

1. 최적 부분 사각형

$A[0][0] \sim A[N-1][N-1]$ 배열에 1부터 100 사이의 정수들이 채워져 있다. 이 배열로부터 직사각형을 아주 많이 만들어낼 수 있다. 직사각형의 점수는 그 직사각형 안에 있는 짝수를 모두 더하고 홀수를 모두 뺀 값이 된다.

그림1은 $A[0][0] \sim A[4][4]$ 배열에서 가능한 직사각형들의 예를 보인다. 그림의 윗줄 첫 번째 직사각형의 점수는 $84+50+8-37$ 하여 105점이 된다. 아랫줄 첫 번째 직사각형의 점수는 단순히 78점이다.

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

15	3	66	78	39
3	55	1	91	6
65	37	84	52	99
37	50	8	9	22
13	66	23	58	10

그림 1: 가능한 직사각형의 예들

임의의 $A[0][0] \sim A[N-1][N-1]$ 배열이 주어질 때 점수가 가장 큰 직사각형의 점수를 찾는 알고리즘을 작성하라. 단, N 은 200을 넘지 않는다. 수행시간은 10개의 케이스를 모두 수행한 총 시간이 3초를 넘지 않도록 하라. 여러 가지 크기의 문제로 테스트할 것이니 효율이 좋은 알고리즘일수록 점수를 많이 받을 것임. 원시적인 알고리즘으로는 점수를 별로 기대할 수 없음. 컴파일시 옵티마이제이션 옵션은 쓰지 않는다.

[입력]

입력 파일에는 10 개의 테스트 케이스가 주어진다. 각 케이스는 $N+1$ 줄로 이루어진다. 첫 줄에는 N (200 이하)이 주어지고, 이후의 N 줄에는 각각 N 개의 정수(1부터 500 사이)가 주어진다. 첫줄은 $A[0][0...N-1]$, 둘째 줄은 $A[1][0...N-1]$, ..., 마지막 줄은 $A[N-1][0...N-1]$ 의 값을 가지고 있다. 입력파일의 이름은 "input1.txt"이다.

[출력]

각 테스트 케이스에 대해서, 케이스의 번호를 "#x"의 형식으로 출력한 후(여기서 x는 테스트 케이스 번호), 공백을 하나 둔 다음 주어진 케이스에서 받을 수 있는 최대 점수를 기록한다. 출력 결과물을 "output1.txt"로 저장한다.

[예제]

입력 (input1.txt)

2	← 1번 케이스
165 54	
15 353	
3	← 2번 케이스
239 342 62	
267 13 367	
147 349 144	
...	

출력 (output1.txt)

#1 54
#2 404
...