

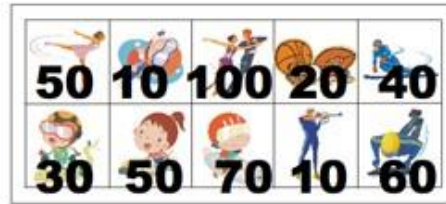
스티커

<https://www.acmicpc.net/problem/9465>

- 스티커가 $2n$ 개가 $2 \times n$ 모양으로 배치되어 있다.
- 스티커가 한 장을 떼면 변을 공유하는 스티커는 모두 찢어져서 사용할 수 없다.
- 점수의 합을 최대로 만드는 문제



(a)



(b)

99	100	99
1	99	1

1	100	1000
100	1	1

- $1 + 100 + 1 = 102$ 를 선택하는 것 보다 $99 + 99 + 99$ 를 선택하는 것이 더 합이 크므로 최댓값만을 선택하는 것이 최선은 아니다.
- 항상 큰 값만을 따라가는 것도 최선은 아니다. 위의 오른쪽 표를 보면 큰 값만 따라간 경우 $100 + 100 + 1$ 이 되지만, $100 + 1000$ 이 가장 큰 값이 된다.

스티커

<https://www.acmicpc.net/problem/9465>

$dp[i][j] = 2 * i$ 스티커에서 스티커 마지막 열 상태가 j 일 때 점수의 최대합
 j 의 상태는 다음과 같다.

X	O	X
X	X	O
0	1	2

0 : 마지막 열 스티커를 모두 사용할 수 없는 경우

1 : 마지막 열 스티커 중 위의 스티커는 사용 가능하고 아래 스티커는 사용 불가

2 : 마지막 열 스티커 중 위의 스티커는 사용 불가능하고 아래 스티커는 사용 가능

$$dp[i][0] = \max(dp[i-1][0], dp[i-1][1], dp[i-1][2])$$

$$dp[i][1] = \max(dp[i-1][0], dp[i-1][2]) + A[0][i]$$

$$dp[i][2] = \max(dp[i-1][0], dp[i-1][1]) + A[1][i]$$

→ $\max(dp[N][0], dp[N][1], dp[N][2])$ 가 정답이 된다.

스티커

<https://www.acmicpc.net/problem/9465>

```
for (int i = 1; i <= N; ++i) {  
    dp[i][0] = max({ dp[i - 1][0], dp[i - 1][1], dp[i - 1][2] });  
    dp[i][1] = max(dp[i - 1][0], dp[i - 1][2]) + A[0][i];  
    dp[i][2] = max(dp[i - 1][0], dp[i - 1][1]) + A[1][i];  
}  
  
printf("%d\n", max({ dp[N][0], dp[N][1], dp[N][2] }));
```