

문제 : 동전 줍기

Alice와 Bob은 행의 개수가 2이고 열의 개수가 m 인 2차원 배열에 저장된 동전을 줍는 게임을 하려고 한다. 배열의 $i(1 \leq i \leq 2)$ 번째 행, $j(1 \leq j \leq m)$ 번째 열에 있는 칸에는 $a_{i,j}$ 개의 동전이 들어 있다.

<그림 1>은 $m=4$ 인 경우, 배열의 각 칸의 위치가 어떻게 표시되는지를 보여준다.

1,1	1,2	1,3	1,4
2,1	2,2	2,3	2,4

<그림 1>

두 사람 모두 초기 위치는 배열의 (1,1)이다. 각 사람은 행렬의 칸 사이를 이동하면서 배열의 (2, m) 위치로 가려고 한다. 이동하면서 자기가 지나간 칸에 담긴 동전을 줍는다. 단, 두 사람 모두 이동할 때, 이동 방향은 오른쪽 또는 아래로만 허용된다.

게임이 시작되면 Alice가 먼저 이동을 시작하면서 동전을 줍는다. 즉, 배열의 (1,1) 칸에서부터 시작하여 이동하면서 (2, m) 칸으로 간다.

Alice의 이동이 종료되면, Bob이 배열의 (1,1) 칸에서부터 시작하여 이동하면서 (2, m) 칸으로 이동한다. Bob 역시 자기가 지나간 칸에 담겨 있는 동전을 줍는다. 다만, Alice가 먼저 거쳐 간 칸에서는 주울 동전이 없다.

이 게임의 score는 Bob이 주운 동전의 총 개수이다. Alice는 이 게임의 score를 가능하면 작게 하려고 하고, Bob은 이 게임의 score를 가능하면 크게 하려고 한다.

예를 들어, <그림 2>는 동전이 저장된 배열에서 Alice가 이동한 경로를 보여 주며, <그림 3>은 Bob이 이동한 경로를 보여준다. 이 게임의 score는 9가 되며, 이는 최적이다.

1	3	4	9
2	6	8	5

<그림 2>

			9
2	6		

<그림 3>

두 사람 모두 이 게임의 규칙을 잘 알고 있고, 최적으로 움직일 경우 게임의 score는 얼마가 되는지를 계산하는 프로그램을 작성하고자 한다.

【입 력】

입력파일의 이름은 coin.inp이다. 첫째 줄에는 검사하고자 하는 총 경우의 수 T 가 주어진다. 각 테스트케이스는 세 줄로 구성된다. 첫째 줄에는 문자열 행렬의 칸의 개수를 나타내는 정수 $m(1 \leq m \leq 10^5)$ 이 주어진다. 이어지는 두 줄의 i 번째 줄에는 $a_{i,1}, a_{i,2}, \dots, a_{i,m}$ ($1 \leq a_{i,j} \leq 10^4$)의 값들이 공백으로 구분되어 주어진다.

【출 력】

출력 파일의 이름은 coin.out이다. 각 검사하는 경우에 대해, 두 사람이 모두 최적으로 이동한다면 얻을 수 있는 게임의 score를 한 줄에 출력하라.

【실행 예】

입력 예	입력 예에 대한 출력
3	7
3	8
1 3 7	0
3 5 1	
3	
1 3 9	
3 5 1	
1	
4	
7	

제한조건: 프로그램은 coin.{c,cpp,java}로 한다.