

Data Structure

Assignment 2

1. Description :

A digital computer in AD 4th century used linked lists as its internal representation of positive integers (in the binary form). Read two positive integers, *ione* and *itwo* (could be very long integers), and carry out one of the following arithmetic operations to get the third (*ithree*). The input numbers are given in the decimal form.

S x	Print the number of nodes in the linked list representation of x (1, 2 or 3)
A	Add <i>ione</i> and <i>itwo</i> and store result in <i>ithree</i> .
M	Multiply <i>ione</i> and <i>itwo</i> and store result in <i>ithree</i> .
P	Print <i>ithree</i> in decimal form.

Input Format:

A single line consisting of two integers *ione* and *itwo*. And then N followed by N instructions one in each line.

Output Format:

As described above.

Sample Input :

```
9347538975384535 328749237423423947
6
S 1
S 2
M
S 3
A
P
2
```

Sample Output:

```
54
59
112
338096776398808482
```

2. Description:

Read two strings from the keyboard character by character. Each string is terminated by a new line. Represent both of them as linked lists in the memory with start pointing to the first characters. Now remove from the first string all characters which are present in the second string.

Input Format:

First string on line1 and second on line2

Output Format:

Single line

Sample Input :

RAILROAD CROSSING WITHOUT ANY VOWELS
AEIOUY

Sample Output:

RLRD CRSSNG WTHT N VWLS

3. Description

N people are waiting in line to enter a concert. People get bored waiting so they turn and look for someone familiar in the line. Two persons A and B standing in line can see each other if they're standing right next to each other or if no person between them is strictly taller than person A or person B. Write a program that determines the number of pairs of people that can see each other.

Input Format:

The first line of input contains an integer N ($1 \leq N \leq 500000$), the number of people standing in line. Each of the following N lines contains a single integer, the height of one person in nanometres. Everyone will be shorter than 231 nanometres. The heights are given in the order in which people are standing in line.

Output Format:

Output the number of pairs of people that can see each other on a single line.
Constraints $1 \leq N \leq 500000$, Height $< 2^{31}$

Sample Input :

7
2
4
1
2
2
5
1
4.5

Sample Output:

10

4. Description:

A histogram is a polygon composed of a sequence of rectangles aligned at a common base line. The rectangles have equal widths but may have different heights. Calculate the area of the largest rectangle in a histogram that is aligned at the common base line, too.

Input Format:

The input contains several test cases. Each test case describes a histogram and starts with an integer n, denoting the number of rectangles it is composed of. You may assume that $1 \leq n \leq 100000$. Then follow n integers h_1, \dots, h_n , where $0 \leq h_i \leq 1000000000$. These numbers denote the heights of the rectangles of the histogram in left-to-right order. The width of each

rectangle is 1. A zero follows the input for the last test case.

Output Format:

For each test case output on a single line the area of the largest rectangle in the specified histogram. Remember that this rectangle must be aligned at the common base line.

Sample Input :

```
7 2 1 4 5 1 3 3
4 1000 1000 1000 1000
0
```

Sample Output:

```
8
4000
```

5. Description

You are given an expression consisting of $()[]\{\}$, check if the expression is a proper bracketed sequence.

Input Format:

First line contains a integer N, followed by N test cases

Sample Format:

Print Yes or No for each test cases in a line.

Sample Input:

```
2
(([]))
([)]
```

Sample Output:

```
Yes
No
```

6. Description

Given a proper mathematical expression, output a fully bracketed expression.

Input Format:

First line contain integer N, followed by N test cases

Output Format:

Print the fully bracketed expression

Sample Input:

```
2
a+b-c
a+b*c-d
```

Sample Output:

```
((a+b)-c)
((a+(b*c))-d)
```

7. Description

You are given a string S and a pattern P. Find the number of times an anagram of P occurs as a substring of S.

Input Format:

First line contain integer N, followed by N test case. Each test contain a string S followed by a pattern P.

Output Format:

For each test case, Output the number of times an anagram of P occurs as a substring of S.