

CS5101: Design and Analysis of Algorithms

Recursion

Note: The problems below can be solved by different methods too. But here you are required to solve them by Recursive Method.

Q1. Given a sorted array of n distinct integers and a target integer, find the index of the target element using binary search implemented recursively. If the target is present in the array, return its 0-based index. If the target is not present, return -1.

Example

Input

```
7                // (Size of array)
2 5 8 12 16 23 38 // (Array elements)
16              // (Target Value)
```

Output

```
4                // (Required Index)
```

TEST CASES

1. Target at the Beginning

Input

```
6
3 7 11 15 19 24
3
```

Expected Output

```
0
```

2. Target Greater than every element

Input

```
8
5 12 18 23 31 47 56 72
100
```

Expected Output

```
-1
```

3. Negative Numbers and Zero

Input

```
9
-20 -15 -8 -3 0 4 9 16 25
-8
```

Expected Output

```
2
```

Q2. Given an array A of n integers, a majority element is an element that appears strictly more than $\lfloor n/2 \rfloor$ times in the array. Your task is to find and return the majority element of the given array. If no majority element exists, return -1.

Example

Input

7 // (Size of array)
2 2 1 1 1 2 2 // (Array Elements)

Output

2

TEST CASES

1. No Majority Element

Input

6
1 2 3 4 5 6

Expected Output

-1

2. Majority Appears Exactly Half the Time

Input

6
3 3 3 4 4 4

Expected Output

-1

3. Negative Numbers

Input

7
-5 -5 -5 2 3 -5 8

Expected Output

-5

Q3. A palindrome is a word, number, phrase, or sequence that reads the same forward and backward. Given a string s, determine whether the string is a palindrome. A string is a palindrome if it remains exactly the same when its characters are read from right to left. Examples of palindromes include level, radar, and 121.

Example

Input

level

Output

Yes

TEST CASES

1. Single Character

Input

a

Expected Output

Yes

2. Mixed letters and digits

Input

alb1a

Expected Output

Yes

3. Not a Palindrome

Input

abbcaba

Expected Output

No

Q4. Given a string s, reverse the string and return the resulting reversed string.

Requirements:

The solution must use recursion to reverse the string.

Do not use loops such as for, while, or do-while.

Do not use any built-in function that directly reverses a string.

The order of all characters, including spaces, digits, and special characters, must be preserved in reverse order.

Example

Input

hello

Output

olleh

TEST CASES

1. Mixed Uppercaase and Lowercase

Input

AbCdEf

Expected Output

fEdCbA

2. Numeric String

Input

123456789

Expected Output

987654321

3. Palindrome***Input***

malayalam

Expected Output

malayalam

Q5. There are **n** people arranged in a circle and numbered from **1 to n**. A person **p** ($1 \leq p \leq n$) is designated as the starting person.

Starting from person **p**, counting proceeds in the increasing order of the person numbers around the circle. After reaching person **n**, counting continues from person **1**. Thus, the counting order starting from **p** is:

p, p+1, p+2, ..., n, 1, 2, ..., p-1, p, ...

Every **m-th** person is eliminated. The person from whom the next count begins is the next surviving person in the increasing-order direction. The process continues until only one person remains. The sequence in which the people are eliminated is called the Josephus permutation.

Given **n**, **m**, and **p**, determine:

The complete Josephus elimination sequence.

The last surviving person.

Example***Input***

5 // (user input for n)

2 // (user input for m)

1 // (user input for p)

Output

2, 4, 1, 5 // (Elimination sequence)

3 // (last survivor)

TEST CASES**1. *Input***

7

3

4

Expected Output

6, 2, 7, 5, 3, 1

4

2. *Input*

10

3

5

Expected Output

7, 10, 4, 8, 3, 1, 2, 6, 9

5

3. *Input*

12

5

4

Expected Output

8, 1, 7, 2, 10, 6, 5, 9, 12, 4, 11

3

Q6. You are given a rectangular board of size $2 \times n$ and an unlimited number of 2×1 dominoes. Each domino can be placed in either of the following orientations:
Vertically — occupying 2 rows and 1 column.
Horizontally — occupying 1 row and 2 columns.
Determine the number of different ways to completely tile the $2 \times n$ board using these dominoes and no two dominoes may overlap. You must solve the problem using recursion.

