

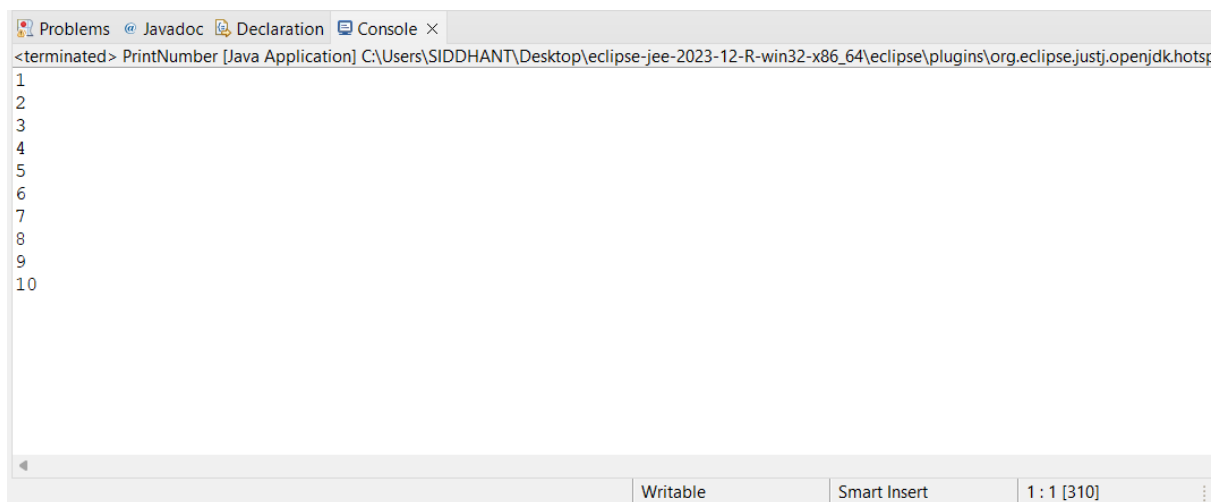
## 1. Print 1 to n without using loops

```
package com.example.main;

import java.util.Scanner;

public class PrintNumber {
    public static void printNumbers(int n) {
        if (n > 0) {
            printNumbers(n - 1);
            System.out.println(n);
        }
    }

    public static void main(String[] args) {
        int n = 10; // Change n to whatever value you want
        printNumbers(n);
    }
}
```



The screenshot shows the Eclipse IDE's console window. The title bar includes tabs for Problems, Javadoc, Declaration, and Console. The console output shows the program has terminated and printed the numbers 1 through 10, each on a new line. The status bar at the bottom indicates the file is 'Writable', has 'Smart Insert' enabled, and is at line 1, column 1 of a 310-line file.

```
<terminated> PrintNumber [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64.jre\bin\java.exe
1
2
3
4
5
6
7
8
9
10
```

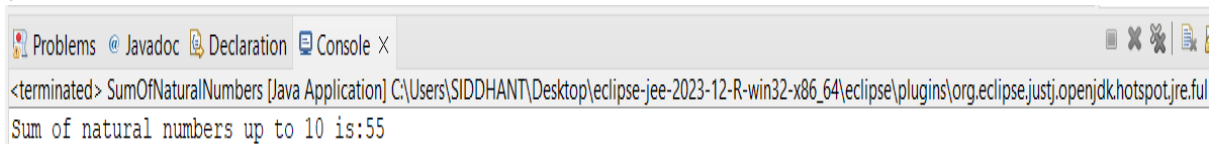
## 2. Sum of natural numbers using recursion

```
package com.example.main;

import java.util.Scanner;

public class SumOfNaturalNumbers {
    public static int sumOfNumbers(int n) {
        if (n == 0) {
            return 0;
        } else {
            return n + sumOfNumbers(n - 1);
        }
    }

    public static void main(String[] args) {
        int n = 10; // Change n to whatever value you want
        int sum = sumOfNumbers(n);
        System.out.println("Sum of natural numbers up to " + n + " is:"
+ sum);
    }
}
```



