

# ADS CCEE Practice Quiz 2

Total points 17/20 ?

✓ What is the output of the following recursive function call? \* 1/1

```
public class Main
{
    static void printBinary(int n) {
        if (n > 1) {
            printBinary(n / 2);
        }
        System.out.print(n % 2);
    }

    public static void main(String[] args) {
        printBinary(13);
    }
}
```

- ☒ a) 1101 ✓
- ☐ b) 1011
- ☐ c) 0110
- ☐ d) Compilation Error

✓ What is the time complexity for inserting an element at the beginning of an array if array has n elements in it ? \*1/1

- ☒ a.  $O(n)$  ✓
- ☐ b.  $O(1)$
- ☐ c.  $O(\log n)$
- ☐ d.  $O(n \log n)$

Name: \*

Prathamesh Patkar  
.....

✓ What is the output of the following recursive function call? \*

1/1

```
public class Main
{
    static void printString(String str) {
        if (str.length() == 0) {
            return;
        }
        printString(str.substring(1));
        System.out.print(str.charAt(0) + " ");
    }

    public static void main(String[] args) {
        printString("hello");
    }
}
```

☐ a) hello☒ b) olleh ✓☐ c) hll o e☐ d) Compilation Error

Centre: \*

☒ Kharghar☐ Juhu

✓ public class Main

\*

1/1

```
{  
    static int power(int base, int exponent) {  
        if (exponent == 0) {  
            return 1;  
        }  
        return base * power(base, exponent - 1);  
    }  
  
    public static void main(String[] args) {  
        System.out.println(power(5, 4));  
    }  
}
```

☐ a) Compilation Error

☐ b) 225

☒ c) 625

✓

☐ d) 125

✗ The postfix form of the expression  $(A + B) * (C * D - E) * F / G$  is? \*

0/1

☐  $AB + CD * E - FG /**$

☐  $AB + CD * E - F **G /$

☐  $AB + CD * E - *F *G /$

☒  $AB + CDE * - *F *G /$

✗

Correct answer

☒  $AB + CD * E - *F *G /$

✓ What is the time complexity for accessing an element in an array? \* 1/1

☐  $O(n^2)$

☐  $O(\log n)$

☒  $O(1)$



☐  $O(n)$

✓ Which of the following is the disadvantage of the array? \* 1/1

☐ Stack and Queue data structures can be implemented through an array.

☐ Index of the first element in an array can be negative

☒ Wastage of memory if the elements inserted in an array are lesser than the allocated size



☐ Elements can be accessed sequentially.

✓ Which of the following is the infix expression? \* 1/1

☒  $A+B*C$



☐  $+A*BC$

☐  $ABC+*$

☐ None of the above

- ✓ Consider the usual algorithm for determining whether a sequence of parentheses is balanced. Suppose that you run the algorithm on a sequence that contains 2 left parentheses and 3 right parentheses (in some order). The maximum number of parentheses that appear on the stack AT ANY ONE TIME during the computation? \*1/1

- ☐ 1
- ☒ 2 ✓
- ☐ 3
- ☐ 4 or more

- ✓ What does the following Java code do? \* 1/1

```
public Object function()
```

```
{  
    if(isEmpty())  
        return -999;  
    else  
    {  
        Object high;  
        high = q[front];  
        return high;  
    }  
}
```

- ☐ Dequeue
- ☐ Enqueue
- ☒ Return the front element ✓
- ☐ Return the last element

✗ What is the output of the following recursive function call? \*

0/1

```
public class Main
{
    static int factorial(int n) {
        if (n == 0) {
            return 1;
        }
        return n * factorial(n - 1);
    }

    public static void main(String[] args) {
        factorial(5);
    }
}
```

☐ a) Compilation Error

☐ b) 60

☒ c) 120

✗

☐ d) No Output

Correct answer

☒ d) No Output

✓ Java uses \_\_\_ type of memory to implement Recursion. \*

1/1

☐ a) Heap

☒ b) Stack

✓

☐ c) Register

☐ d) None

✓ Which of the following is not a real-life example of a Queue? \*

1/1

- ☐ Waiting in line to order food at a restaurant
- ☐ Waiting in line to buy movie tickets
- ☐ Managing tasks on a CPU
- ☒ Using a stack of dishes



PRN \*

240840320073

✓ What is tail recursion? \*

1/1

- ☒ a) A type of recursion where the function calls itself at the end of each recursive call ✓
- ☐ b) A type of recursion where the function calls itself at the beginning of each recursive call
- ☐ c) A type of recursion where the function does not call itself
- ☐ d) A type of recursion where the function uses a loop instead of recursion

✗ What is the output of the following recursive function call? \*

0/1

```
public class Main
{
    int sumDigits(int n) {
        if (n == 0) {
            return 0;
        }
        return n % 10 + sumDigits(n / 10);
    }

    public static void main(String[] args) {
        System.out.println(sumDigits(1234));
    }
}
```

☐ a) No output

☒ b) 10

✗

☐ c) 11

☐ d) Compilation Error

Correct answer

☒ d) Compilation Error

✓ What is recursion? \*

1/1

☐ a) A loop that executes until a condition is met

☒ b) A function that calls itself

✓

☐ c) A data structure that holds a collection of elements

☐ d) An algorithm that sorts data in ascending order



✓ What is the base case in recursion? \*

1/1

- ☐ a) The case where the recursion starts
- ☒ b) The case where the recursion ends ✓
- ☐ c) The case where the recursion reaches its maximum depth
- ☐ d) The case where the recursion encounters an error

✓ What will be the output of the program? \*

1/1

```
class Exam{
public static void main(String abc[] ){
int x =10;
int y= 15;

if(x++ >10 && -y>10){

}
System.out.print(x+","+y);
    if(x++>10 && -y>10){
}
System.out.print(x+","+y);
}
}
```

- ☐ 11,15 11,14
- ☒ 11,15 12,14 ✓
- ☐ 10,15 12,14
- ☐ 11,14 11,15

✓ Consider the following operation performed on a stack of size 5. \*1/1

Push(1);

Pop();

Push(2);

Push(3);

Pop();

Push(4);

Pop();

Pop();

Push(5);

After the completion of all operation, the number of elements present in stack is?

☒ 1



☐ 2

☐ 3

☐ 4

✓ What is the disadvantage of using recursion? 1/1

☐ a) It is slower than iterative solutions

☐ b) It is harder to implement than iterative solutions

☒ c) It can lead to stack overflow errors



☐ d) It cannot be used to solve complex problems

This content is neither created nor endorsed by Google. [Report Abuse](#) - [Terms of Service](#) - [Privacy Policy](#)

Google Forms

