**Recursion Assignment**

1. **Geometric Sum**

#### Given k, find the geometric sum i.e.

1 + 1/2 + 1/4 + 1/8 + ... + 1/(2^k)

#### using recursion.

##### Input format :

Integer k

##### Output format :

Geometric sum

##### Constraints :

0 <= k <= 1000

##### Sample Input 1 :

3

##### Sample Output 1 :

1.875

##### Sample Input 2 :

4

##### Sample Output 2 :

1.93750

1. **Check Palindrome (recursive)**

**Send Feedback**

Check whether a given String S is a palindrome using recursion. Return true or false.

**Input Format :**

String S

**Output Format :**

'true' or 'false'

**Constraints :**

0 <= |S| <= 1000

where |S| represents length of string S.

**Sample Input 1 :**

racecar

**Sample Output 1:**

true

**Sample Input 2 :**

ninja

**Sample Output 2:**

false

1. **Sum of digits (recursive)**

**Send Feedback**

Write a recursive function that returns the sum of the digits of a given integer.

**Input format :**

Integer N

**Output format :**

Sum of digits of N

**Constraints :**

0 <= N <= 10^9

**Sample Input 1 :**

12345

**Sample Output 1 :**

15

**Sample Input 2 :**

9

**Sample Output 2 :**

9

1. **Multiplication (Recursive)**

**Send Feedback**

Given two integers M & N, calculate and return their multiplication using recursion. You can only use subtraction and addition for your calculation. No other operators are allowed.

**Input format :**

Line 1 : Integer M

Line 2 : Integer N

**Output format :**

M x N

**Constraints :**

0 <= M <= 1000

0 <= N <= 1000

**Sample Input 1 :**

3

5

**Sample Output 1 :**

15

**Sample Input 2 :**

4

0

**Sample Output 2 :**

0

1. **Count Zeros**

**Send Feedback**

Given an integer N, count and return the number of zeros that are present in the given integer using recursion.

**Input Format :**

Integer N

**Output Format :**

Number of zeros in N

**Constraints :**

0 <= N <= 10^9

**Sample Input 1 :**

10204

**Sample Output 1 :**

2

**Sample Input 2 :**

708000

**Sample Output 2 :**

4

1. **String to Integer**

**Send Feedback**

Write a recursive function to convert a given string into the number it represents. That is input will be a numeric string that contains only numbers, you need to convert the string into corresponding integer and return the answer.

**Input format :**

Numeric string S (string, Eg. "1234")

**Output format :**

Corresponding integer N (int, Eg. 1234)

**Constraints :**

0 <= |S| <= 9

where |S| represents length of string S.

**Sample Input 1 :**

1231

**Sample Output 1 :**

1231

**Sample Input 2 :**

12567

**Sample Output 2 :**

12567

1. **Pair star**

**Send Feedback**

Given a string S, compute recursively a new string where identical chars that are adjacent in the original string are separated from each other by a "\*".

**Input format :**

String S

**Output format :**

Modified string

**Constraints :**

0 <= |S| <= 1000

where |S| represents length of string S.

**Sample Input 1 :**

hello

**Sample Output 1:**

hel\*lo

**Sample Input 2 :**

aaaa

**Sample Output 2 :**

a\*a\*a\*a

1. **Check AB**

**Send Feedback**

Suppose you have a string, S, made up of only 'a's and 'b's. Write a recursive function that checks if the string was generated using the following rules:

a. The string begins with an 'a'

b. Each 'a' is followed by nothing or an 'a' or "bb"

c. Each "bb" is followed by nothing or an 'a'

If all the rules are followed by the given string, return true otherwise return false.

**Input format :**

String S

**Output format :**

'true' or 'false'

**Constraints :**

0 <= |S| <= 1000

where |S| represents length of string S.

**Sample Input 1 :**

abb

**Sample Output 1 :**

true

**Sample Input 2 :**

abababa

**Sample Output 2 :**

false

1. **Staircase**

**Send Feedback**

A child is running up a staircase with N steps, and can hop either 1 step, 2 steps or 3 steps at a time. Implement a method to count how many possible ways the child can run up to the stairs. You need to return number of possible ways W.

**Input format :**

Integer N

**Output Format :**

Integer W

**Constraints :**

1 <= N <= 30

**Sample Input 1 :**

4

**Sample Output 1 :**

7

**Sample Input 2 :**

5

**Sample Output 2 :**

13