1. Given two strings **s** and **t.**Return the minimum number of operations required to convert **s**to **t**.  
The possible operations are permitted:

1. Insert a character at any position of the string.
2. Remove any character from the string.
3. Replace any character from the string with any other character.

**Input:**

s = "geek", t = "gesek"

**Output:** 1

**Explanation:** One operation is required

inserting 's' between two 'e's of s.

**Input :**

s = "gfg", t = "gfg"

**Output:**

0

**Explanation:** Both strings are same.

2. The stock span problem is a financial problem where we have a series of **n** daily price quotes for a stock and we need to calculate the span of stocks price for all **n** days.   
The span **Si** of the stocks price on a given day **i** is defined as the maximum number of consecutive days just before the given day, for which the price of the stock on the current day is less than or equal to its price on the given day.  
For example, if an array of 7 days prices is given as {100, 80, 60, 70, 60, 75, 85}, then the span values for corresponding 7 days are {1, 1, 1, 2, 1, 4, 6}.

**Input**:

N = 7, price[] = [100 80 60 70 60 75 85]

**Output**:

1 1 1 2 1 4 6

**Explanation**:

Traversing the given input span for 100

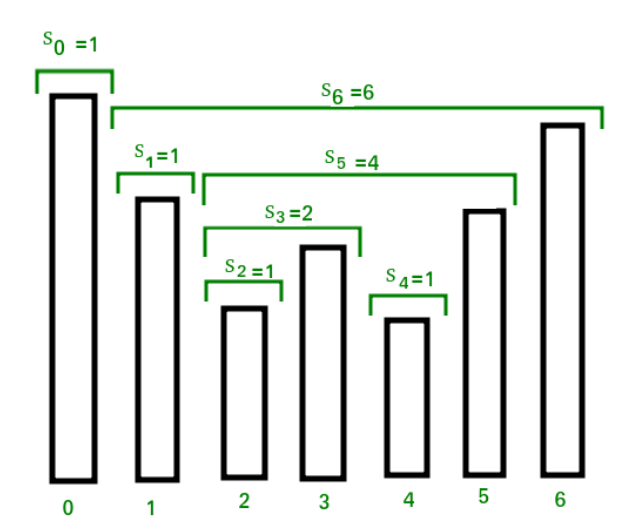
will be 1, 80 is smaller than 100 so the

span is 1, 60 is smaller than 80 so the

span is 1, 70 is greater than 60 so the

span is 2 and so on. Hence the output will

be 1 1 1 2 1 4 6.



3. Given a set of **N** nuts of different sizes and **N** bolts of different sizes. There is a one-one mapping between nuts and bolts. Match nuts and bolts efficiently.

Comparison of a nut to another nut or a bolt to another bolt is not allowed. It means nut can only be compared with bolt and bolt can only be compared with nut to see which one is bigger/smaller.  
The elements should follow the following order ! # $ % & \* @ ^ ~ .

**Input:**

N = 5

nuts[] = {@, %, $, #, ^}

bolts[] = {%, @, #, $ ^}

**Output:**

# $ % @ ^

# $ % @ ^

N = 9

nuts[] = {^, &, %, @, #, \*, $, ~, !}

bolts[] = {~, #, @, %, &, \*, $ ,^, !}

**Output:**

! # $ % & \* @ ^ ~

! # $ % & \* @ ^ ~

4.You are given an array price where prices[i] denotes the price of a stock on the ith day. You want to maximize the profit by buying a stock and then selling it at a higher price.

Suppose you can do at most k transactions (k buys and k sells), what is the maximum profit that you can make?

Note:

Return 0 if you cannot make a profit.

You cannot buy/hold more than 1 stock at a time.

You need to sell a stock before buying again.

You can sell a stock and buy it again on the same day.

* prices: [6, 1, 4, 2, 5, 3]
* k: 2
* Answer: 6
* Explanation
* Buy on day 2 (price: 1) and Sell on day 3 (price: 4).
* Buy on day 4 (price: 2) and Sell on day 5 (price: 5).
* Profit: (4 - 1) + (5 - 2) = 6.

5.Given two version numbers, compare them.

A version number consists of one or more revisions connected by a dot.  
Each revisions consists of digits and may contain leading zeroes. Each revision consists atleast one digit.

Revisions are 0-indexed from left to right.  
To compare two versions, compare revisions in the left-to-right order. Revisions are compared using their integer value ignoring any leading zeroes.

Example 1

version 1: 1.1.0  
version 2: 1.2.0

version 2 > version 1.

Example 2

version 1: 1.001.2  
version 2: 1.1.2

version 2 = version 1.

Example 3

version 1: 1.100.2  
version 2: 1.1.2

version 2 < version 1.