

# 브루트 포스 알고리즘

= 무식한 알고리즘

= 무식한 알고리즘  
or 전체탐색, 완전탐색

장점

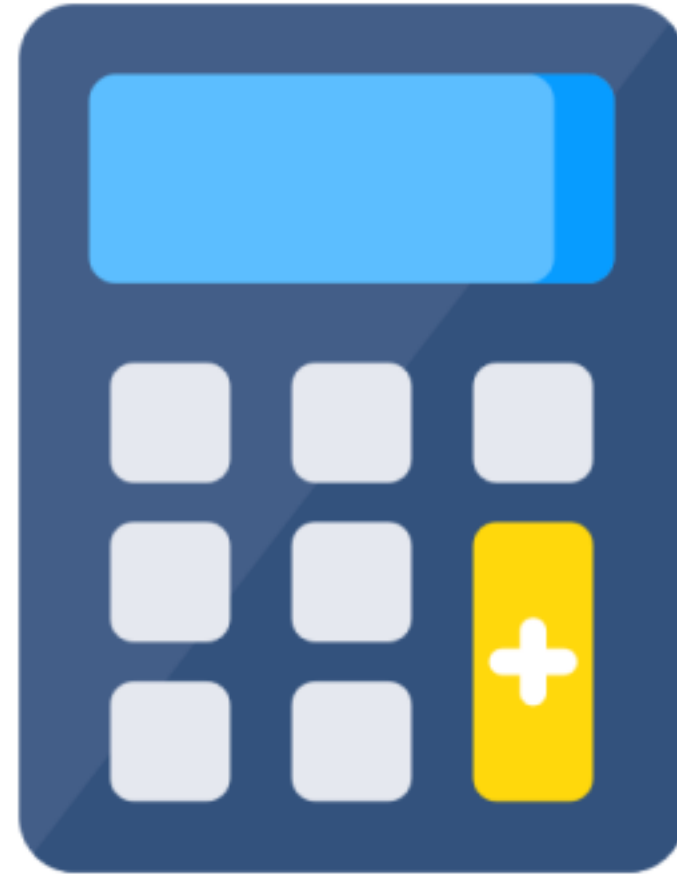
확실한 정답

빠른 구현

단점

효율

느린 속도



$$1 + 2 + 3 + 4 + 5 + 6 + 7 + 8 + 9 + 10$$

$$S_n = \frac{n\{2a + (n - 1)d\}}{2}$$

등차수열의 합 공식

# 예제

백준 2798 - 블랙잭 (<https://www.acmicpc.net/problem/2798>)



시간 제한	메모리 제한	제출	정답	맞힌 사람	정답 비율
1 초	128 MB	181103	90616	69405	48.741%

## 문제

카지노에서 제일 인기 있는 게임 블랙잭의 규칙은 상당히 쉽다. 카드의 합이 21을 넘지 않는 한도 내에서, 카드의 합을 최대한 크게 만드는 게임이다. 블랙잭은 카지노마다 다양한 규정이 있다.

한국 최고의 블랙잭 고수 김정인은 새로운 블랙잭 규칙을 만들어 상근, 창영이와 게임하려고 한다.

김정인 버전의 블랙잭에서 각 카드에는 양의 정수가 쓰여 있다. 그 다음, 딜러는 N장의 카드를 모두 숫자가 보이도록 바닥에 놓는다. 그런 후에 딜러는 숫자 M을 크게 외친다.

이제 플레이어는 제한된 시간 안에 N장의 카드 중에서 3장의 카드를 골라야 한다. 블랙잭 변형 게임이기 때문에, 플레이어가 고른 카드의 합은 M을 넘지 않으면서 M과 최대한 가깝게 만들어야 한다.

N장의 카드에 써져 있는 숫자가 주어졌을 때, M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 구해 출력하시오.

## 입력

첫째 줄에 카드의 개수  $N$  ( $3 \leq N \leq 100$ )과  $M$  ( $10 \leq M \leq 300,000$ )이 주어진다. 둘째 줄에는 카드에 쓰여 있는 수가 주어지며, 이 값은 100,000을 넘지 않는 양의 정수이다.

합이 M을 넘지 않는 카드 3장을 찾을 수 있는 경우만 입력으로 주어진다.

## 출력

첫째 줄에 M을 넘지 않으면서 M에 최대한 가까운 카드 3장의 합을 출력한다.

### 예제 입력 1 [복사](#)

```
5 21
5 6 7 8 9
```

### 예제 출력 1 [복사](#)

```
21
```

```
lst = [0, 1, 2, 3, 4]
```

```
N = len(lst)
```

```
for i in range(N-2):
```

```
    for j in range(i+1, N-1):
```

```
        for k in range(j+1, N):
```

```
            print(lst[i], lst[j], lst[k])
```

출력: 0 1 2

0 1 3

0 1 4

0 2 3

0 2 4

0 3 4

1 2 3

1 2 4

1 3 4

2 3 4

i j k  
i = 0일 때 j = 1, k = 2 [0, 1, 2, 3, 4]

i j k  
[0, 1, 2, 3, 4]

i j k  
[0, 1, 2, 3, 4]

i j k  
i = 0일 때 j = 2, k = 3 [0, 1, 2, 3, 4]

i j k  
[0, 1, 2, 3, 4]

i j k  
i = 0일 때 j = 3, k = 4 [0, 1, 2, 3, 4]

i j k  
i = 1일 때 j = 2, k = 3 [0, 1, 2, 3, 4]

.

.

.

# Python 풀이

```
# input
# 5 21
# 5 6 7 8 9

# N = 5, M = 21
N, M = map(int, input().split())
# lst = [5, 6, 7, 8, 9]
lst = list(map(int, input().split()))

# 조합 합 저장해줄 빈 배열 선언
combinations = []
```

```
# 모든 조합 탐색
for i in range(N-2):
    for j in range(i+1, N-1):
        for k in range(j+1, N):
            temp = lst[i] + lst[j] + lst[k]
            # 세 카드의 합이 M 이하이면
            if temp <= M:
                # 리스트에 추가
                combinations.append(temp)

# 리스트 중 가장 큰 수 출력
print(max(combinations))
```

# Java 풀이 | <https://st-lab.tistory.com/97>

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in = new Scanner(System.in);

        int N = in.nextInt();
        int M = in.nextInt();

        int[] arr = new int[N];

        for (int i = 0; i < N; i++) {
            arr[i] = in.nextInt();
        }

        int result = search(arr, N, M);
        System.out.println(result);
    }
}

// 탐색
static int search(int[] arr, int N, int M) {
    int result = 0;

    // 3개를 고르기 때문에 첫번째 카드는 N - 2 까지만 순회
    for (int i = 0; i < N - 2; i++) {
        // 두 번째 카드는 첫 번째 카드 다음부터 N - 1 까지만 순회
        for (int j = i + 1; j < N - 1; j++) {
            // 세 번째 카드는 두 번째 카드 다음부터 N 까지 순회
            for (int k = j + 1; k < N; k++) {
                // 3개 카드의 합 변수 temp
                int temp = arr[i] + arr[j] + arr[k];

                // M과 세 카드의 합이 같다면 temp return 및 종료
                if (M == temp) {
                    return temp;
                }

                // 세 카드의 합이 이전 합보다 크면서 M 보다 작을 경우 result 갱신
                if (result < temp && temp < M) {
                    result = temp;
                }
            }
        }
    }
}
```

감사합니다