

다음 100X100의 2차원 배열이 주어질 때, 각 행의 합, 각 열의 합, 각 대각선의 합 중 최댓값을 구하는 프로그램을 작성하여라.

단, 배열의 크기는 100X100으로 동일하고 각 행의 합은 integer 범위를 넘어가지 않는다.
동일한 최댓값이 있을 경우, 하나의 값만 출력한다.

예시)

4	4	3	2	1
2	2	1	6	5
3	5	4	6	7
4	2	5	9	7
8	1	9	5	6

1) 열의 합 → 행우선 순회 (반복문)

2) 행의 합 → 열우선 순회 (반복문)

3) 대각 합 → (반복문)

$i=j$ 인 원소들!

시작점에서 i 는 +1 / j 는 -1

(0,4), (1,3), (2,2), (3,1), (4,0)

[SWEA]4835

N개의 정수가 들어있는 배열에서 이웃한 M개의 합을 계산하는 것은 디지털 필터링의 기초연산이다.
M개의 합이 가장 큰 경우와 가장 작은 경우의 차이를 출력하는 프로그램을 작성하시오.

예시) N=5 M=3

index	0	1	2	3	4
list	1	2	3	4	5

N-M

```
for i: 0 → N-M
  for j: 0 → M-1
    합 += list[i+j] → Max-Min
```

[SWEA]5203

0부터 9까지인 숫자 카드 4세트를 섞은 후 6개의 카드를 골랐을 때,
연속인 숫자가 3개 이상이면 run, 같은 숫자가 3개 이상이면 tript이라고 한다.
게임을 시작하면, 플레이어 1과 플레이어 2가 교대로 한 장 씩 카드를 가져가며, 6장을 채우기 전이라도 먼저 run이나 trip이 되는 사람이 승자가 된다.

두 사람이 가져가게 되는 순서대로 12장의 카드에 대한 정보가 주어졌을 때, 승자를 알아내는 프로그램을 작성하시오.
만약 무승부인 경우 0을 출력한다.

예시) 995656114221

A = [9 | 5 | 5 | 1 | 4 | 2] (반복문) 카드가 3개 이상이면 뽑을 때마다 확인하기

B = [9 | 6 | 6 | 1 | 2 | 1] (반복문)

카운팅 정렬 (인덱스 편하게 하기 위해 카드 0 추가)

Count = [| 1 | 1 | | 1 | 2 | | | | 1]
 1 2 3 4 5 6 7 8 9

값이 3이상이면, triplet!

값이 1이상인 인덱스가 연속되면 run!

3개 이상

```
for i: 0 → 9
  if counting[i] ≥ 3 triplet!
for i: 0 → 7
  if counting[i]
    [i+1] ≥ 1 run!
    [i+2]
```


민기는 햄버거 가게에서 자신이 먹었던 햄버거의 재료에 대한 맛을 자신의 오랜 경험을 통해 점수를 매겨놓았다. 민기의 햄버거 재료에 대한 점수와 가게에서 제공하는 재료에 대한 칼로리가 주어졌을 때, 민기가 좋아하는 햄버거를 먹으면서도 다이어트에 성공할 수 있도록 정해진 칼로리 이하의 조합 중에서 민기가 가장 선호하는 햄버거를 조합해주는 프로그램을 만들어보자.

단, 여러 재료를 조합하였을 햄버거의 선호도는 조합된 재료들의 맛에 대한 점수의 합으로 결정되고, 같은 재료를 여러 번 사용할 수 없으며, 햄버거의 조합의 제한은 칼로리를 제외하고는 없다.

예시) 재료의 수(N)=5 / 제한 칼로리=1,000

재료정보 = [(100, 200), (300, 500), (250, 300), (500, 1000), (400, 400)]
(선호도, 칼로리)

- 1. 부분집합 구하기.
조건1) 칼로리 1000 이하
- 2. 부분집합 원소들의 선호도 합 구하기 → MAX

```
1<<N
for i: 0 → 2^N-1
  for j: 0 → N-1
    조건1. i & (1<<j) = 1
    조건2. 재료정보[i][1]의 합이 ≤ 1,000
    ⇒ 재료정보[i][0]의 합의 MAX를 구하기!
```

[SWEA]11453

상자들이 쌓여있는 방이 있다. 방이 오른쪽으로 90도 회전하여 상자들이 중력의 영향을 받아 낙하한다고 할 때, 낙차가 가장 큰 상자를 구하여 그 낙차를 리턴하는 프로그램을 작성하시오.

중력은 회전이 완료된 후 적용되며 상자들은 모두 한 쪽 벽면에 붙여진 상태로 쌓여 2차원의 형태를 이루며 벽에서 떨어져서 쌓인 상자는 없다.

방의 가로 길이는 항상 100이며, 세로 길이도 항상 100이다.
즉, 상자는 최소 0, 최대 100 높이로 쌓을 수 있다.

예시) 높이를 값으로 갖는 배열 생성

arr = [10, 9, 8, 9, 10, 6, 9, 5, 2, 10]

회전



∴ 낙차 = 나보다 작은 수의 개수 (반복문) → 조건 i) 내 뒤에 위치한다

```
i, j = index
for i: 0 → 98 (마지막 상자 제외)
  for j: i+1 → 99
    조건: arr[i] > arr[j]
    count = count + 1 → 최댓값
```