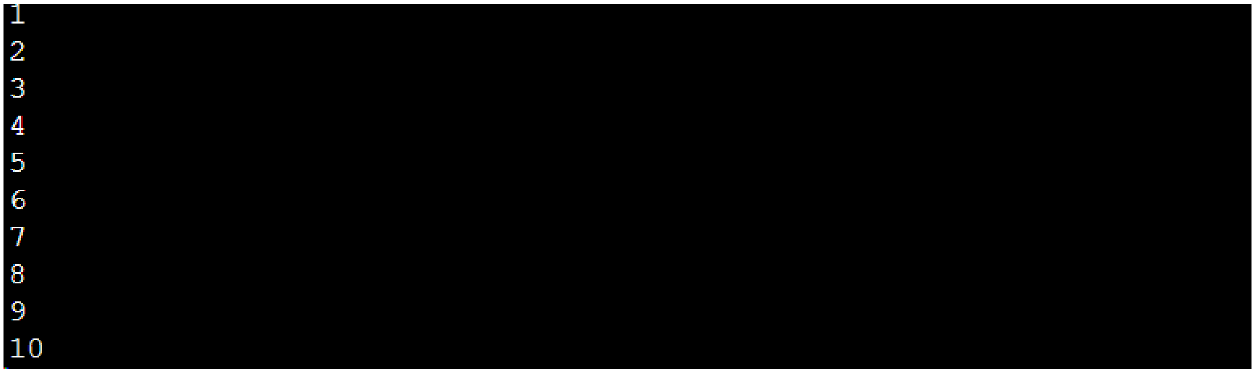
Day 2 : Special Logic Building Assignment: 10 special Recursion Programs

1. **Print 1 to n without using loops**

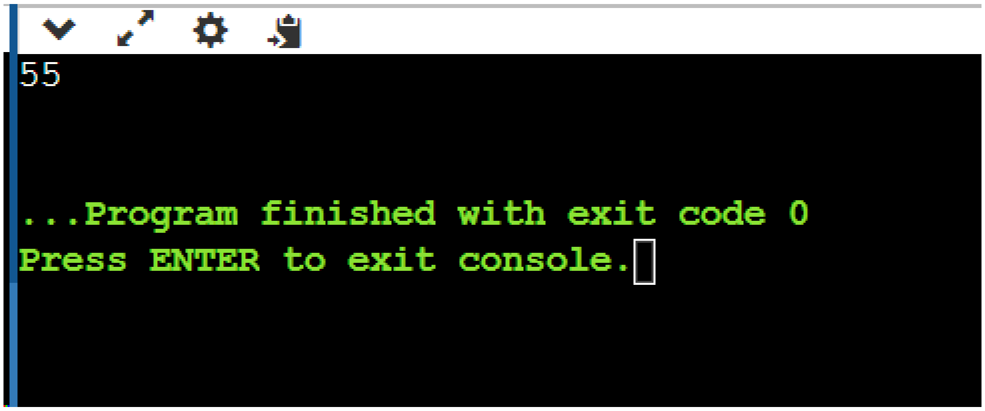
```
public class Main
{
   static void recursion(int n,int i)
   {
      if(i<n){
      System.out.println(i+1);
      recursion(n,i+1);
      }
   }
   public static void main(String[] args) {
                  recursion(10,0);
          }
}
```



```
1
2
3
4
5
6
7
8
9
10
```

