3 \leq N \leq 5,000$
- $1 \leq M \leq \min(1,000,000, N \times (N - 1)/2)$



## Examples

표준 입력	표준 출력
3	0
3 1	4
1 2	2
5 3	
1 2	
2 3	
1 3	
5 3	
1 2	
1 3	
4 5	

## Problem E. 카풀 매칭

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 3 초  
Memory limit: 512 MB

당신은 친구를 도와 새로운 카풀 앱을 개발해야 한다. 그 중 운전자와 승객을 매칭 시켜주는 알고리즘을 당신이 개발해야 한다. 현재 이 앱은 베타 버전이기 때문에 동-서로 길게 늘어진 고속도로 위에 위치한 지역에서만 서비스를 하고 있으며 아래와 같은 제한이 있다.

- 모든 운전자와 승객은 가장 동쪽에 위치한 “출발지” 도시에서 출발한다.
- 총  $N$ 명의 승객이 있으며,  $i$ 번째 승객은 출발지에서 서쪽으로  $x_i$  미터 떨어진 곳에 위치한 목적지에 가고 싶어한다 ( $0 < x_i$ ).
- 총  $M$ 명의 운전자가 있으며,  $j$ 번째 운전자는 출발지에서 서쪽으로  $y_j$  미터 이상  $z_j$  미터 이하 떨어진 곳 사이에 가고 싶어하는 승객을 최대 한 명 태워줄 의향이 있다 ( $0 < y_j \leq z_j$ ).

당신은 이 조건들을 만족하면서 최대한 많은 승객-운전자를 매칭시키는 알고리즘을 작성해야 한다.

예를 들어, 3명의 승객과 3명의 운전자가 있을 때,  $x, y, z$  값이 아래와 같다고 하자.

- $x = [10, 20, 30]$
- $y = [8, 2, 25]$
- $z = [8, 18, 35]$

이 경우 승객 1, 2, 3은 각각 정확 10미터, 20미터, 30미터 떨어진 곳으로 가고 싶어하며, 운전자 1은 정확히 8미터 떨어진 곳에 가려는 승객을 태울 의향이 있고, 운전자 2는 2미터 이상 18미터 이하 떨어진 곳에 가려는 승객을 태울 의향이 있고, 운전자 3은 25미터 이상 35미터 이하 떨어진 곳에 가려는 승객을 태울 의향이 있다.

이 경우, 승객1-운전자2, 승객3-운전자3을 매칭시켜주면 총 두 쌍을 매칭시킬 수 있고 이보다 많은 수의 승객-운전자를 매칭시킬 수 있는 방법은 없다.

입력으로  $N, M$ , 그리고  $x, y, z$  값들을 입력 받아 최대한 많은 승객-운전자 쌍을 매칭시키면 몇 쌍을 매칭시킬 수 있는지 계산하는 프로그램을 작성하시오.

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스의 첫 줄에는  $N, M$  이 주어진다.

그 다음 줄에  $x_i$  가 공백으로 구분되어 주어진다 ( $1 \leq x_i \leq 1,000,000,000$ ).

다음  $M$ 줄에 걸쳐 각 줄에  $y_j$  와  $z_j$  가 공백으로 구분되어 주어진다 ( $1 \leq y_j \leq z_j \leq 1,000,000,000$ ).

### Output

각 테스트 케이스에 대해, 최대한 많은 승객-운전자를 매칭 시켰을 때 몇 쌍을 매칭시킬 수 있는지 출력한다.

### Subtask 1 (7 points)

- $1 \leq N, M \leq 100$

### Subtask 2 (13 points)

- $1 \leq N, M \leq 100,000$

### Examples

표준 입력	표준 출력
3	2
3 3	4
10 20 30	0
8 8	
2 18	
25 35	
4 4	
2 3 4 5	
1 4	
2 4	
3 5	
3 4	
3 3	
1 2 3	
10 20	
30 40	
50 60	

## Problem F. 이진 문자열

Input file: 표준 입력  
Output file: 표준 출력  
Time limit: 1.5 초  
Memory limit: 512 MB

0과 1로만 이루어진 길이  $n$ 의 이진 문자열  $S$ 가 주어져 있다.

