# UVA948

# 題目

The well known Fibonacci sequence is obtained by starting with 0 and 1 and then adding the two last numbers to get the next one. For example the third number in the sequence is 1 (1=1+0), the forth is 2 (2=1+1), the fifth is 3 (3=2+1) and so on.

| i | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| F ib(i) | 0 | 1 | 1 | 2 | 3 | 5 | 8 | 13 | 21 | 34 |

Figure 1 - The first numbers in the Fibonacci sequence

有名的費氏數列是以 0 和 1 開始，然後把最後的兩個數字相加以得到下一項。例如數列的第三項為 1 (1=1+0)，第四項為 2 (2=1+1)，第五項為 3 (3=2+1)，等等。

# 題目

The sequence appears on many things in our life, in nature, and has a great significance. Among other things, do you know that all positive integer numbers can be represented as a sum of numbers in the Fibonacci sequence? More than that, all positive integers can be represented as a sum of a set of Fibonacci numbers, that is, numbers from the sequence, without repetition. For example: 13 can be the sum of the sets {13}, {5,8} or {2,3,8} and 17 is represented by {1,3,13} or {1,3,5,8}.

這個數列很重要且出現在我們生活及大自然的許多事物上。其中，你知道所有的正整數都可以用費氏數列中的不重覆的數字集合的和來代表嗎？例如：13 可以是集合 {13}, {5,8} 或{2,3,8} 的和，而 17 則可用 {1,3,13} 或 {1,3,5,8}來表示。

# 題目

Since all numbers have this property (do you want to try to prove this for yourself?) this set could be a nice way to use as a “base” to represent the number. But, as we have seen, some numbers have more than one set whose sum is the number. How can we solve that? Simple! If we add the constraint that the sets cannot have two consecutive Fibonacci numbers, than we have a unique representation for each number! This restriction is because the sum of any two consecutive Fibonacci numbers is just the following Fibonacci number.

即然每個數都有這個特性（你要自己證證看嗎？）這個數列可以用作表示數字的「底」。但如前所示，有些數字有多種表示法，這該怎麼辦呢？簡單，因為費氏數列中任兩個連續項的和就是下一項，所以只要加上一個限制：不得使用連續的項，那麼每個數字都有一個唯一的表示法。

# 題目

Now that we know all this we can prepare a nice way to represent any positive integer. We will use a binary sequence (just zeros and ones) to do that. For example, 17 = 1 + 3 + 13 (remember that no two consecutive Fibonacci numbers can be used). Let’s write a zero for each Fibonacci number that is not used and one for each one that is used, starting at the right.

有了這個認知後我們可以建立一個很酷的方式來表示任意正整數。我們一連串的 0 與 1 來表示。例如：17 = 1 + 3 + 13 (記住不可以使用費氏數列中連續的項)，以 0 來表示沒有用到的項，以 1 來表示有用到的項，由右至左排列。

# 題目

Then, 17 = 100101. See figure 2 for a detailed explanation. In this representation we should not have zeros at the left, this is, we should only write starting with the first one. In order for you to understand better, note that in this scheme, not using two consecutive Fibonacci numbers means that the binary sequence will not have two consecutive ones. When we use this representation for a number we say that we are using the Fibonaccimal base, and we write it like 17 = 100101 (fib).

| | | | | | | |
|---|---|---|---|---|---|---|
| 17= | 1 | 0 | 0 | 1 | 0 | 1 |
| 13+3+1= | 13 | 8 | 5 | 3 | 2 | 1 |

Given a set of numbers in decimal base, your task is to write them in the Fibonaccimal base.

因此 17 = 100101. 詳情參閱圖 2。在這個表示法中，最左邊的那一位數一定是 1，不可以是 0。根據我們的限制，這種表示法中不可以出現相鄰的 1。我們這種表示法為「費氏進位」並以下列方式表示 17 = 100101 (fib).
給你一組十進位的數字，你的任務是將它們以費氏進位輸出。

# 輸入與輸出

Input：The first line of input contains a single number N, representing the quantity of numbers that follow (1 ⩽ N ⩽ 500). Than follow exactly N lines, each one containing a single positive integer smaller than 100,000,000. These numbers can come in any order

Output：You should output a single line for each of the N integers in the input, with the format ‘DEC_BASE = FIB_BASE (fib)’. DEC_BASE is the original number in decimal base and FIB_BASE is its representation in Fibonaccimal base. See the sample output for an example.