2. Sum of natural numbers using recursion

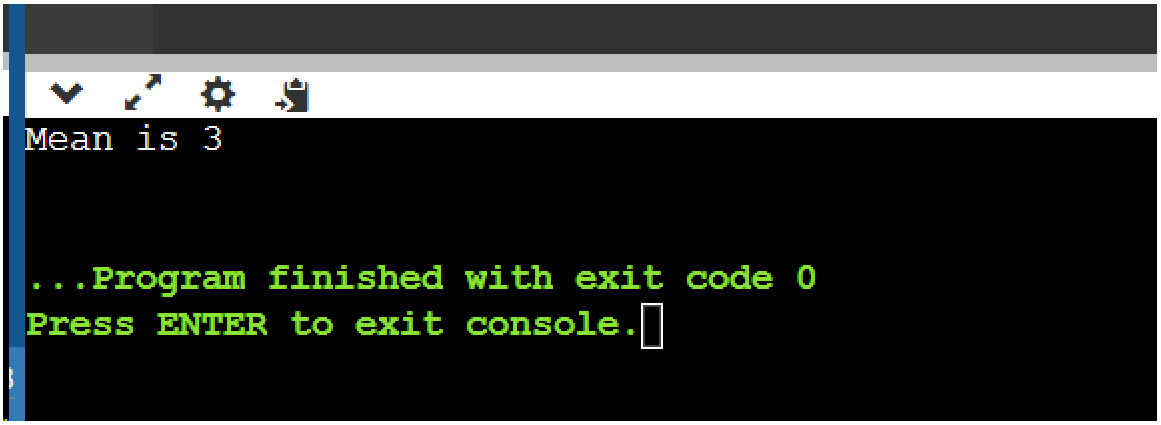
```
public class Main
{
   static int recursion(int n)
   {
      if(n==0){
         return 0;}
      return n+recursion(n-1);

   }
   public static void main(String[] args) {
                      System.out.println(recursion(10));
           }}
```



```
55


...Program finished with exit code 0
Press ENTER to exit console.
```

3. Mean of Array using Recursion

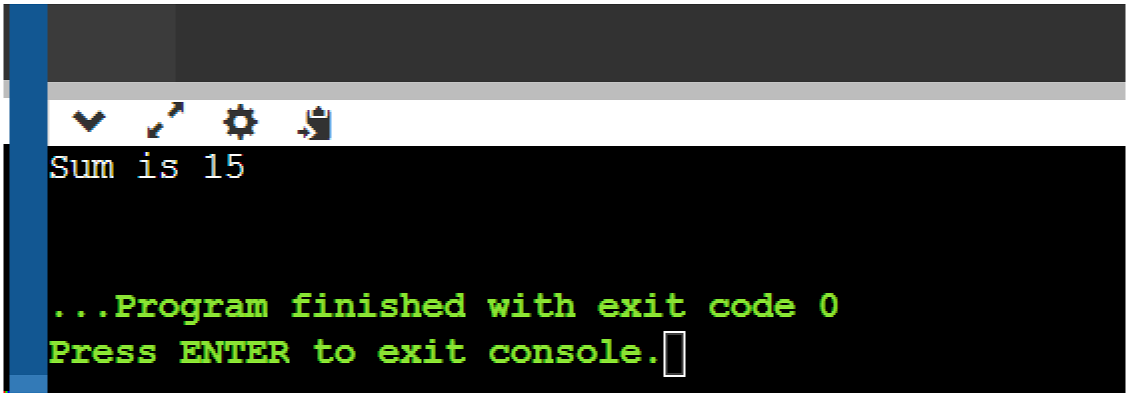
```
public class Main
{
   static int recursion(int[] arr,int length)
   {
      if(length==0)
      {
         return arr[0];
      }
      return arr[length]+recursion(arr,length-1);
   }
   public static void main(String[] args) {
      int arr[]=new int[]{1,2,3,4,5};
      int length=arr.length;
                  System.out.println("Mean is "+recursion(arr,length-1)/length);
          }
}
```



```
Mean is 3


...Program finished with exit code 0
Press ENTER to exit console.
```