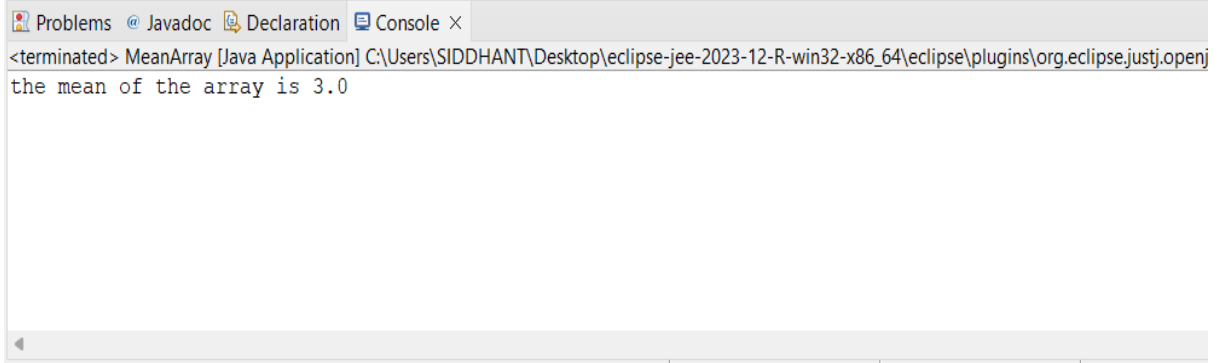
The screenshot shows the Eclipse IDE interface with the 'Console' tab selected. The console output displays the result of the program execution: 'Sum of natural numbers up to 10 is:55'. The title bar of the console window indicates the application is 'SumOfNaturalNumbers [Java Application]' running on 'C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86\_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full'.

```
<terminated> SumOfNaturalNumbers [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openjdk.hotspot.jre.full
Sum of natural numbers up to 10 is:55
```

### 3. Mean of Array using Recursion

```
package com.example.main;
```

```
class MeanArray {  
    public static double Mean(int[] arr, int n) {  
        if (n == 0)  
            return 0.0;  
        return (Mean(arr, n - 1) * (n - 1) + arr[n - 1]) / n;  
    }  
  
    public static void main(String[] args) {  
        int arr[] = { 1, 2, 3, 4, 5 };  
        System.out.println("the mean of the array is " + Mean(arr,  
arr.length));  
    }  
}
```



The screenshot shows the Eclipse IDE's console window. The title bar includes 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output displays the message 'the mean of the array is 3.0'.

```
<terminated> MeanArray [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openj  
the mean of the array is 3.0
```

### 3. Sum of array elements using recursion

```
package com.example.main;

public class Sumofarray {

    public static int arraySum(int[] arr, int n) {

        if (n <= 0) {
            return 0;
        } else {

            return arr[n - 1] + arraySum(arr, n - 1);
        }
    }

    public static void main(String[] args) {
        int[] arr = {1, 2, 3, 4, 5};
        int sum = arraySum(arr, arr.length);
        System.out.println("Sum of array elements: " + sum);
    }
}
```

Problems Javadoc Declaration Console ×  
<terminated> Sumofarray [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86\_64\eclipse\plugins\org.eclipse.justj.openjdk.  
Sum of array elements: 15

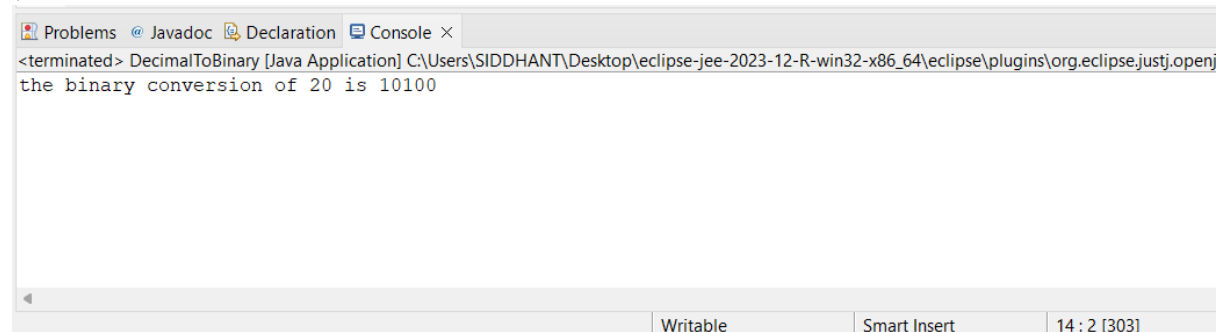
Writable Smart Insert 22:2 [470]

#### 4. Decimal to binary number using recursion

```
package com.example.main;

class DecimalToBinary {
    public static int Binary(int n) {
        if (n == 0)
            return 0;
        return Binary(n / 2) * 10 + n % 2;
    }

    public static void main(String[] args) {
        int n = 20;
        System.out.println("the binary conversion of " + n + " is " +
Binary(n));
    }
}
```



The screenshot shows the Eclipse IDE's console window. The top bar includes tabs for 'Problems', 'Javadoc', 'Declaration', and 'Console'. The console output displays the result of the program execution: 'the binary conversion of 20 is 10100'. The status bar at the bottom indicates 'Writable', 'Smart Insert', and the cursor position '14:2 [303]'.

