

오늘은 백준 브론즈 2 문제

```
// C99
#include <stdio.h>

int main(void)
{
    int N; // 카드의 개수 ( $3 \leq N \leq 100$ )
    int M; // 한도 ( $10 \leq M \leq 300,000$ )
    int card[100]; // 카드에 쓰여 있는 수 (100,000을 넘지 않는 양의 정수)
    int sum = 0; // M을 넘지 않으면서 M에 가장 가까운 카드 3장의 합
    int temp; // 카드 3장의 합을 임시로 저장할 변수

    scanf("%d%d", &N, &M);
    for (int i = 0; i < N; i++)
        scanf("%d", &card[i]);

    for (int i = 0; i < N - 2; i++)
    {
        for (int j = i + 1; j < N - 1; j++)
        {
            for (int k = j + 1; k < N; k++)
            {
                temp = card[i] + card[j] + card[k];
                if (sum < temp && temp <= M)
                    sum = temp;
            }
        }
    }
    printf("%d", sum);

    return 0;
}
```

BAE<KJOON>
ONLINE JUDGE

2798번: 블랙잭

2798번 제출 맞은 사람 슷코딩 재채점 결과 채점 현황 강의 블랙잭 출처 ...

www.acmicpc.net

최소 3, 최대 100장의 카드 세 장의 합 중
M을 넘지 않으면서 M에 가장 가까운 것을 구해 출력해야 합니다
카드 100장 중 3장을 선택하는 경우의 수는

$${}_{100}C_3 = 161700$$

굉장히 적은 수입니다
시간 제한도 1초로 넉넉하므로(대략 1초에 1억 번 정도 간단한 연산, 탐색이 가능)
이럴 때는 브루트 포스 알고리즘으로 문제를 푸는 것이 적합합니다

브루트 포스(Brute force) 알고리즘은 영어 말뜻 그대로입니다
단순무식하게 일일이 모든 경우를 탐색하여 답을 찾아내는 것입니다
비효율적으로 보이지만 가장 확실하고 정확한 문제 풀이 방법입니다

오늘도 마찬가지로
브루트 포스 알고리즘에 대한
보다 자세한 사항은 [나무위키](#)를 참고하시길 바랍니다

모든 경우를 탐색하되 똑같은 카드 세 장의 합을 구하는 일이 없도록
반복문만 잘 작성해주면 됩니다
k 반복문에 i, j, k 값을 출력해주는 코드를 넣어주면 코드 작성, 디버깅에 많은 도움이 됩니다