Round 1

1. Which Data Structure is mainly used for implementing the recursive algorithm? (Easy) (Programming Fundamentals - DSA)
a) Queue
b) Stack
c) Linked List
d) Tree
Answer: b

2. What's happen if base condition is not defined in recursion? (Easy) (Programming Fundamentals - DSA)
a) Stack underflow
b) Stack Overflow
c) None of these
d) Both a and b
Answer: b

3. Choose the correct answer. (Easy) (Programming Fundamentals - DSA)
a) Recursion is always better than iteration.
b) Recursion uses more memory compared to iteration.
c) Recursion uses less memory compared to iteration.
d) Iterative function is always better and simpler to write than recursion.
Answer: b

4. Recursion is similar to which of the following? (Easy) (Programming Fundamentals - DSA)
a) Switch Case
b) Loop
c) If-else
d) None
Answer: b

5. Recursion is a method in which the solution of a problem depends on ..... (Easy) (Programming Fundamentals - DSA)
a) Larger instances of different problems
b) Larger instances of same problem
c) Smaller instances of same problem
d) Smaller instances of different problems
Answer: c

6. Which of the following problem cannot be solved using recursion? (Easy) (Programming Fundamentals - DSA)
a) Tower of Hanoi
b) Fibonacci series

c) Tree Traversal
d) Problems without base case
Answer: d

7. What's the output of the following code ? (Medium) (Programming Fundamentals - JavaScript)

```
function doSomething(a, b)
{
if (b==1)
return a;
else
return a + doSomething(a,b-1);
}
doSomething(2,3);
```

a) 4
b) 2
c) 3
d) 6
Answer: d

8. In recursion the condition after which the function will stop calling itself is (Easy) (Programming Fundamentals - DSA)
a) Base condition
b) Function call
c) Both
d) None
Answer: a

9. consider the code snippet given below:- (Medium) (Programming Fundamentals - JavaScript)
var count = [1 , , 3];
what is your observation?
a) The omitted value takes "undefined"
b) This results in an error
c) This results in an exception
d) None of the above
Answer a

10. What is the observation made in the following JavaScript code? (Easy) (Programming Fundamentals - JavaScript)

```
if (!a[i]) continue;
```

a) Skips the defined elements
b) Skips the existent elements
c) Skips the null elements

d) Skips the defined & existent elements
Answer: c

11. .The function definitions in JavaScript begins with (Easy) (Programming Fundamentals - JavaScript)
a) Identifier and Parantheses
b) Return type and Identifier
c) Return type, Function keyword, Identifier and Parantheses
d) Identifier and Return type
Answer: b

12. when does function name becomes optional in javascript? (Easy) (Programming Fundamentals - JavaScript)
a) When the function is defined as a looping statement
b) When the function is defined as expressions
c) When the function is predefined
d) All of the mentioned
Answer: b

13.The length property specifies the number of elements in the ..... array in JavaScript. (Easy) (Programming Fundamentals - JavaScript)
A) special
B) sparse
C) heterogeneous
D) dense
Answer: d

14. If an array with five elements a=[1,2,3,4,5]; what will do the expression a.length=0 (Easy) (Programming Fundamentals - JavaScript)
A) checks length of array is 0 or not
B) deletes all elements
C) replaces all elements with 0
D) adds 0 at the beginning
Answer: b

15. In JavaScript array methods, ... does not provide a way to terminate iteration before all elements have been passed to the function. (Easy) (Programming Fundamentals - JavaScript)
A) for loop
B) for/in
C) forEach
D) every( )
Answer: c

16. The method or operator used to identify the array is _____ (Easy) (Programming Fundamentals - JavaScript)
a) isarrayType()

b) ==
c) ===
d) typeof
Answer: d

17. The primary purpose of the array map() function is that it _____ (Easy) (Programming Fundamentals - JavaScript)
a) maps the elements of another array into itself
b) passes each element of the array and returns the necessary mapped elements
c) passes each element of the array on which it is invoked to the function you specify, and returns an array containing the values returned by that function
d) pass the elements of the array into another array
Answer: c