4. Sum of array elements using recursion

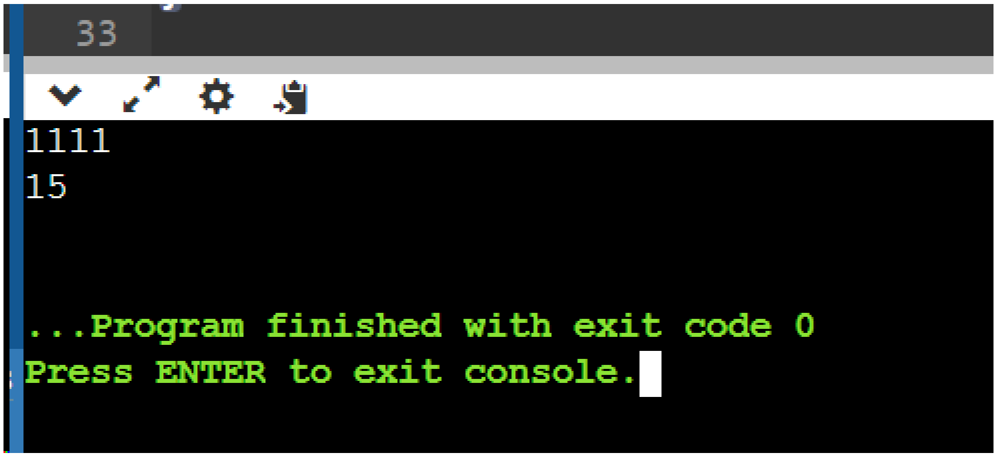
```
public class Main
{
   static int recursion(int[] arr,int length)
   {
      if(length==0)
      {
         return arr[0];
      }
      return arr[length]+recursion(arr,length-1);
   }
   public static void main(String[] args) {
      int arr[]=new int[]{1,2,3,4,5};
      int length=arr.length;
                        System.out.println("Sum is "+recursion(arr,length-1));
         }
}
```



```
Sum is 15


...Program finished with exit code 0
Press ENTER to exit console.
```

5. Decimal to binary number using recursion

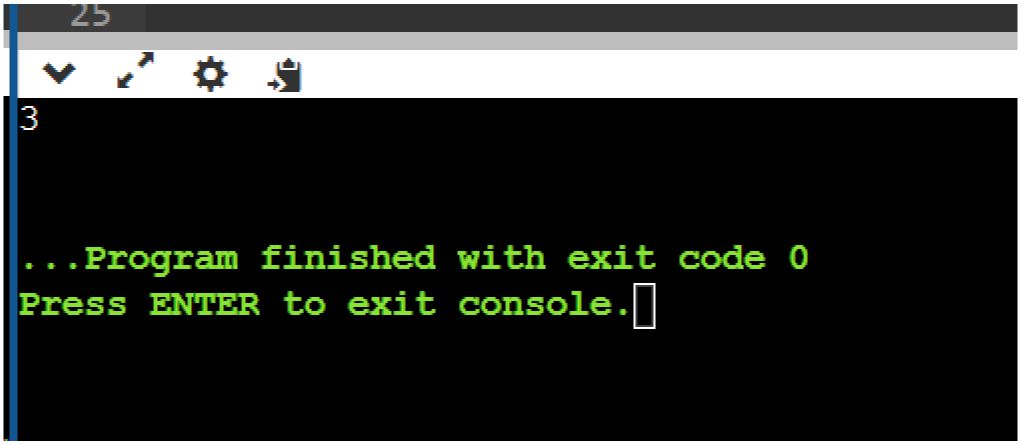
```
import java.io.*;
import java.util.*;
public class Main {
  static int i=0;
  static int sum=0;
  public static void recursion(int arr[], int length){
    if(length>=0){
      sum = sum + arr[i]*(int)Math.pow(2,length);
        i++;
        recursion(arr,length-1);
    }
    else
    {System.out.println(sum);}
  }
  public static void main(String[] args) {
    Scanner sc=new Scanner(System.in);
    int num=sc.nextInt();
    char arr[]=String.valueOf(num).toCharArray();
    int intarr[]=new int[arr.length];
    for (int i=0;i<arr.length;i++)
    intarr[i]=Integer.parseInt(String.valueOf(arr[i]));
    recursion(intarr,intarr.length-1);
  }
}
```



```
1111
15


...Program finished with exit code 0
Press ENTER to exit console.
```

