Fullstack Development Module Test

Round 1

1. . What are the different types of data types in JavaScript?
A) Primitive
B) Reference Type
C) All of the above
D) None of the above
Correct Ans: All of the above

2. The below statement is an example of:
while (3==3) {}
A) Typographical error
B) An infinite loop
C) An illegal javascript statement
D) None of the above
Correct Ans: An infinite loop

3. How many rows are in array declared as::
let array = [[0, 0, 3,7], [0, 0, 2,5],[16, 0, 0 ,1]];
A) 2
B) 3
C) 4
D) 5
Correct Ans: 3

4. What is the index poisition of first element of a 2D array?
A) 0,0
B) 0,1
C) 1,1
D) 1,0
Correct Ans: 0,0

5. What does the following code does?
let A = [[1,2,3],[3,4,5]]
sum=0
for(let i=0;i<2;i++)
for(j=0;j<3;j++)
sum+=A[i][j]
console.log(sum)
A) Sum of all the elements
B) Sum of elements in first row
C) Sum of element in last row
D) Sum of element in first column
Correct Ans: Sum of all the elements

6. Any change in the variable inside the function will make a change in the value of the outside variable. This is possible on passing variable to a function as:
A) Value
B) Reference
C) Name
D) None of these
Correct Ans: Reference

7. What will be the output of the below code snippet?
const numbers = [15.5, 2.3, 1.1, 4.7];
console.log(numbers.reduce((total, num) => {return total + Math.round(num)}, 0));

A) 20
B) 24
C) 1314
D) Undefined
Correct Ans: 24

8. Which of the following can be stored inside Arrays in JavaScript?
A) Primitive types
B) Object
C) Both a and b
D) None of these
Correct Ans: Both a and b

9. Which of the following statement is true about JSON and JavaScript Objects?
A) Key should always be in double quotes in JSON
B) JSON and JS objects both resembles key-value pairs
C) Both a and b
D) None of the above
Correct Ans: Both a and b

10. What will be the output of the below code snippet?
let arr = [1, 2, 3, 4, 5];
console.log(arr.map(x => x*x));
A) [1, 2, 3, 4, 5]
B) [1, 4, 9, 16, 25]
C) Undefined
D) None of these
Correct Ans: [1, 4, 9, 16, 25]

11. Which of the following is not a requirement for a recursive function?
A) It has two base cases.
B) It has a recursive case
C) Its recursive case approaches a base case.
D) None of these
Correct Ans: It has two base cases.

12. A linearly recursive function always has the recursive call at the ___ of the function.
A) Start
B) End
C) Either place
D) Some where in middle
Correct Ans: Either place

13. What does the following function does? (s is a string)
function mystery(s) { if (s==="") return 0;
else return(1 + mystery(s+1));
}
A) Find length of string
B) Compare string
C) Convert string to char
D) Find char in string
Correct Ans: Find length of string

14. Recursive approach to find power of a number is preferred over iterative approach.
A) TRUE
B) FALSE
C) Not Sure
D) None of these
Correct Ans: FALSE

15. function sum(n) {
if (n==0)
return n;
else
return n + sum(n-1);
}
What will be the output for sum(8)
A) 40
B) 36
C) 8
D) 15
Correct Ans: 36

16. Which of the following problem cannot be solved using recursion?
A) Tower of Hanoi
B) Fibonacci series
C) Tree Traversal
D) Problems without base case
Correct Ans: Problems without base case

17. What this code snippet is trying to achieve:

```
function fun(a, b)
{
if (b == 0)
return 0;
if (b % 2 == 0)
return fun(a + a, Math.floor(b/2));
return fun(a + a, Math.floor(b/2)) + a;
}
```

A) a raised to the power b.
B) a * b
C) log(a+a)/log(b)
D) Sum of b geometric permutation starting from a.

Correct Ans: a*b

18. Infinite recursion leads to .....
A) Overflow of run-time stack
B) Underflow of registers usage
C) Overflow of I/O cycles
D) Underflow of run-time stack
Correct Ans: Overflow of run-time stack

19. Consider the following recursive implementation used to find the length of a string:

```
function recursive_get_len( s, len)
{
if(s[len] === undefined){
return 0;
}
return _____;
}
let s = "abcdef";
let len = recursive_get_len(s,0);
console.log(len)
```

Which of the following lines should be inserted to complete the above code?
A) 1
B) len
C) recursive_get_len(s, len+1)
D) 1 + recursive_get_len(s, len+1)
Correct Ans: 1 + recursive_get_len(s, len+1)

20. Consider the following recursive implementation to find the sum of digits of number:

```
function recursive_sum_of_digits(n)
{
if(n == 0)
return 0;
return _________;
}
let n = 1201;
let ans = recursive_sum_of_digits(n);
console.log(ans)
```

Which of the following lines should be inserted to complete the above code?

