**Recursion-2**

**Remove X**

**Send Feedback**

Given a string, compute recursively a new string where all 'x' chars have been removed.

**Input format :**

String S

**Output format :**

Modified String

**Constraints :**

1 <= |S| <= 10^3

where |S| represents the length of string S.

**Sample Input 1 :**

xaxb

**Sample Output 1:**

ab

**Sample Input 2 :**

abc

**Sample Output 2:**

abc

**Sample Input 3 :**

xx

**Sample Output 3:**

**Remove Duplicates Recursively**

**Send Feedback**

Given a string S, remove consecutive duplicates from it recursively.

**Input Format :**

String S

**Output Format :**

Output string

**Constraints :**

1 <= |S| <= 10^3

where |S| represents the length of string

**Sample Input 1 :**

aabccba

**Sample Output 1 :**

abcba

**Sample Input 2 :**

xxxyyyzwwzzz

**Sample Output 2 :**

xyzwz

**Merge Sort Code**

**Send Feedback**

Sort an array A using Merge Sort.

Change in the input array itself. So no need to return or print anything.

**Input format :**

Line 1 : Integer n i.e. Array size

Line 2 : Array elements (separated by space)

**Output format :**

Array elements in increasing order (separated by space)

**Constraints :**

1 <= n <= 10^3

**Sample Input 1 :**

6

2 6 8 5 4 3

**Sample Output 1 :**

2 3 4 5 6 8

**Sample Input 2 :**

5

2 1 5 2 3

**Sample Output 2 :**

1 2 2 3 5

**Quick Sort Code**

**Send Feedback**

Sort an array A using Quick Sort.

Change in the input array itself. So no need to return or print anything.

**Input format :**

Line 1 : Integer n i.e. Array size

Line 2 : Array elements (separated by space)

**Output format :**

Array elements in increasing order (separated by space)

**Constraints :**

1 <= n <= 10^3

**Sample Input 1 :**

6

2 6 8 5 4 3

**Sample Output 1 :**

2 3 4 5 6 8

**Sample Input 2 :**

5

1 5 2 7 3

**Sample Output 2 :**

1 2 3 5 7

**Tower Of Hanoi - Problem**

**Send Feedback**

Tower of Hanoi is a mathematical puzzle where we have three rods and n disks. The objective of the puzzle is to move all disks from source rod to destination rod using third rod (say auxiliary). The rules are :

1) Only one disk can be moved at a time.

2) A disk can be moved only if it is on the top of a rod.

3) No disk can be placed on the top of a smaller disk.

Print the steps required to move n disks from source rod to destination rod.

Source Rod is named as 'a', auxiliary rod as 'b' and destination rod as 'c'.

**Input Format :**

Integer n

**Output Format :**

Steps in different lines (in one line print source and destination rod name separated by space)

**Constraints :**

0 <= n <= 20

**Sample Input 1 :**

2

**Sample Output 1 :**

a b

a c

b c

**Sample Input 2 :**

3

**Sample Output 2 :**

a c

a b

c b

a c

b a

b c

a c