

Lecture 3: Recursion 1

Power

Write a program to find x to the power n (i.e. x^n). Take x and n from the user. You need to return the answer.

Do this recursively.

Input format :

Two integers x and n (separated by space)

Output Format :

x^n (i.e. x raise to the power n)

Constraints :

$0 \leq x \leq 30$

$0 \leq n \leq 30$

Sample Input 1 :

3 4

Sample Output 1 :

81

Sample Input 2 :

2 5

Sample Output 2 :

32

Print Numbers

[Send Feedback](#)

Given is the code to print numbers from 1 to n in increasing order recursively. But it contains few bugs that you need to rectify such that all the test cases pass.

Input Format :

Integer n

Output Format :

Numbers from 1 to n (separated by space)

Constraints :

$1 \leq n \leq 10000$

Sample Input 1 :

6

Sample Output 1 :

1 2 3 4 5 6

Sample Input 2 :

4

Sample Output 2 :

1 2 3 4

Number of Digits

[Send Feedback](#)

Given the code to find out and return the number of digits present in a number recursively. But it contains few bugs, that you need to rectify such that all the test cases should pass.

Input Format :

Integer `n`

Output Format :

Count of digits

Constraints :

$1 \leq n \leq 10^6$

Sample Input 1 :

156

Sample Output 1 :

3

Sample Input 2 :

7

Sample Output 2 :

1

Sum of Array

[Send Feedback](#)

Given an array of length N, you need to find and return the sum of all elements of the array.

Do this recursively.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Output Format :

Sum

Constraints :

$1 \leq N \leq 10^3$

Sample Input 1 :

3
9 8 9

Sample Output 1 :

26

Sample Input 2 :

3
4 2 1

Sample Output 2 :

7

Check Number

[Send Feedback](#)

Given an array of length N and an integer x, you need to find if x is present in the array or not. Return true or false.

Do this recursively.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Line 3 : Integer x

Output Format :

'true' or 'false'

Constraints :

$1 \leq N \leq 10^3$

Sample Input 1 :

```
3
9 8 10
8
```

Sample Output 1 :

```
true
```

Sample Input 2 :

```
3
9 8 10
2
```

Sample Output 2 :

```
false
```

First Index of Number

[Send Feedback](#)

Given an array of length N and an integer x, you need to find and return the first index of integer x present in the array. Return -1 if it is not present in the array.

First index means, the index of first occurrence of x in the input array.

Do this recursively. Indexing in the array starts from 0.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Line 3 : Integer x

Output Format :

first index or -1

Constraints :

$1 \leq N \leq 10^3$

Sample Input :

```
4
9 8 10 8
8
```

Sample Output :

```
1
```

Last Index of Number

[Send Feedback](#)

Given an array of length N and an integer x, you need to find and return the last index of integer x present in the array. Return -1 if it is not present in the array.

Last index means - if x is present multiple times in the array, return the index at which x comes last in the array.

You should start traversing your array from 0, not from (N - 1).

Do this recursively. Indexing in the array starts from 0.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Line 3 : Integer x

Output Format :

last index or -1

Constraints :

$1 \leq N \leq 10^3$

Sample Input :

```
4
9 8 10 8
8
```

Sample Output :

```
3
```

All Indices of Number

[Send Feedback](#)

Given an array of length N and an integer x, you need to find all the indexes where x is present in the input array. Save all the indexes in an array (in increasing order).

Do this recursively. Indexing in the array starts from 0.

Input Format :

Line 1 : An Integer N i.e. size of array

Line 2 : N integers which are elements of the array, separated by spaces

Line 3 : Integer x

Output Format :

indexes where x is present in the array (separated by space)

Constraints :

$1 \leq N \leq 10^3$

Sample Input :

```
5
9 8 10 8 8
8
```

Sample Output :

```
1 3 4
```


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 :

$0 \leq M \leq 1000$

$0 \leq N \leq 1000$

Sample Input 1 :

3

5

Sample Output 1 :

15

Sample Input 2 :

4

0

Sample Output 2 :

0

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 :

$0 \leq N \leq 10^9$

Sample Input 1 :

0

Sample Output 1 :

1

Sample Input 2 :

00010204

Sample Output 2 :

2

Explanation for Sample Output 2 :

Even though "00010204" has 5 zeros, the output would still be 2 because when you convert it to an integer, it becomes 10204.

Sample Input 3 :

708000

Sample Output 3 :

4

Geometric Sum

[Send Feedback](#)

Given k, find the geometric sum i.e.

$$1 + 1/2 + 1/4 + 1/8 + \dots + 1/(2^k)$$

using recursion.

Input format :

Integer k

Output format :

Geometric sum (upto 5 decimal places)

Constraints :

$0 \leq k \leq 1000$

Sample Input 1 :

3

Sample Output 1 :

1.87500

Sample Input 2 :

4

Sample Output 2 :

1.93750

Explanation for Sample Input 1:

$$1 + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} = 1.87500$$

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 :

$0 \leq |S| \leq 1000$

where |S| represents length of string S.

Sample Input 1 :

racecar

Sample Output 1:

true

Sample Input 2 :

ninja

Sample Output 2:

false

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 :

$0 \leq N \leq 10^9$

Sample Input 1 :

12345

Sample Output 1 :

15

Sample Input 2 :

9

Sample Output 2 :

9