6. Sum of digit of a number using recursion

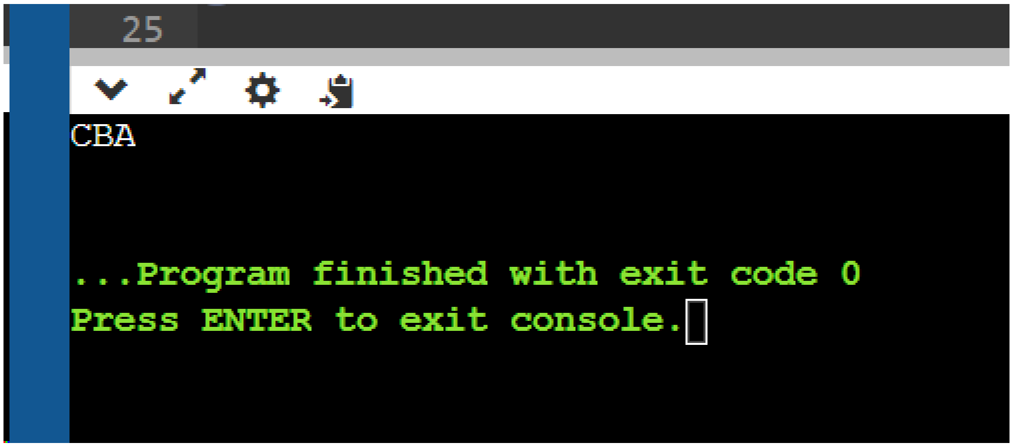
```
import java.io.*;
import java.util.*;
public class Main {
  static int i=0;
  static int sum=0;
  public static int recursive(int num){
    if(num==0){
      return num;
    }
    return num%10+(recursive(num/10));
  }
  public static void main(String[] args) {
    int n=111;
    System.out.println(recursive(n));
  }
}
```



```
3


...Program finished with exit code 0
Press ENTER to exit console.
```

7. Print reverse of a string using recursion

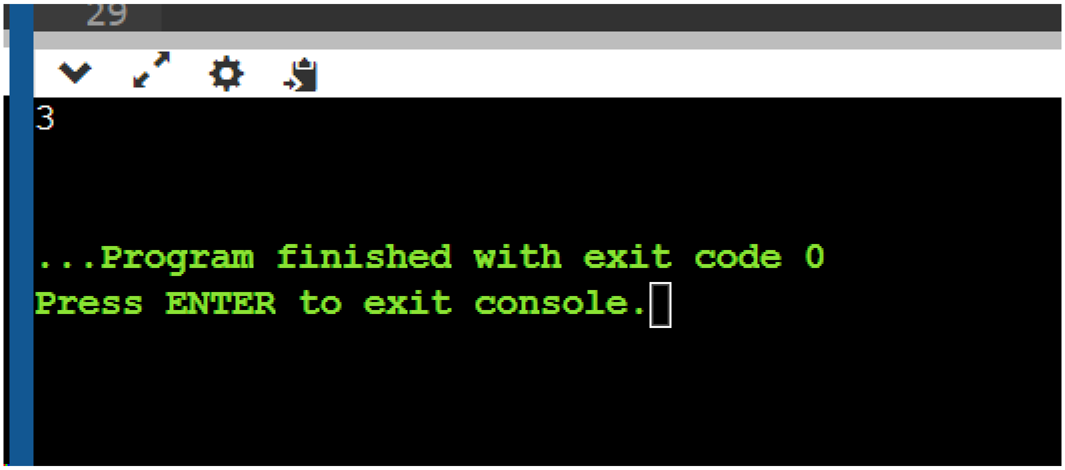
```
import java.io.*;
import java.util.*;
public class Main {
    static int i=0;
    static int sum=0;
    public static String recursive(String str, int length){
        if(length<0){
            return "";
        }
        return str.charAt(length)+recursive(str.substring(0,length),length-1);
    }
    public static void main(String[] args) {
        String str="ABC";
        System.out.println(recursive(str,str.length()-1));
    }
}
```



```
CBA


...Program finished with exit code 0
Press ENTER to exit console.
```