A) (n / 10) + recursive_sum_of_digits(n % 10)
B) (n) + recursive_sum_of_digits(n % 10)
C) (n % 10) + recursive_sum_of_digits(Math.floor(n / 10))
D) (n % 10) + recursive_sum_of_digits(n % 10)

Correct Ans: (n % 10) + recursive_sum_of_digits(Math.floor(n / 10))

21. What is the output of the following code?

```
function my_recursive_function( n)
{
if(n == 0)
return
console.log(n)
my_recursive_function(n-1);
}
my_recursive_function(10);
```

A) 10
B) 1
C) 10 9 8 7 6 5 4 3 2 1
D) 10 9 8 7 6 5 6 7 8 9 10

Correct Ans: 10 9 8 7 6 5 4 3 2 1

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

```
var a1 = [,,,];
var a2 = new Array(3);
0 in a1
0 in a2
```

A) false false
B) false true
C) true true
D) false true

Correct Ans: false false

23. What is the observation made in the following JavaScript code?
var count = [1,,3];
A) The omitted value takes “undefined”
B) This results in an error
C) This results in an exception
D) The omitted value takes an integer value
Correct Ans: The omitted value takes “undefined”

24. Consider the following code:
function recursive_sum(n)
{
if(n == 0)
return 0;
return _____;
}
let n = 5;
let ans = recursive_sum(n);
console.log(ans)
Which of the following lines is the recurrence relation for the above code ?
A) (n - 1) +recursive_sum(n)
B) n + recursive_sum(n)
C) n + recursive_sum(n - 1)
D) (n -1) + recursive_sum(n - 1)
Correct Ans: n + recursive_sum(n - 1)

25. Which of the following sorting techniques is stable?
A) Quick sort
B) Counting sort
C) Heap sort
D) Selection sort
Correct Ans: Counting sort

26. function fn(num) {
if(num < 2) {
return num;
}
else {
return fn(num-1) + fn(num - 2);
}
}
console.log(fn(6))
What will be the output for sum()
A) 5
B) 8
C) 13
D) 3
Correct Ans: 8

27. Which of the following is a Divide and Conquer algorithm?
A) Buble
B) Selection
C) Merge
D) Quick
Correct Ans: Merge

28. What is the advantage of counting sort over quick sort?
A) Counting sort has lesser time complexity when range is comparable to number of input elements
B) Counting sort has lesser space complexity
C) Counting sort is not a comparison based sorting technique
D) It has no advantage
Correct Ans: Counting sort has lesser time complexity when range is comparable to number of input elements

29. Partition and exchange sort is ��..
A) quick sort
B) tree sort
C) heap sort
D) bubble sort
Correct Ans: quick sort

30. What is the disadvantage of selection sort?
A) It requires auxiliary memory
B) It is not scalable
C) It can be used for small keys
D) It takes linear time to sort the elements
Correct Ans: It is not scalable

31. Predict the output of below code:
```
function factorial(number){
if(number == 0){
return 1;
}
return factorial(number - 1);
}
console.log(factorial(5));
```
A) 0
B) 1
C) 120
D) Error
Correct Ans: 1

32. The given array is arr = {1,2,3,4,5}. (bubble sort is implemented with a flag variable)The number of iterations in selection sort and bubble sort respectively are _____
A) 5 and 4
B) 1 and 4
C) 0 and 4
D) 4 and 1
Correct Ans: 4 and 1

33. How many queues are needed to implement a stack. Consider the situation where no other data structure like arrays, linked list is available to you.
A) 1
B) 2
C) 3
D) 4
Correct Ans: 2

34. When does underflow happens?
A) When the queue is empty and deque is performed
B) When the queue is full and insertion is performed
C) In either cases
D) In neither cases
Correct Ans: When the queue is empty and deque is performed

35. What does the following code does?

```
if (this.isEmpty()) {
return undefined;
}
let result = this.items[this.front];
this.items[this.front] = 0;
this.front++;
return result;
```

A) remove element from back
B) remove element from front of queue
C) insert element to front of queue
D) insert elemenet to back of queue
Correct Ans: remove element from front of queue

36. The popular notion to describe stack is?
A) Last in First out
B) Frist in First out
C) None of the above
D) Both a and b
Correct Ans: Last in First out

37. Postfix form of following expression.
D + (E * F)
A) EF * D+
B) DEF * +
C) DEF +*
D) EFD *+
Correct Ans: DEF * +

38. The process of inserting an element in the stack is called?
A) Enqueue
B) Insert
C) Push
D) Pop
Correct Ans: Push

39. What will be the space complexity of the program where we use an additional stack for storing n elements?
A) O(nlogn)
B) O(n)
C) O(1)
D) O(logn)
Correct Ans: O(n)

40. In which of the following type of LinkedList, the next pointer is never null?
A) Singly LL
B) Doubly LL
C) Circular LL
D) None of the above
Correct Ans: Circular LL

Round 2

Problem-1:

Problem Statement

Six friends go on a trip and are looking for accommodation. After looking for hours, they find a hotel which offers two types of rooms — double rooms and triple rooms. A double room costs Rs. X, while a triple room costs Rs. Y.
The friends can either get three double rooms or get two triple rooms. Find the minimum amount they will have to pay to accommodate all six of them.