輸入：輸入的第一行含有一個數字 N，代表以下有幾個數字 ( $1 \leq N \leq 500$)。

接下來有 N 行，每行有一個小於 100 000 000 的正整數。數字不一定按順序出現。

輸出：對於 N 個整數中的每一個整數要用下列格式輸出一行，"DEC_BASE = FIB_BASE (fib)"。DEC_BASE 原始的十進位數字而 FIB_BASE 則是它的費氏進位表示法。詳情參閱範例輸出。

# 範例測資

Input：
10
1
2
3
4
5
6
7
8
9
10

Output：
1 = 1 (fib)
2 = 10 (fib)
3 = 100 (fib)
4 = 101 (fib)
5 = 1000 (fib)
6 = 1001 (fib)
7 = 1010 (fib)
8 = 10000 (fib)
9 = 10001 (fib)
10 = 10010 (fib)

# 程式碼說明

## Step 1：建表

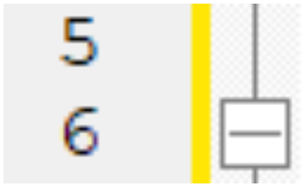
```
4   int f[43]={1,2,3,5,8,13,
5              21,34,55,89,
6              144,233,377,
7              610,987,1597,
8              2584,4181,6765,
9              10946,17711,28657,
10             46368,75025,121393,
11             196418,317811,514229,
12             832040,1346269,2178309,
13             3524578,5702887,9227465,
14             14930352,24157817,39088169,
15             63245986,102334155,165580141,
16             267914296,433494437,701408733};
```

| 已宣告變數 | 註解 |
| --- | --- |
| f | 費氏數列 |

截圖方便，寫成一行即可

# 程式碼說明

## Step 2：輸入測資




```
int n;cin>>n;
while(cin>>n){
```

| 已宣告變數 | 註解 |
| --- | --- |
| f | 費氏數列 |
| n | 十進位數字 |

# 程式碼說明

## Step 3：尋找比n小的最大費氏數列

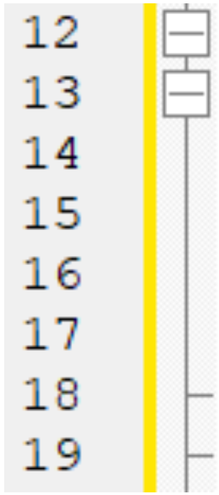
```
 9
10
11
```

```
int p=0;
for(p;f[p]<=n;p++);
p--;
```

| 已宣告變數 | 註解 |
| --- | --- |
| f | 費氏數列 |
| n | 十進位數字 |
| p | 比n小最大費氏數列編號 |

# 程式碼說明

## Step 4：判斷和輸出1或0



```
12
13
14
15
16
17
18
19
```

```
for(p;p>=0;p--){
    if(n>=f[p]){
        cout<<1;
        n-=f[p];
    }else{
        cout<<0;
    }
}
```

| 已宣告變數 | 註解 |
| --- | --- |
| f | 費氏數列 |
| n | 十進位數字 |
| p | 比n小最大費氏數列編號 |

# 補充

```cpp
#include<iostream>
using namespace std;
int main(){
    int a[43]={1,2},n=3;
    for(int i=2;i<43;i++,n++){
        a[i]=a[i-1]+a[i-2];
        if(a[i]>=1000000000){
            break;
        }
    }
    for(int i:a){
        cout << i << ",";
    }
}
```

# 完整程式碼

```cpp
#include<iostream>
using namespace std;
int main(){
    int f[43]={1,2,3,5,8,13,21
    int n;cin>>n;
    while(cin>>n){

        cout<<n<<" = ";
        int p=0;
        for(p;f[p]<=n;p++);
        p--;
        for(p;p>=0;p--){
            if(n>=f[p]){
                cout<<1;
                n-=f[p];
            }else{
                cout<<0;
            }
        }
        cout<<" (fib)"<<endl;
    }
}
```

完整費氏數列請回到第9頁

# 資料來源

英文題目：

https://vjudge.net/problem/UVA-948

中文翻譯：

https://zerojudge.tw/ShowProblem?problemid=a134

# Thanks!