

오늘은 백준 골드 5 문제

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

int max(int n1, int n2)
{
    return n2 < n1 ? n1 : n2;
}

int main(void)
{
    static int DP[101][100001]; // DP[i][j]: 0 ~ i번째 물건을 최대 j만큼의 무게만을 넣을 수
    int N; // 물품의 수 (1 ≤ N ≤ 100)
    int K; // 준서가 버틸 수 있는 무게 (1 ≤ K ≤ 100,000)
    int W[101]; // W[i]: i번째 물건의 무게 (1 ≤ W[i] ≤ 100,000)
    int V[101]; // V[i]: i번째 물건의 가치 (0 ≤ V[i] ≤ 1,000)

    scanf("%d%d", &N, &K);
    for (int i = 1; i <= N; i++)
        scanf("%d%d", &W[i], &V[i]);

    for (int i = 0; i <= N; i++)
    {
        for (int j = 0; j <= K; j++)
        {
            if (i == 0 || j == 0) // 0번째 물건은 없음 / 물건의 무게는 최소 1이므로 최대 0만
                DP[i][j] = 0;
            else if (j < W[i]) // 배낭의 무게 한도보다 i번째 물건이 더 무거운 경우 해당 물건
                DP[i][j] = DP[i - 1][j];
            else // W[i] <= j
                DP[i][j] = max(DP[i - 1][j], DP[i - 1][j - W[i]] + V[i]);
        }
    }
    printf("%d", DP[N][K]);

    return 0;
}
```

# BAE<K>JOON>

## ONLINE JUDGE

### 12865번: 평범한 배낭

12865번 제출 맞은 사람 숏코딩 재채점 결과 채점 현황 강의 평범한 배낭...

[www.acmicpc.net](http://www.acmicpc.net)

코드는 간결하게 나오지만 이해하기 꽤나 어려운 문제였습니다  
동적 계획법(다이나믹 프로그래밍)을 이용하여 풀 수 있는 문제입니다  
배낭 문제의 유형은 크게 분할 가능한 배낭 문제, 0-1 배낭 문제로 나뉘는데  
이 문제는 0-1 배낭 문제에 해당합니다  
물품마다 '담는다'와 '담지 않는다'라는 두 가지 경우만 있어서 붙은 이름이 아닐까 싶습니다  
보다 자세한 설명은 [나무위키](#)에서 확인하시면 됩니다

$\max()$ 는 매개변수  $n1, n2$  중 더 큰 값을 리턴하는 함수입니다  
인수(argument)와 매개변수(parameter)는 헷갈리기 쉽지만 이해하고 나면 어렵지 않습니다  
함수로 전달하는 값이 인수, 함수에서 인수를 담는 변수가 매개변수입니다  
인수  $DP[i - 1][j]$ ,  $DP[i - 1][j - W[i]] + V[i]$ 이 각각  $n1, n2$ 에 담기는 것입니다

선언부에서 DP 배열만 정적으로 선언한데는 이유가 있습니다  
지역 변수로 DP 배열을 선언하고 실행하면  
Segmentation fault (core dumped), 메모리 초과 에러가 발생합니다

이 에러는 함수 밖에 변수를 선언하거나(전역 변수)  
정적 변수를 사용해주면 해결됩니다  
저는 정적 변수를 사용했습니다

선언부에는 주석을 자세히 달아놨으니 이 쯤에서 선언부 설명은 마무리하고  
지금부터는 구현부 설명을 해보겠습니다  
반복문 내의 조건문을 그림과 함께 살펴봅시다

$\langle i = 0 \rangle$  0번째 물건

→ 없는 물건이므로 가치는 0으로 함



물건의 최소 무게: 1, 배낭에 물건을 담을 수 있는 최대 무게: 0

→ 배낭에는 아무것도 담을 수 없으므로

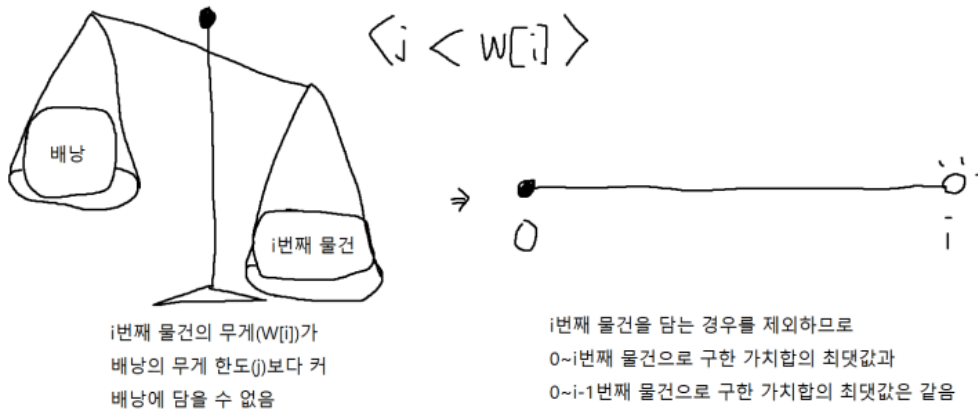
배낭에 넣을 수 있는 물건들의 가치의 최댓값은 0

if

if 조건문을 설명하는 그림입니다

i와 j가 0일 때 0을 대입하는 코드를 작성했지만

이제와서 보니 애초에 DP 배열을 0으로 초기화해놓고 i와 j가 1일 때부터 반복문을 시작하는 것이 더 좋을 듯 합니다



else if

else if 조건문을 설명하는 그림입니다



else

else 조건문을 설명하는 그림입니다  
그림을 설명하는 데 제대로 활용하지 못한 것 같아 아쉽습니다  
제 부족한 설명은 여기까지입니다

배열의 내용물이 궁금하시다면  
j 반복문 마지막에 공백 문자를 출력하고  
i 반복문 마지막에 개행 문자를 출력하는 코드를 추가해보시기 바랍니다