

과제 #5 계단 오르기 - 201921438 조현태

1) 소스코드

```
#include <stdio.h>
#include <iostream>
using namespace std;
// n의 최대범위
#define MAX 100000

int n;
// 2차원 배열
int dp[MAX][4] = {0, };

int main()
{
    cin >> n;

    // 초기값 설정
    dp[0][1] = 1;

    // 점화식을 이용한 DP풀이
    for (int i = 1; i <= n; i++)
        for (int j = 1; j <= 3; j++)
            // 배열 인덱스 벗어나지않도록 조정.
            if ((i - j) >= 0)
            {
                if (j == 3)
                    dp[i][j] = (dp[i - j][1] + dp[i - j][2]) % 1004;
                else
                    dp[i][j] = (dp[i - j][1] + dp[i - j][2] + dp[i - j][3])
% 1004;
            }

    // 정답 출력
    cout << (dp[n][1] + dp[n][2] + dp[n][3]) % 1004 << endl;

    return 0;
}
```

2) 문제 설명

이 문제는 계단을 1칸, 2칸, 3칸 중에 선택해서 올라갈 수 있고,
3칸을 연속으로 올라가는 것이 불가능할 때의 모든 경우의 수를 구하는 것입니다.

예전에 계단 오르기 문제를 풀었던 기억이 있어서 점화식을 세워서 풀었습니다.
점화식을 세우기 위해서 n 을 점점 늘려가면서 규칙을 찾아보았습니다.

$n = 1$ 일 때, (1)로 1가지 방법이 있습니다.

$n = 2$ 일 때, (1,1) (2)로 2가지 방법이 있습니다.

$n = 3$ 일 때, (1,1,1) (1,2) (2,1) (3)로 4가지 방법이 있습니다.

$n = 4$ 일 때

1층에서 3칸을 올라가는 경우 : (1,3)

2층에서 2칸을 올라가는 경우 : (1,1,2) (2,2)

3층에서 1칸을 올라가는 경우 : (1,1,1,1) (1,2,1) (2,1,1) (3,1)

$\therefore 1 + 2 + 4 = 7$

위의 경우로 $an = an-1 + an-2 + an-3$ 이라는 점화식을 구할 수 있습니다.

이때, 3칸을 연속으로 올라가지 못한다는 조건이 있으므로
모든 경우의 수에서 3칸씩을 올라가는 경우를 빼려고 했으나
점화식이 2개가 나와서 직관적으로 생각하기가 어려워서
3칸을 올라가는 경우에는 1칸, 2칸을 올라가는 경우만 더하도록 했습니다.

이를 위해서 2차원 배열로 만들어서 이해하기 쉽게 만들었습니다.

```
int dp[100000][4] = {0, };
```

```
for (int i = 1; i <= n; i++)  
    for (int j = 1; j <= 3; j++)  
        if (j == 3)  
            (dp[i][j] = dp[i - j][1] + dp[i - j][2]) % 1004;  
        else  
            (dp[i][j] = dp[i - j][1] + dp[i - j][2] + dp[i - j][3]) % 1004;
```

이를 1차원 배열로 나타내면 아래와 같습니다.

	1	2	3
1	1	0	0
2	1	1	0
3	2	1	1
4	4	2	1
5	7	4	2
6	13	7	3
7	23	13	6

이때 출력값은 모든 경우의 수이므로 n에 해당하는 모든 값을 더해주면 정답입니다.

```
cout << (dp[n][1] + dp[n][2] + dp[n][3]) % 1004 << endl;
```