**★★☆☆☆**

**題組：基礎48題**

**題號：Q151: Power Crisis**

**整理者：呂易澄**

**學號：ACS105111**

**使用語言:C++**

**解題日期：2018年7月22日**



**題目:**

During the power crisis in New Zealand this winter (caused by a shortage of rain and hence low levels in the hydro dams), a contingency scheme was developed to turn off the power to areas of the country in a systematic, totally fair, manner. The country was divided up into N regions (Auckland was region number 1, and Wellington number 13). A number, m, would be picked ‘at random’, and the power would first be turned off in region 1 (clearly the fairest starting point) and then in every m’th region after that, wrapping around to 1 after N, and ignoring regions already turned off. For example, if N = 17 and m = 5, power would be turned off to the regions in the order:1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7.

The problem is that it is clearly fairest to turn off Wellington last (after all, that is where the Electricity headquarters are), so for a given N, the ‘random’ number m needs to be carefully chosen so that region 13 is the last region selected.

Write a program that will read in the number of regions and then determine the smallest number m that will ensure that Wellington (region 13) can function while the rest of the country is blacked out.

**Input**

Input will consist of a series of lines, each line containing the number of regions (N) with 13 ≤ N < 100.

The file will be terminated by a line consisting of a single ‘0’.

**Output**

Output will consist of a series of lines, one for each line of the input. Each line will consist of the number m according to the above scheme.

**Sample Input**

17  
0

**Sample Output**

7

**問題描述：**能源危機 在今年冬天發生於紐西蘭的能源危機事 件中（由於缺水導致水壩因低水位無法發電所引起），他們發展出了一套緊急配套計畫來有系統且完全公正地決定國內哪個地區要被斷電。這個國家分成N個區域（澳克蘭是1號區域，威靈頓是第13號區域）。首先他們會「隨機」選擇一個數m，接著從第1號區域（這是最公平的起始地點了）開 始斷電，接著每往後數m個區域（忽略掉已經斷電過的區域，超 過N就從頭繼續數），被點到的最後一個區 域就會成為下一個被斷電的區域。例如說，如果 N = 17 並且 m = 5，那麼依序被斷電的區域是：1,6,11,16,5,12,2,9,17,10,4,15,14,3,8,13,7. 問題是，最公平的情況下，威靈頓應該 要是最後一個被斷電的區域（畢竟那是電力供應廠的所在地），所以每一個「隨機」選擇的數字m應當被小心地選擇好讓區域13是最後一個被選上的斷電區域。 編寫一個程式，可以讀入區域的數量並 且決定最小的數字m使得威靈頓（區域13）在剩下的區域都黯淡無光以前仍然在運作。

**Input**

輸入含有多組測試資料。  
每組測試資料一列，包含一個數字N，其中 13 <= N < 100  
當 n=0 代表輸入結束，請參考Sample Input。

**Output**

對每組測試資料輸出一列 。輸出滿足上述需求的最小m。

**Sample Input**

17  
0

**Sample Output**

7

**解法：**

以迴圈尋找數字m

**解法範例：**

1. 以bool確認該地區是否已停電
2. 將m假設為1
3. 從區域1開始，利用副程式對已停電的地區進行忽略並尋找下i個停電區
4. 以迴圈持續將各區域停電，若遇到威靈頓(區域13)則跳出
5. 確認是否所有區域皆已停電，不是則將m++後返回步驟3，是則作為結果輸出

**討論：**

* Josephus Problem
* 需跳過已停電的區域
* 注意陣列起始為0

**程式：**

//Josephus Problem

#include <iostream>

using namespace std;

//找出下一個地區

int next(bool area[],int i,int j,int total\_area){

int k;

for(k=0;k<i;){

j=(j+1)%total\_area;

//如果已停電則跳過

if(area[j]==false)k++;

}

return j;

}

int main(){

int total\_area,i,j,count;

bool area[101];

for(;;){

cin>>total\_area;

if(total\_area == 0)break;

for(i=1;i<=total\_area;i++){

//初始化

for(j=0;j<total\_area;j++){

area[j]=false;

}

count=0;

//逐一停電

for(j=0;j<total\_area;j=next(area,i,j,total\_area)){

if(j==12 || count>=total\_area)break;

area[j]=true;

count++;

}

if(j==12 && count==total\_area-1){

//經過所有區域(扣掉威靈頓)則表示為正確答案

cout<<i<<endl;

break;

}

}

}

}