



이진변환 반복하기, 변환법

이진수

: 0과 1만으로 숫자를 표현하는 방법

십진수	이진수
1	1
2	10
3	11
4	100
5	101
6	110
7	111
8	1000
9	1001
10	1010

1. n bit로 표현할 수 있는 숫자의 개수는 2^n 개다.

ex) bit 3개로 표현 가능한 수는 $2^3 = 8$ 개다.

2. n bit로 표현 가능한 가장 큰 양수는 $2^n - 1$ 이다.

ex) bit 3개로 표현 가능한 가장 높은 양수는 $2^3 - 1 = 7$ 이다.

3. 특정 자리의 bit이 의미하는 숫자는 $2^{(n-1)}$ 이다.

ex) 3번째 자리의 bit이 1일 경우 (100) 해당 자리의 값은 $2^{(3-1)} = 4$ 이다.

1	1	1	1	1	1	1
64	32	16	8	4	2	1

이진수→십진수

이진수로 11001이 주어졌다고 하자.

이때, 1인 자리는 첫 번째 자리, 네 번째 자리, 다섯 번째 자리이다.

이전에 언급한 팩트 3번에 의하면 n번째 자리의 bit 값은 $2^{(n-1)}$ 이다.

첫 번째 자리 → $2^{(1-1)} = 1$

네 번째 자리 → $2^{(4-1)} = 8$

다섯 번째 자리 → $2^{(5-1)} = 16$

이 값들을 모두 합치면 25, 즉 11001은 25다.

1	1	0	0	1
16	8	0	0	1

십진수→이진수

1. 십진수를 2로 나눠서 나머지를 적어둔다.

2. 2로 나눈 몫을 또 2로 나워서 나머지를 적어둔다.

3. 몫이 0이 될 때까지 위의 단계를 반복한다.

4. 지금까지 기록한 나머지를 최신순으로 이진수를 왼쪽에서 오른쪽으로 적는다.

십진수 29이 주어져다고 하자.

1. 29을 2로 나누면 나머지는 1, 몫은 14이다.

$$\begin{array}{r|l} 1 & 29 \\ & 14 \end{array}$$

2. 14을 2로 다시 나누면 나머지는 0, 몫은 7이다.

$$\begin{array}{r|l} 1 & 29 \\ 0 & 14 \\ & 7 \end{array}$$

3. 다시 7을 2로 나누면 나머지는 1, 몫은 3이다.

$$\begin{array}{r|l} 1 & 29 \\ 0 & 14 \\ 1 & 7 \\ & 3 \end{array}$$

4. 3을 2로 나누면 나머지는 1, 몫은 1이다.

$$\begin{array}{r|l} 1 & 29 \\ 0 & 14 \\ 1 & 7 \\ 1 & 3 \\ & 1 \end{array}$$

5. 마지막으로 1을 2로 나누면 나머지는 1, 몫은 0이다.

1	29
0	14
1	7
1	3
1	1
	0

6. 지금까지 기록한 나머지를 최신순부터 이진수로 왼쪽에서 오른쪽으로 적는다.

그럼 11101이 나온다.

즉, 십진수 29는 이진수로 11101이다.

문제 설명

0과 1로 이루어진 어떤 문자열 x 에 대한 이진 변환을 다음과 같이 정의합니다.

x 의 모든 0을 제거합니다.

x 의 길이를 c 라고 하면, x 를 " c 를 2진법으로 표현한 문자열"로 바꿉니다.

예를 들어, $x = "01111010"$ 이라면, x 에 이진 변환을 가하면 $x = "01111010" \rightarrow "1111" \rightarrow "100"$ 이 됩니다.

0과 1로 이루어진 문자열 s 가 매개변수로 주어집니다. s 가 "1"이 될 때까지 계속해서 s 에 이진 변환을 가했을

제한사항

s 의 길이는 1 이상 150,000 이하입니다.

s 에는 '1'이 최소 하나 이상 포함되어 있습니다.

입출력 예

```
s    result
"110010101001"  [3,8]
"011110"  [3,3]
"1111111"   [4,1]
```

```
#include <string>
#include <vector>
#include <algorithm>
using namespace std;

vector<int> solution(string s) {
    vector<int> answer;

    string tmp="";
    int BinaryCount=0;
    int ZeroCount=0;

    while(s!="1"){ // 문자열은 " "
```

```

    for(int i=0; i<s.length();i++){
        if(s[i]!='0'){ // 문자는 ' '
            tmp+=s[i];
        }
        else{
            ZeroCount++;
        }
    }

    int length = tmp.length();
    s.clear();
    tmp.clear(); //!!!!클리어 안해줘서 무한루트 돌았던 것 같음!!!

    while(length>0){ //!!! 0이 아닐 때로 조건을 걸었는데 그럼 0.x인 경우 문제 발생!!!
        s+=to_string(length%2);
        length=length/2;
    }
    BinaryCount++;
    reverse(s.begin(),s.end());
}
answer.push_back(BinaryCount);
answer.push_back(ZeroCount);

return answer;
}

```