18. The complexity of the sorting algorithm measures the ..... as a function of the number n of items to be sorter. (Easy) (Programming Fundamentals - JavaScript)
a) average time
b) running time
c) average-case complexity
d) case-complexity
Answer: b

19. What will be the output of the following JavaScript code? (Easy) (Programming Fundamentals - JavaScript)
```
var arr=[1,2,3];
var rev=arr.reverse();
document.writeln(rev);
```
a) 1, 2, 3
b) 3, 2, 1
c) 3
d) 1
Answer: b

20. Sorting algorithm can be characterized as ..... (Easy) (Programming Fundamentals - DSA)
a) Simple algorithm which requires the order of n2 comparisons to sort n items.
b) Sophisticated algorithms that require the O(nlog2n) comparisons to sort items.
c) Both of the above
d) None of the above
Answer: c

21. The _____ operator in JavaScript returns "object" for arrays. **(Easy) (Programming Fundamentals - JavaScript)**
a) and
b) or
c) not

d) typeof
Answer: d

22. Which of the following sorting algorithm is of divide and conquer type? (Easy) (Programming Fundamentals - DSA)
a) Bubble sort
b) Insertion sort
c) Merge sort
d) Selection sort
Answer: c

23. Which of the following sorting algorithm is of priority queue sorting type? (Easy) (Programming Fundamentals - DSA)
a) Bubble sort
b) Insertion sort
c) Merge sort
d) Selection sort
Answer: d

24. Partition and exchange sort is ..... (Easy) (Programming Fundamentals - DSA)
a) quick sort
b) tree sort
c) heap sort
d) bubble sort
Answer: a

25. What will be the output of the following JavaScript code?

```
var sum=0;
var arr = [10,15,20,30];
arr.forEach(function myFunction(element)
{
        sum= sum+element;
});
console.log(sum);
```

a) 70
b) 75
c) 10
d) error
Answer: b

26. The operation that combines the element is of A and B in a single sorted list C with n=r+s element is called .... (Easy) (Programming Fundamentals - DSA)
a) Inserting
b) Mixing

c) Merging
d) Sharing
Answer: c

27. While iterating the elements of an array a, what will do the following expression (Easy) (Programming Fundamentals - JavaScript)
for(var i=0; i<a.length; i++) {
if(a[i]= = = undefined) continue; // loop body
}
a) Skip null, undefined and non existent elements
b) Skip non existent elements
c) Skip undefined and non existent elements
d) Skip inherited properties
Answer: c

28. The function used to modify the way of sorting the keys of records is called ..... **(Easy) (Programming Fundamentals - DSA)**
a) Indexing function
b) Hash function
c) Addressing function
d) All of the above
Answer: b

29. arrays always use numbered indexes. (Easy) (Programming Fundamentals - JavaScript)
a) Yes
b) No
c) Can be yes or no
d) Can not say
Answer: a

30. The time complexity of heap sort is .... (Easy) (Programming Fundamentals - DSA)
a) O(n)
b) O(logn)
c) O(n2)
d) O(n logn)
Answer: d

Round 2

Problem Statement 1

You are given two strings **X** and **Y** of the same length. Each string contains **N** Lower case character (from 'a' to 'z'). A shift operation will remove the first character of a string and add the same character at the end of that string. For example after you perform a shift operation on a string 'abcd', the new string will be 'bcda'. If you perform this operation two times, the new string will be 'cdab'. You need to use some (maybe none) shift operations on the string **Y** to maximize the length of the longest common prefix of **X** and **Y.** If more than one result can be found pick the one that use smallest number of shift operations.

Input

The first line of the input contains a single integer **N**. The second and the third line contains the string **X** and **Y** respectively.

Output

Contains a single integer which is the number of shift operations.