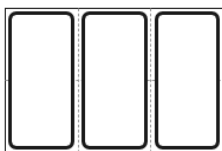
Example

Input

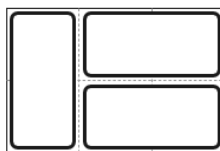
$n = 3$

Output

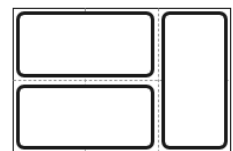
3



Tiling 1



Tiling 2



Tiling 3

TEST CASES

1. *Input*

5

Expected Output

8

2. Input

6

Expected Output

13

3. Input

4

Expected Output

5

Q7. A subsequence of a sequence is obtained by deleting zero or more elements while maintaining the relative order of the remaining elements. Given a string *s* of length *n*, generate all possible subsequences of the string using recursion. The empty string is also considered a valid subsequence.

Example

Input

abc

Output

""

"a"

"b"

"c"

"ab"

"ac"

"bc"

"abc"

TEST CASES

1. Input

abcd

Expected Output

""

"a"

"b"

"c"

"d"

"ab"

"ac"

"ad"

"bc"

```
"bd"  
"cd"  
"abc"  
"abd"  
"acd"  
"bcd"  
"abcd"
```

2. *Input*

```
aaa
```

Expected Output

```
""  
"a"  
"a"  
"a"  
"aa"  
"aa"  
"aa"  
"aaa"
```

Q8. You are given k sorted arrays, where each array is sorted in non-decreasing order. Merge all k arrays into a single sorted array containing all the elements from the given arrays. The merging operation must be implemented using recursion.

Example

Input

```
3           //no of sorted arrays  
[1, 4, 7]   //sorted array 1  
[2, 5, 8]   //sorted array 2  
[3, 6, 9]   //sorted array 3
```

Output

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```

TEST CASES

1. *Input*

```
3  
[1, 10]  
[2, 3, 8, 15]  
[4, 5, 6, 7, 20]
```

Expected Output

```
[1, 2, 3, 4, 5, 6, 7, 8, 10, 15, 20]
```

2. ***Input***

3

[1, 3, 5]

[1, 2, 5]

[2, 4, 5]

Expected Output

[1, 1, 2, 2, 3, 4, 5, 5, 5]

3. ***Input***

2

[1, 4, 7, 10]

[2, 3, 8, 9]

Expected Output

[1, 2, 3, 4, 7, 8, 9, 10]

Q9. Given an array of n integers, determine whether the array is sorted in non-decreasing order using recursion. An array is sorted in non-decreasing order if: $A[i] \leq A[i+1]$, for every valid index i. The solution must use recursion to check the array.

Example

Input

5 // (array size)

1 2 3 4 5 // (array elements)

Output

true

TEST CASES

1. ***Input***

5

1 3 2 4 5

Expected Output

false

2. ***Input***

7

1 2 2 3 3 4 5

Expected Output

true

3. ***Input***

6

-10 -7 -7 -3 0 5

Expected Output

true

Q10. Given an array of n integers and a target value, determine the number of times the target occurs in the array using recursion. The solution must examine the array recursively and return the total number of occurrences of the target.

Example

Input

6
2 5 2 7 2 9
2

Output

3

TEST CASES

1. ***Input***

8
1 3 5 7 9 11 13 15
4

Expected Output

0

2. ***Input***

6
5 5 5 5 5
5

Expected Output

6

3. ***Input***

9
-2 4 -2 7 3 -2 8 -5 -2
-2

Expected Output

4
