ĐỀ THI 5

**Bài 1 Số Sphenic**

Số tự nhiên N được gọi là số Sphenic nếu N là tích của chính xác ba số nguyên tố khác nhau. Một vài số Sphenic đầu tiên là 30,42, 66, 70, 78,102,105,110,114...

Cho một số N nhiệm vụ của bạn là kiểm tra xem đó có phải là số Sphenic hay không.

Ví dụ: N = 30 = 2 \* 3 \* 5 là số Sphenic, còn 60 = 2­­2 \*3 \* 5 thì không phải vì thừa số 2 có số mũ lớn hơn 1.

**Đầu vào:** Đọc từ file **SphenicNum**.INP gồm:

Dòng đầu tiên chứa số nguyên T là số bộ dữ liệu vào kiểm tra, mỗi bộ dữ liệu vào gồm một dòng chứa số nguyên dương N.

**Ràng buộc**

0 < T <=100; 0 < N <=10­­6

**Đầu ra:** Ghi vào file **SphenicNum**.OUT gồm:

In ra T dòng, nếu số nguyên N từ đầu vào là số Sphenic thì in ra số 1, ngược lại thì in ra số 0.

**Ví dụ:**

|  |  |
| --- | --- |
| **Đầu vào** | **Đầu ra** |
| 2  30  60 | 1  0 |

**Bài 2 Những ngôi nhà có ánh nắng**

Cho dãy N ngôi nhà sắp thành một hàng thẳng được đánh số từ 1 đến N từ trái sang phải. Mặt trời mới mọc và chiếu sáng từ bên trái dãy nhà. Với mỗi ngôi nhà thứ i có chiều cao H[i] thì những ngôi nhà bên phải nó mà có chiều cao nhỏ hơn H[i] sẽ không nhận được ánh nắng. Hãy cho biết có bao nhiêu ngôi nhà nhận được ánh nắng.

**Đầu vào:** Đọc từ file **Buildings**.INP gồm:

Dòng đầu tiên của đầu vào chứa một số nguyên T biểu thị số bộ dữ liệu cần kiểm tra. Trong đó, mỗi bộ dữ liệu gồm:

* Dòng đầu chứa số nguyên N
* Dòng thứ hai chứa N số nguyên cách nhau một dấu cách biểu thị độ cao các ngôi nhà.

**Đầu ra:** Ghi vào file **Buildings**.OUT gồm:

Ứng với mỗi bộ dữ liệu đầu vào, in ra một dòng chứa kết quả tương ứng với bộ dữ liệu.

**Ràng buộc**

* 1 <= T <= 100; 1 <= N <= 10­­5 ; 1 <= H[i] <= 100, i = 1..N

**Ví dụ**

|  |  |
| --- | --- |
| **Đầu vào** | **Đầu ra** |
| 3  6  6 2 8 4 11 13  5  2 5 1 8 3  7  3 4 1 0 6 2 3 | 4  3  3 |

**Bài 3 Các từ quá dài**

Đôi khi một số từ như “ localization” hoặc “internationalization” quá dài đến nỗi việc viết chúng nhiều lần trong một văn bản khá mệt mỏi.

Hãy coi một từ là quá dài nếu độ dài của nó dài hơn 10 kí tự. Tất cả các từ quá dài nên được thay thế bằng một từ viết tắt đặc biệt.

Chữ viết tắt này được thực hiện theo quy tắc sau: ta viết ra chữ cái đầu tiên và chữ cái cuối cùng của một từ và ở giữa ta viết một con số là số lượng chữ cái đứng giữa chữ cái đầu tiên và chữ cái cuối cùng. Con số đó nằm trong hệ thập phân và không chứa chữ số 0 ở đầu.

Do đó, từ “ localization” sẽ được viết là “l10n” và “internationalization” sẽ được viết là “i18n”.

Bạn hãy lập trình để tự động hóa quá trình chuyển đổi các từ quá dài thành các từ tắt. Các từ không quá dài sẽ được viết như cũ.