Input Format

- The first line contains a single integer T - the number of test cases. Then the test cases follow.
- The first and only line of each test case contains two integers X and Y - the cost of a double room and the cost of a triple room.

Output Format

For each testcase, output the minimum amount required to accommodate all the six friends.

Constraints

- $1 \leq T \leq 100$
- $1 \leq X < Y \leq 100$

Sample 1:

Input

```
3
10 15
6 8
4 8
```

Output

```
30
16
12
```

Sample 2:

Input

```
3
40 30
60 80
40 80
```

Output

```
60

160

120
```

Sample 3:

Input

```
4
80 90
38 50
50 80
100 150
```

Output

180

100

150

300

Sample 4:

Input

```
5
750 900
348 530
500 840
1000 1500
2000 3000
```

Output

1800

1044

1500

3000

6000

Sample 5:

Input

```
7
7050 9900
3408 5030
5070 8490
10080 15090
20800 30900
7657 9798
7900 3434
```

Output

19800

10060

15210

30180

61800

19596

6868

Explanation:

Test case 1: The friends can take three double rooms and thus pay a total of Rs. 30.
Test case 2: The friends can take two triple rooms and thus pay a total of Rs. 16.
Test case 3: The friends can take three double rooms and thus pay a total of Rs. 12.

Template:

```
const t = parseInt(readline());

function sixFriends () {
for (let i = 1; i<=t; i++) {
// Implementation logic here
}
}

sixFriends();
```

Solution:

```
const t = parseInt(readline());

function sixFriends () {
for (let i = 1; i<=t; i++) {
let [x, y] = readline().split(' ');
console.log(Math.min(2 * parseInt(y) , 3 * parseInt(x)));
}
}

sixFriends();
```

IDEOne link - https://www.ideone.com/8OdSnm

Problem-2:

Given an absolute path for a file as A. Return the string A as a simplified absolute path

Note:
Path is in Unix System format
In Unix-style file system:

- A period '.' refers to the current directory.
- A double period '..' refers to the directory up a level.
- Any multiple consecutive slashes '//' are treated as a single slash '/'.

In Simplified Absolute path:

- The path starts with a single slash '/'.
- Any two directories are separated by a single slash '/'.
- The path doesn't end with trailing slashes '/'.
- The path only contains the directories on the path from the root directory to the target file or directory (i.e., no period '.' or double period '..')
- Path will not have whitespace characters.

Input Format
`The only argument given is string A.`

Output Format
`Return a string denoting the simplified absolue path for a file (Unix-style).`

Samples 1:

```
Input:
    A = "/home/"
Output:
    "/home"
```

Samples 2:

```
Input:
    A = "/a/./b/../../c/"
Output:
    "/c"
```

Template

```
let simplifyPath = function(a){

      }
```

const A = readline();

```
res = simplifyPath(A)
console.log(res)
```

Solution:

```
let simplifyPath = function(a){
        a = a.split("/");
        res = [];
        for(var i=0;i<a.length;i++) {
            if(!a[i] || a[i] == ".") continue;
            if(a[i] == "..") res.pop();
            else res.push(a[i]);
        }
        return "/"+res.join("/");
      }
```

const A = readline();

```
res = simplifyPath(A)
console.log(res)
```

Round 3

In this round you have to create a backend application for a librarian to manage books in a university.

Functionality for the application users will be as follow:

Admin Functionality:

1. I should be able to log in
2. I should be able to Add Books
3. I should be able to Edit Books
4. I should be able to Delete Books
5. I should be able to get all book loan details
6. I should be able to approve loan request
7. I should be able to delete loan request
8. I should be able to get all the books
9. I should be able to send notification to user

User Functionality:

1. I should be able to sign up
2. I should be able to log in
3. I should be able to get my loan details
4. I should be able to create loan request for book
5. I should be able to get all the books
6. I should be able to get all the notification
7. I should be able to delete the notiication

Note:

1. Loan of book signifies lending of book of the user from admin
2. Please follow the proper folder structure as taught in class and maintain the code quality

Round 4

In this round, you have to create a frontend application for a librarian to manage books in a university.

Functionality for the application users will be as follow:

Admin Functionality:

0. I should be able to log in
0. I should be able to Add Books
0. I should be able to Edit Books
0. I should be able to Delete Books
0. I should be able to get all book loan details
0. I should be able to approve loan request
0. I should be able to delete loan request
0. I should be able to get all the books
0. I should be able to send notification to user

User Functionality:

0. I should be able to sign up
0. I should be able to log in
0. I should be able to get my loan details
0. I should be able to create loan request for book
0. I should be able to get all the books
0. I should be able to get all the notification
0. I should be able to delete the notiication

Note:

1. Loan of book signifies lending of book of the user from admin
2. Please follow the proper folder structure as taught in class.

Solution:

Please refer to the git repo for the sol of both Rounds 3 & 4:

https://github.com/singh-atul/Library_Management/tree/main/library_management