

Problem A

Maximum Consecutive Sum

Time limit: 2 sec.

Problem Description

There is an integer array. We want to find a consecutive subsequence in it, which have the maximal sum in all subsequences. For example, an array like this : -1, 5, 7, -2, 8, -2, 1, -5, -8, 4, 6, -2, the consecutive subsequence sum have $(-1)+5=4$, $(-1)+5+7+(-2)+8=17$, $5+7=12$,...etc. And the Maximum consecutive sum in this array is $5+7+(-2)+8=18$.

Input Format

The first line contains an integer n which indicates the number of test cases. Each of the following n lines is a test case. Each test case starts with an integer m followed by m integers which are the elements of the array, separated by a space. Each integer is between -1000 and 1000. The number of elements in each case is at most 5000.

Output Format

Each of the n lines contains an integer which is the answer of the given. Output 0 when all elements are negative.

Example

| Sample Input | Sample Output |
|-------------------------------|---------------|
| 2 | 0 |
| 8 -2 -3 -1 -1 -3 -2 -3 -3 | 8 |
| 10 -8 6 -5 -2 8 -7 3 -3 -8 -8 | |

Hint

本題要在一個整數陣列中找總和最大的連續一段。此題有多種解法，其中一種利用習題中的 prefix-sum query。在計算完 prefix sum 後，從位置 i 到 j 的總和可以用一個減法得到，我們可以窮舉所有的 $i \leq j$ ，找最大的就是答案。

Problem B

Binary search for a root

Time limit: 2 sec

Problem Description

對一個連續函數 $f(x)$ ，若 $f(r)=0$ ，則 r 是 f 的根。對於 $a < b$ ，若 $f(a)*f(b) < 0$ ，則在 (a,b) 區間內必有一根，此為勘根定理。我們可以用二分法來找到一個近似根。本題假設 f 是一個多項式次數不超過 P 。在程式實作上，限撰寫一個函數來計算 f 的值

Double $f(\text{int } p, \text{double } x, \text{double } c[])$; 其中 c 是傳入函數的係數。

給定 $a < b$ ，且 $f(a)*f(b) < 0$ ，我們求其平均值 $x=(a+b)/2$ ，計算 $f(x)$ 的值，如果 $f(a)*f(x) < 0$ 則把區間縮小為 (a,x) ；如果 $f(x)*f(b) < 0$ 則把區間縮小為 (x,b) 。如此繼續下去直到達到要求的精準位數。在浮點數的計算中，我們不可以 $= 0$ 來判別，因此本題假設在絕對數值 < 0.000001 時，當作 0。

Input Format

第一行是測資筆數 T 。每筆測資三行：第一行是一個整數 P 代表多項式的次數， $P \leq 6$ 。第二行有 $P+1$ 個浮點數分別是由高而低的係數，第三行兩個浮點數 a 與 b 。

Output Format

每筆測資輸出介於 a 與 b 之間的根，四捨五入到小數點以下第 5 位。

Example

| Sample Input | Sample Output |
|--------------|---------------|
| 2 | 1.00000 |
| 1 | 1.41421 |
| 1.0 -1.0 | |
| -1 2 | |
| 2 | |
| 1 0 -2 | |
| 0 2 | |

Problem C

Special Fibonacci % P

Time limit: 2 sec

Problem Description

費氏數列是數學上相當著名的數列，其特徵是每一項數字皆為前兩項之和，現在我們定義一個類似費氏數列的數列，如下所示

- $F_0 = 0$
- $F_1 = 1$
- $F_n = F_{n-1} + 2 * F_{n-2} \quad (n \geq 2)$

本題要找出此一數列第 N 項除以特定正整數 P 的餘數

Input Format

有多筆測資，每筆測資會有兩個正整數 N 和 P，兩數之間以空白隔開， $n \leq 1000$ ， $p \leq 1000$
當 $N=P=0$ 時結束程式，不需處理這一筆測資。

Output Format

每筆測資以一行輸出費氏數列第 N 項除以 P 的餘數

Example

| Sample Input | Sample Output |
|--------------|---------------|
| 7 6 | 1 |
| 10 13 | 3 |
| 0 0 | |

Problem D

Reverse and Add

Time limit: 2 sec

Problem Description

The "reverse and add" method is simple: choose a number, reverse its digits and add it to the original. If the sum is not a palindrome (which means, it is not the same number from left to right and right to left), repeat this procedure.

For example:

195 Initial number

591

786

687

1473

3741

5214

4125

9339 Resulting palindrome

In this particular case the palindrome 9339 appeared after the 4th addition. This method leads to palindromes in a few steps for almost all of the integers. But there are interesting exceptions.

196 is the first number for which no palindrome has been found. It is not proven though, that there is no such a palindrome.

Task :

You must write a program that give the resulting palindrome and the number of iterations (additions) to compute the palindrome.

You might assume that all tests data on this problem:

- will have an answer,
- will be computable with less than 1000 iterations (additions),
- will yield a palindrome that is not greater than 4,294,967,295.

Input Format

The first line will have a number N (the number of test cases), and each of the next N lines will have a number P to compute its palindrome.

Output Format

For each of the N tests you will have to write a line with the following data: minimum number of iterations (additions) to get to the palindrome and the resulting palindrome itself separated by one space.

Example

| Sample Input | Sample Output |
|---------------------|----------------------|
| 3 | 4 9339 |
| 195 | 5 45254 |
| 265 | 0 121 |
| 121 | |

Problem E

Prime Words

Time limit: 1 sec.

Problem Description

A prime number is a number that has only two divisors: itself and the number one. Examples of prime numbers are: 1, 2, 3, 5, 17, 101 and 10007. In this problem you should read a set of words, each word is composed only by letters in the range a-z and A-Z. Each letter has a specific value, the letter a is worth 1, letter b is worth 2 and so on until letter z that is worth 26. In the same way, letter A is worth 27, letter B is worth 28 and letter Z is worth 52. You should write a program to determine if a word is a prime word or not. A word is a prime word if the **sum of its letters** is a prime number.

Input Format

The input consists of a set of words. Each word is in a line by itself and has L letters, where $1 \leq L \leq 20$. The input is terminated by end of file (EOF).

Output Format

For each word you should print: "It is a prime word.", if the sum of the letters of the word is a prime number, otherwise you should print: "It is not a prime word."

Example

| Sample Input | Sample Output |
|--------------|-------------------------|
| UFRN | It is a prime word. |
| contest | It is not a prime word. |
| AcM | It is not a prime word. |