Constraints

30 points:

- **$1 \leq N \leq 5000$**

30 points:

- **$1 \leq N \leq 10^4$**

40 points:

- **$1 \leq N \leq 10^6$**

Sample Input:

5
ccadd
bddcc

Sample Output:

3

Explanation :-

In this case we have make ccadd as addcc by shift dda that is 3 shift to maximize the longest common prefix with bddcc
longest common prefix is ddcc

Js Template:

```
process.stdin.resume(); // A Readable Stream that points to a standard input stream (stdin)
process.stdin.setEncoding('utf-8'); // so that the input doesn't transform

let inputString1 = "";
```

```
let inputString = "";
let currentLine = 0;

process.stdin.on('data', function (userInput) {
    inputString1 = inputString1 + userInput; // taking the input string
});

process.stdin.on('end', function(x) {
    inputString1.trim();
    inputString1 = inputString1.split("\n"); // end line
    for(let i = 0; i< inputString1.length ; i++ ) {
        inputString += inputString1[i].trim() + ' ';
    }
    inputString.trim();
    inputString = inputString.split(" ");
    main();
});

function readline() {
    let result = inputString[currentLine++];
    return  result;
}
function main() {
                let n = parseInt(readline());
                let a = readline();
                let b = readline();
       // write your logic here

}
```

Solution :-
https://ideone.com/wlguGq

Problem Statement 2

You are given a score log of a football match between two teams. Every time when one of the teams scored a goal, the name of that team was written in the score log on a separate line.
At the end of the match, the team that scored strictly more goals than the other team wins the match. If both teams scored an equal number of goals, the match ends in a tie. Determine the name of the winning team or that the match ended in a tie.

Input

- The first line of the input contains a single integer **T** denoting the number of test cases. The description of **T** test cases follows.

- The first line of each test case contains a single integer **n** denoting the number of records in the score log (the total number of goals scored during the match).
- Each of the following **n** lines contains a string **s** denoting the name of the team that scored a goal.
- **Output**

For each test case, print a single line containing one string — the name of the winning team or "Draw" (without quotes) in case of a tie.

Constraints

- $1 \le \mathbf{T} \le 10^5$
- $0 \le \mathbf{n} \le 10^5$
- $1 \le |\mathbf{s}| \le 20$
- **s** consists only of lowercase English letters
- sum of **n** over all test cases $\le 10^5$
- for each test case, there are at most two different team names

Sample Input 1

2
4
ab
bc
bc
ab
3
xxx
yyy
yyy

Sample Output 1

Draw
yyy

Solution:-

https://ideone.com/sKBTWX

Problem Statement 3

You are given an integer n, find the nth fibonacci number using **recursion** only.

Constraint

• **1 <= n <= 40**

Input Format

• The single line of input consists of a single integer n

Output Format

• Print the n-th fibonacci number, starting from 0th eg in sequence 0 1 1 2 3
0 is the 0th number, 2 is the 3rd number

Sample Input 1

4

Sample Output 1

3

Explanation of Sample 1

0 1 1 2 3 -> 4th number in the fibonacci series is 3 (starting from 0th)

Sample Input 2

0

Sample Output 2

0

Explanation of Sample 2

0 is the first element in fibonacci series

Sample Input 3

12

Sample Output 3

144

Explanation of Sample 3

0 1 1 2 3 5 8 13 21 34 55 89 **144,** 144 is the 12th element in fibonacci series

Solution:

https://ideone.com/W7KA1H

Round 3

Employee database - Create an application that will store employee details such as
id, name, age, department, position
We should be able to perform CRUD on employee database

GET /employee/:id - fetch employee by id
GET /employee/all - fetch all employee
POST /employee - update employee details
DELETE /employee/:id - delete an employee
PUT /employee - insert new employee

Instructions:- create an employee table in database with the following fields id,name,age,department,position and In this application, first we will create a connection with database and then create API calls. These API calls will handle CRUD operations. CRUD stands for create, read, update, and Delete.

Solution
https://github.com/shrey8599/employeeManagementBackend