8. Program for length of a string using recursion

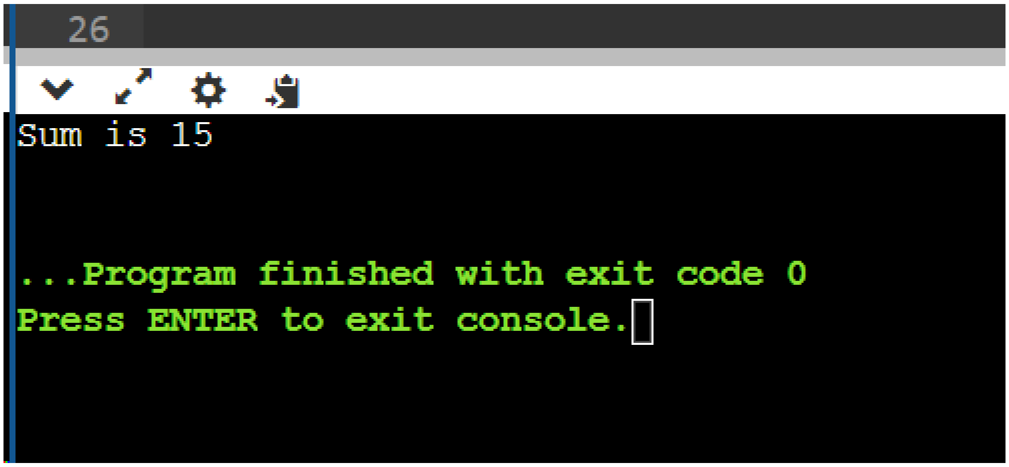
```
import java.io.*;
import java.util.*;
public class Main {
    static int i=0;
    static int sum=0;
    public static int recursive(String str, int i){
       if(!str.isEmpty()){
             sum++;
            recursive(str.substring(i+1),i);
            return sum;
       }
       else{
           return 0;
       }
    }
    public static void main(String[] args) {
        String str="ABC";
        System.out.println(recursive(str,0));
    }
}
```



```
3


...Program finished with exit code 0
Press ENTER to exit console.
```

9. **Tail recursion to calculate sum of array elements.**

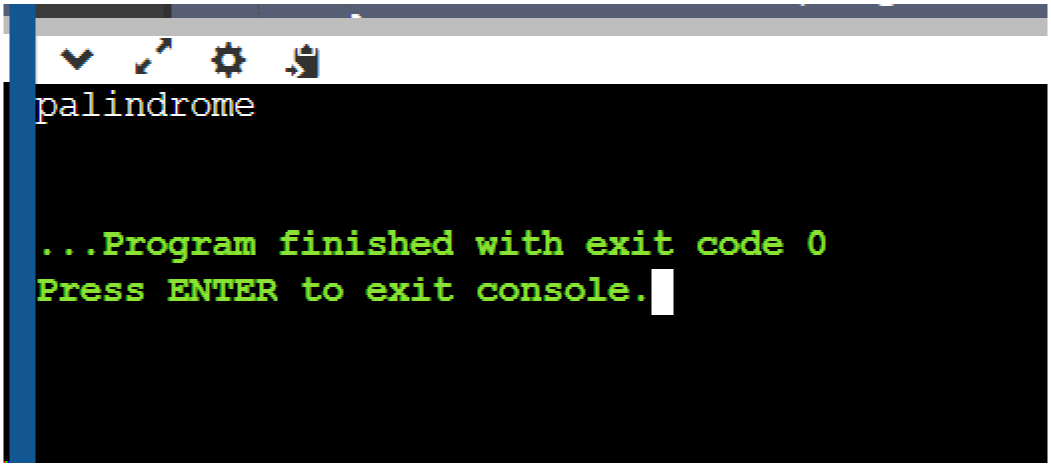
```
public class Main
{
  static int recursion(int[] arr,int length)
  {
    if(length==0)
    {
      return arr[0];
    }
    return arr[length]+recursion(arr,length-1);
  }
  public static void main(String[] args) {
    int arr[]=new int[]{1,2,3,4,5};
    int length=arr.length;
                System.out.println("Sum is "+recursion(arr,length-1));
       }
}
```



```
Sum is 15


...Program finished with exit code 0
Press ENTER to exit console.
```