**Đầu vào:** Đọc từ file **LongWord**.INP gồm:

Dòng đầu tiên chứa một số nguyên n (1 <=n <=100). Mỗi dòng trong n dòng sau chứa một từ. Tất cả các từ là các chữ cái Latinh in thường khác rỗng và có độ dài không quá 100 ký tự.

**Đầu ra:** Ghi vào file **LongWord**.OUT gồm:

In ra n dòng, dòng thứ i phải chứa kết quả thay thế từ thứ i từ dữ liệu vào.

**Ví dụ**

|  |  |
| --- | --- |
| **Đầu vào** | **Đầu ra** |
| 4  word  localization  internationalization  pcongcongcongcongcongcongcongcongcongcongcons | word  l10n  i18n  p43s |

**Bài 4: Dãy con tăng liền kề dài nhất và có tổng nguyên tố lớn nhất**

Cho dãy A gồm N số nguyên. Hãy tìm dãy con tăng dài nhất gồm các phần tử liền kề, nếu có nhiều dãy con tăng dài nhất như thế thì hãy chỉ ra dãy có tổng các phần tử là số nguyên tố lớn nhất.

Ví dụ: A[] = {4, 2, 3, 5, 1, 6, 7, 8, 9 }. Trong ví dụ này ta có dãy con { 1, 6, 7, 8, 9 } là dài nhất gồm 5 phần tử và có tổng các số nguyên tố = 7 cũng là lớn nhất.

**Đầu vào:** Đọc từ file **LISMaxSumPrime**.INP gồm:

Dòng đầu tiên chứa số nguyên T là số bộ dữ liệu, mỗ bộ dữ liệu gồm:

* Dòng đầu chứa số nguyên N
* Dòng sau chứa N số nguyên, các số cách nhau một dấu cách là các phần tử của dãy A.

**Ràng buộc**

* 1 <= T <=100; 1<= N <=10­­5
* 1 <= A[i] <= 10­­6 , i = 1..N

**Đầu ra:** Ghi vào file **LISMaxSumPrime**.OUT gồm:

Ứng với mỗi bộ dữ liệu test in ra một dòng, chứa độ dài dãy con tăng liền kề dài nhất và tổng các số nguyên tố trong dãy đó.

**Ví dụ**

|  |  |
| --- | --- |
| **Đầu vào** | **Đầu ra** |
| 3  9  4 2 3 5 1 6 7 8 9  10  2 3 5 7 4 1 6 5 4 8  5  2 2 2 2 2 | 5 7  4 17  1 2 |

**Bài 5. Tổng số k**

Cho một dãy gồm N số nguyên a1, a2, …, aN và một số nguyên dương K. Ta có biểu thức: ? a1 ? a2 … ? aN = X

Trong đó ta có thể thay thế dấu ‘?’ bằng cách phép toán ‘+’ hoặc ‘-‘. Hãy đếm xem có tất cả bao nhiêu cách thay dấu khác nhau để X chia hết cho K. Hai cách thay dấu gọi là khác nhau nếu tồn tại một vị trí mà ở đó dấu ở cách này khác dấu ở cách kia.

***Dữ liệu:*** Vào từ file **CAU5.INP** gồm:

+ Dòng đầu tiên gồm 2 số nguyên dương N, K.

+ Dòng thứ hai gồm N số nguyên dương a1, a2, …, aN. (ai ≤ 109)

***Kết quả:*** Ghi ra file **CAU5.OUT** là số cách thay dấu lấy modulo cho 109 + 7.

***Ví dụ:***

|  |  |
| --- | --- |
| **CAU5.INP** | **CAU5.OUT** |
| 5 3  1 7 14 6 7 | 12 |

***Giới hạn:***

+ Có 40% số điểm: N ≤ 20, K ≤ 10.

+ Có 60% số điểm còn lại: N ≤ 10000, K ≤ 1000.