

## 数列描述

### 题目描述：

有一个数列  $a[N]$  ( $N=60$ )，从  $a[0]$  开始，每一项都是一个数字。数列中  $a[n+1]$  都是  $a[n]$  的描述。其中  $a[0]=1$ 。

规则如下：

$a[0]:1$

$a[1]:11$ (含义：其前一项  $a[0]=1$  是 1 个 1，即“11”。表示  $a[0]$  从左到右，连续出现了 1 次“1”)

$a[2]:21$ (含义：其前一项  $a[1]=11$ ，从左到右：是由两个 1 组成，即“21”。表示  $a[1]$  从左到右，连续出现了两次“1”)

$a[3]:1211$ (含义：其前一项  $a[2]=21$ ，从左到右：是由一个 2 和一个 1 组成，即“1211”。表示  $a[2]$  从左到右，连续出现了 1 次“2”，然后又连续出现了 1 次“1”)

$a[4]:111221$ (含义：其前一项  $a[3]=1211$ ，从左到右：是由一个 1、一个 2、两个 1 组成，即“111221”。表示  $a[3]$  从左到右，连续出现了 1 次“1”，连续出现了 1 次“2”，连续出现了两次“1”)

请输出这个数列的第  $n$  项结果 ( $a[n]$ ,  $0 \leq n \leq 59$ )。

### 输入描述：

数列的第  $n$  项( $0 \leq n \leq 59$ ):

4

### 输出描述：

数列的内容：

111221

### 补充说明：

### 示例

#### 示例 1

输入：

4

输出：

111221

说明：

```
#include <iostream>
```

```
#include <string>
```

```
#include <vector>
```

```

#include <map>
#include <algorithm>

using namespace std;
string resOfNums(int n){
    if (n==0){
        return "1";
    }
    string str = resOfNums(n-1);
    char ch = str[0];
    vector<pair<char,int>> vp;

    int times=1;
    for (int i = 1; i < str.size(); ++i){
        if (str[i]==ch){
            ++times;
        }else{
            vp.push_back(make_pair(ch,times));
            ch = str[i];
            times = 1;
        }
    }
    vp.push_back(make_pair(ch,times));
    str.clear();
    for (int i = 0; i < vp.size(); ++i){
        str += to_string((vp[i].second))+vp[i].first;
    }
    return str;
}

int main()
{
    int n;
    cin >> n;
    cout << resOfNums(n) << endl;
    return 0;
}

```