10. Recursive function to check if a string is palindrome

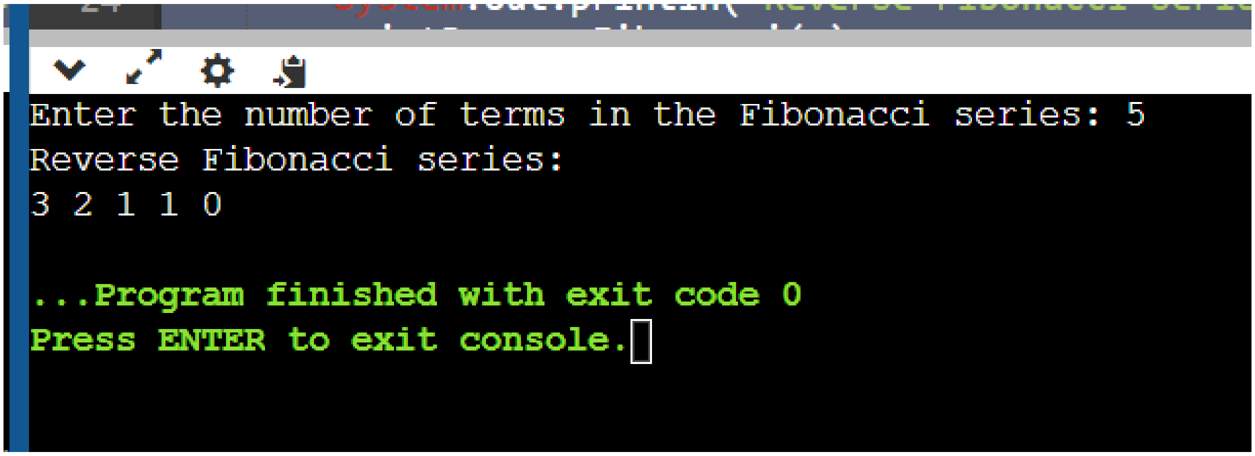
```
import java.io.*;
import java.util.*;
public class Main {
  static int i=0;
  static int sum=0;
  public static String recursive(int length,String str){
    if(length==0)
      return str;
    else{
      return recursive(length-1,str.substring(1))+str.charAt(0);
    }

  }
  public static void main(String[] args) {
    String str="madam";
    int length=str.length();
    String res=recursive(length-1,str);
    int check=str.compareTo(res);
    if(check==0)
      System.out.println("palindrome");
    else
      System.out.println("Not Palindrome");

  }
}
```



```
palindrome


...Program finished with exit code 0
Press ENTER to exit console.
```

11. Print Fibonacci Series in reverse order using Recursion

```
import java.util.Scanner;

public class Main {
   public static int fibonacci(int n) {
      if (n <= 1) {
         return n;
      } else {
         return fibonacci(n - 1) + fibonacci(n - 2);
      }
   }

   public static void printReverseFibonacci(int n) {
      for (int i = n - 1; i >= 0; i--) {
         System.out.print(fibonacci(i) + " ");
      }
   }

   public static void main(String[] args) {
      Scanner scanner = new Scanner(System.in);
      System.out.print("Enter the number of terms in the Fibonacci series: ");
      int n = scanner.nextInt();
      scanner.close();

      System.out.println("Reverse Fibonacci series:");
      printReverseFibonacci(n);
   }
}
```



```
Enter the number of terms in the Fibonacci series: 5
Reverse Fibonacci series:
3 2 1 1 0

...Program finished with exit code 0
Press ENTER to exit console.
```