

콜라츠 추측, 오버플로우

분명 맞게 코드 짰는데, 답이 자꾸 틀림.

이유 → **오버플로우** 때문이었음

입출력 예#3에 626331을 입력한다면 엉뚱하게 488이 나옴

이것은 오버플로우로 인한 문제로 곱하기 3과 +1을 하는 과정에서 int 범위를 넘어서기에 발생

그래서 long long형으로 풀어야함

오버플로우

: 변수의 자료형이 표현할 수 있는 최대값을 넘어서거나 최소값보다 작아지는 상황

→ 이 경우, 계산 결과가 예상하지 못한 값이 될 수 있으며, 프로그램의 버그나 오류를 유발

	int	long long
크기	4바이트(32비트)	8바이트(64비트)
범위	-2,147,483,648 (-2^{31}) ~ 2,147,483,647 ($2^{31}-1$)	-9,223,372,036,854,775,808 (-2^{63}) ~ 9,223,372,036,854,775,807 ($2^{63}-1$)

문제 설명

1937년 Collatz란 사람에 의해 제기된 이 추측은, 주어진 수가 1이 될 때까지

1-1. 입력된 수가 짝수라면 2로 나눕니다.

1-2. 입력된 수가 홀수라면 3을 곱하고 1을 더합니다.

2. 결과로 나온 수에 같은 작업을 1이 될 때까지 반복합니다.

예를 들어, 주어진 수가 6이라면 $6 \rightarrow 3 \rightarrow 10 \rightarrow 5 \rightarrow 16 \rightarrow 8 \rightarrow 4 \rightarrow 2 \rightarrow 1$

제한 사항

입력된 수, num은 1 이상 8,000,000 미만인 정수입니다.

입출력 예

n	result
6	8
16	4
626331	-1

```
#include <string>
#include <vector>

using namespace std;

int solution(int num) {
    int answer = 0;
    //자료형 long long 으로 해주지 않으면 오버플로우 발
    long long n=num;
    while(true){
        if(answer>500){
            answer=-1;
            break;
        }
        if(n==1){
            return answer;
        }

        if(n%2==0){
            n=n/2;
        }
        else{
            n=n*3+1;
        }
        answer++;
    }
    return answer;
}
```