$S$ 의 인덱스는 1부터 시작하며,  $S[a : b]$ 는 인덱스  $a$  이상  $b$  이하까지의  $S$ 의 부분 문자열로 정의하자.

이 때, 이 문자열에, 기존 문자열의 부분 문자열을 반전시켜 삽입하는 연산을  $m$ 번 적용한다.

조건들은 다음과 같다.

- 삽입 연산은 2개의 정수 파라미터  $x \leq y$ 를 받는다. 이 때  $m$ 번의 연산 각각에 사용되는  $x, y$  값들은 다를 수 있다.
- 삽입할 문자열은  $S[x : y]$ 의 0/1을 반전시킨 것이다.
- 이 문자열을 삽입할 위치는  $S$ 의  $y$ 번째 문자 바로 뒤이다. 삽입된 이후  $S$ 의 길이는  $y - x + 1$ 만큼 증가한다.
- 삽입 연산시마다  $S$ 는 업데이트된다. 즉, 두 번째 삽입 연산을 시작할 때는 첫 번째 삽입 연산이 끝나고 업데이트된 새로운 문자열  $S$ 를 사용한다.

이 문제의 목적은, 연산  $m$ 번을 모두 적용한 후 마지막 결과  $S$ 에 대해,  $S$ 의 최초  $k$ 개의 문자를 맞추는 것이다.

예를 들어  $S = 01010110$  이고  $n = 8, m = 2, k = 12$  그리고 두 연산 ( $x = 2, y = 4$ )와 ( $x = 3, y = 5$ )를 순서대로 적용한다면 아래와 같은 순서로  $S$ 가 변경된다.

- 첫 번째 연산을 적용하면  $S[2 : 4] = 101$  이니 이를 반전시켜 삽입하여  $S = 01010100110$  을 얻게 된다 (굵은 표시가 된 부분이 삽입된 부분).
- 두 번째 연산을 적용하면  $S[3 : 5] = 010$  이니 이를 반전시켜 삽입하여  $S = 01010101100110$  를 얻게 된다 (굵은 표시가 된 부분이 삽입된 부분).
- $k = 12$  이므로 처음 12글자인 “010101011001” 가 답이 된다.

입력으로  $n, m, k$ , 시작 문자열  $S$ , 그리고  $m$ 번의 연산에 사용되는  $x, y$  값들을 입력 받아 모든 연산을 적용한 후 얻어지는  $S$ 의 처음  $k$ 개의 글자를 출력하시오.

### Input

첫 줄에 테스트 케이스의 수  $T$ 가 주어진다 ( $1 \leq T \leq 10$ ).

각 테스트 케이스의 첫 줄에는  $n, m, k$ 가 공백으로 구분되어 주어진다.

두 번째 줄에는 길이가  $n$ 인 문자열이 주어지며 각 문자는 0 혹은 1이다.

다음  $m$ 줄에 걸쳐 한 줄에 두 개의 정수  $x[i], y[i]$ 가 주어진다. 입력으로 주어지는  $x[i], y[i]$ 는 언제나 다음 조건을 만족한다:  $1 \leq x[i] \leq y[i] \leq i$ 번째 연산을 적용하기 직전의 문자열  $S$ 의 길이.

마찬가지로,  $k$ 는  $m$ 번의 연산을 모두 적용한 후 마지막에 얻은 문자열  $S$ 의 길이와  $10^6$  중 작은 값을 넘지 않는다.

### Output

각 테스트 케이스에 대해 한 줄에 길이가  $k$ 인 문자열을 출력한다.

### Subtask 1 (5 points)

- $1 \leq n \leq 10^3, 1 \leq m \leq 10^3, 1 \leq k \leq 10^5$

### Subtask 2 (23 points)

- $1 \leq n \leq 10^6, 1 \leq m \leq 10^4, 1 \leq k \leq 10^6$

### Examples

표준 입력	표준 출력
3 8 2 12 01010110 2 4 3 5 4 1 4 1010 1 1 8 2 16 10101100 1 2 2 8	010101011001 1001 1001101111001000