MCQ Test (RECURSION) Total Marks: 10
Name of the Student: _____
Branch / Class: _____

Q. 1. What is the output of the following code snippet?

```
class Demo{
 public static int specialAdd(int num1) {
      if (num1!=0)
         return (num1+2)+specialAdd(num1-1) ;
      elsereturn 3;
 }
 public static int extraordinaryAdd(int num2) {
       if (num2!=0)
          return
specialAdd(num2)+extraordinaryAdd(num2-1) ;
        elsereturn 0;
 }
  public static void main (String [ ] args) {
         System.out.println( (extraordinaryAdd(5) ) ) ;
 }
}
```

a. 80
b. 52
c. 70
d. 25

Q. 2.

```
public class RecTest {
   public static void mystery(int n) {
      if (n <= 0) return;
      mystery(n - 1);
      System.out.print(n + " ");
      mystery(n - 2);
   }

   public static void main(String[] args) {
      mystery(3);
   }
}
```

a. 1 2 3 1
b. 1 2 3 2 1
c. 3 2 1 2 1
d. 1 2 1 3 2 1

Q. 3.

```
class Main{
static void fun(int n) {
   if (n == 0) return;
   fun(n / 2);
   System.out.print(n % 2);
}
public static void main(String[] args) {
   fun(13);
}
}
```

a. 1011
b. 1101
c. 1110
d. 0110

Q. 4.

```
class Main{
static int reverse(int n, int rev) {
   if (n == 0) return rev;
   return reverse(n / 10, rev * 10 + n % 10);
}
public static void main(String[] args) {
   System.out.println(reverse(1234, 0));
}
}
```

a. 1234
b. 0
c. 4321
d. Compilation Error

Q. 5.

```
class Main{
static void zigzag(int n) {
   if (n <= 0) return;
   System.out.print(n + " ");
   zigzag(n - 1);
   System.out.print(n + " ");
   zigzag(n - 2);
   System.out.print(n + " ");
}
public static void main(String[] args) {
   zigzag(2);
}
}
```

a. 2 1 1 2 1 2
b. 1 1 2 2 1 1
c. Infinity
d. 2 1 1 1 2 2

Q. 6.

```
class Main{
static int mystery(int n) {
   if (n == 0) return 1;
   return n * mystery(n - 1) + mystery(n - 1);
}
public static void main(String[] args) {
   System.out.println(mystery(3));
}
}
```

a. 24
b. 26
c. 32
d. 22

Q. 7.

```
class Main{
static void recurse(int n) {
   if (n == 0) return;
   System.out.print(n + " ");
   if (n % 2 == 0)
      recurse(n - 2);
   else
      recurse(n - 1);
   System.out.print(n + " ");
}
public static void main(String[] args) {
   recurse(4);
}
}
```

a. 4 4 2 2
b. 4 2 2 4
c. 2 4 2 4
d. 4 2 4 4

Q. 8.

```
class Main{
static int weird(int a, int b) {
   if (b == 0) return 0;
   return (b % 2 == 0) ? weird(a + a, b / 2) : weird(a + a, b /
2) + a;
}
public static void main(String[] args) {
   System.out.println(weird(3, 5));
}
}
```

a. 15
b. 8
c. 0
d. Compilation Error

Q. 9.

```
class Main{
static int oddEvenRec(int n) {
   if (n == 0) return 0;
   if (n % 2 == 0)
      return oddEvenRec(n - 1) - 1;
   else
      return oddEvenRec(n - 1) + 1;
}
public static void main(String[] args) {
   System.out.println(oddEvenRec(5));
}
}
```

a. 0
b. 1
c. 11
d. Compilation Error

Q. 10.

```
class Main{
static void trickyPrint(int n) {
   if (n == 0) return;
   trickyPrint(n - 1);
   System.out.print(n + " ");
   trickyPrint(n - 1);
}
public static void main(String[] args) {
   trickyPrint(3);
}
}
```

a. 2 1 1 2 3 1 2
b. 3 3 3 2 2 2 1
c. 1 2 1 1 2 2 3
d. 1 2 1 3 1 2 1