**★★☆☆☆**

**題組：Problem Set Archive** **with Online Judge**

**題號：11264: Coin Collector**

**解題者：徐筱媛**

**解題日期：2021年2月28日**

**題意：**

給定Case數(<=1000)、可用的面額數、面額(<1000000000)，依據以下的換錢模式：優先最大面額，若剩餘金額不足最大面額，則給予次大的面額，以此類推；求各個Case中，在任意金額之下最多能夠換到幾種面額？

**題意範例：**

[Sample Input]

2

6

1 2 4 8 16 32

6

1 3 6 8 15 20

[Sample Output]

6

4

**解法：**

1. 讀取測資數量並記錄面額(coin)。
2. 處理測資：

　for i=0 to n-1:

　　 if(sum(coin[0:i]’)<coin[i+1]):

 　 //換完coin[i+1]的面額後，coin[0:i]中至少有一個面額無法拿到，

 　 //不使用coin[i]可確保僅coin[i]這一面額無法拿到。

 　 不使用coin[i]且從sum中抹去coin[i]

　　 else:

 　 使用coin[i]

1. 輸出使用面額的數量。

**解法範例：**

無。

**討論：**

使用暴力法將會TLE，而由於題目中已經規定可用面額輸入的規定：非降序且無重複值，為上述解法提供了很好的預處理，因此選用此法。

**程式：**

#include <iostream>

using namespace std;

int main(){

 int c;

 cin>>c;

 while(c--){

 //input

 int catalog;

 cin>>catalog;

 int coin[catalog]={0};

 for(int i=0;i<catalog;i++) cin>>coin[i];

 //handle

 int sum=coin[0],ans=2;

 for(int i=1;i<catalog-1;i++){

 sum+=coin[i];

 //if (sum(coin[0:i+1]>coin[i+1]),

 //that means if we implement with coin[i], at least one of coin[0:i+1] will be excluded,

 //and thus just excluded coin[i]

 if(sum<coin[i+1]) ans++;

 else sum-=coin[i];

 }

 //output

 cout<<ans<<endl;

 }

 return 0;

}