★★☆☆☆

題組：Problem Set Archive with Online Judge

題號：10015 : Joseph's Cousin

解題者：呂易澄

學號：ACS105111

使用語言:C++

解題日期：2018 年 10 月 28 日

22207600	10015 Joseph's Cousin	Accepted	C++	0.170	2018-10-28 15:20:51

題目:

The Josephs problem is notoriously known. For those who are not familiar with the problem, among n people numbered 1, 2, . . . , n, standing in circle every mth is going to be executed and only the life of the last remaining person will be saved. Joseph was smart enough to choose the position of the last remaining person, thus saving his life to give the message about the incident.

Although many good programmers have been saved since Joseph spread out this information, Josephs Cousin introduced a new variant of the malignant game. This insane character is known for its barbarian ideas and wishes to clean up the world from silly programmers. We had to infiltrate some the agents of the ACM in order to know the process in this new mortal game.

In order to save yourself from this evil practice, you must develop a tool capable of predicting which person will be saved.

The Destructive Process

The persons are eliminated in a very peculiar order; m is a dynamical variable, which each time takes a different value corresponding to the prime numbers succession (2,3,5,7,...). So in order to kill the i-th person, Josephs cousin counts up to the i-th prime.

Input

It consists of separate lines containing n [1..3501], and finishes with a ‘0’.

Output

The output will consist in separate lines containing the position of the person which life will be saved.

Sample Input

6

0

Sample Output

4

問題描述：

某部落酋長有 n 個俘虜（編號從 1, 2, 3, ……, n），他叫他們排成一個圈圈，然後開始數，第 m 個人要被煮來吃掉（第一次從編號 1 的人開始數），按照此規則繼續下去，直到只剩下一個人，那一個人可以保留性命。例如：n=6, m=5 則被吃掉的人的編號依序是 5, 4, 6, 2, 3 最號只有編號 1 活了下來。Joseph 是個很聰明的人，他總是能挑到最後存留的位置，所以這件事才被披露出來。在這之後，救了不少被抓的程式設計師。

Joseph's Cousin 很討厭程式設計師，所以他寫了一封信介紹了一種新的算法給酋長，以增加計算的難度。在他的算法中 m 不是固定的，而是根據質數的順序（2, 3, 5, 7, 11, 13, ……）產生的。所以當酋長要吃第 i 個人時，他就算第 i 個質數。

你的任務就是要找出最後一個存活者的位置。

Input

每行一個整數 n（0<n<3502）, n=0 代表輸入結束。

Output

根據輸入的 n，輸出最後存活者的位置。

Sample Input

6

0

Sample Output

4

解法：

將前 3501 個質數建立成質數表，以迴圈持續減少人數直到減少至最後一人

解法範例：

1. 由 2 開始將檢查後得出的質數一個個堆入 vector 之中，堆至第 3501 個數後停止
2. 依輸入人數建立一串依序編號的 vector
3. 以迴圈開始尋找目標，找到後將其 erase
4. 當 vector 的 size 剩下 1 時將其編號印出

討論：

- 在程式開始時便建立質數表而非每個測資重新建立
- 利用已建立的質數表來加速後續質數的尋找
- 利用取餘數而非迴圈的方式尋找下一個目標
- 利用 erase 功能以便於扣除已死亡者

程式：

```
//UVA10015 Joseph's Cousin

#include<iostream>
#include<vector>

using namespace std;

bool isPrime(int, vector<int>);

int main(){

	vector<int> people;
	vector<int> primes;
	int input;

	//建立質數表
	for(int i=2;primes.size()<3503;i++){
		if(isPrime(i,primes))primes.push_back(i);
	}

	for(;;){
		//輸入
		cin>>input;
		if(input==0)break;

		//建立生存者列表
		people.clear();
		for(int i=0;i<input;i++){
			people.push_back(i+1);
		}

		//處理目標
		int count=0;
		for(int i=0;i<input-1;i++){
			count--;
			count=(count+primes[i])%people.size();
			people.erase(people.begin()+count);
```

```
                }

                //輸出
                cout<<people[0]<<endl;
        }
}

//檢查是否為質數
bool isPrime(int num, vector<int> primes){
        for(int i=0;i<primes.size() && primes[i]*primes[i]<=num;i++){
                if(num%primes[i]==0)return false;
        }

        return true;
}
```