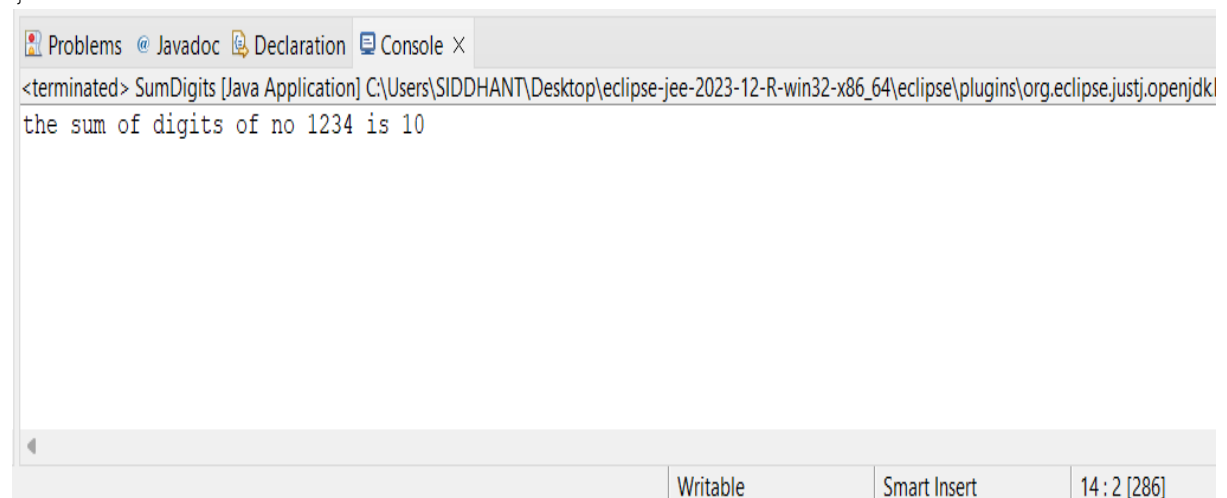
```
<terminated> DecimalToBinary [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.eclipse.justj.openj
the binary conversion of 20 is 10100
```

## 5. Sum of digit of a number using recursion

```
package com.example.main;

class SumDigits {
    public static int Sum(int n) {
        if (n == 0)
            return 0;
        return Sum(n / 10) + n % 10;
    }

    public static void main(String[] args) {
        int n = 1234;
        System.out.println("the sum of digits of no " + n + " is " +
Sum(n) );
    }
}
```



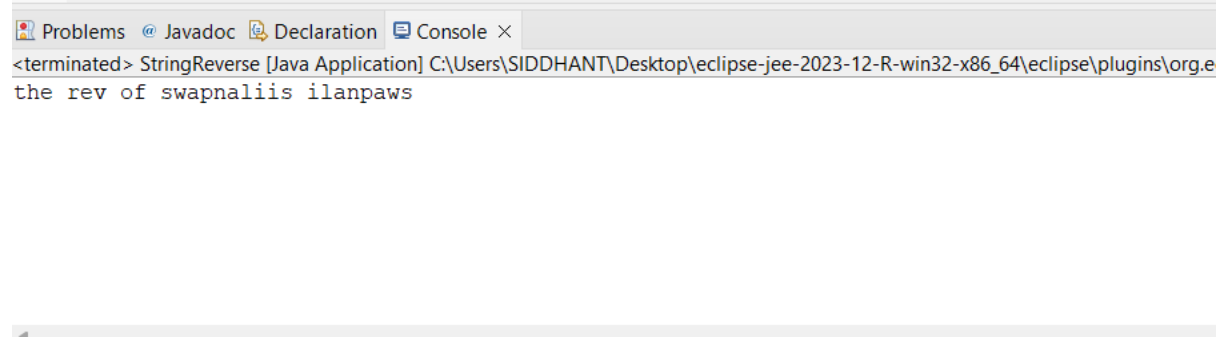
The screenshot shows the Eclipse IDE's console window. The title bar includes 'Problems', '@ Javadoc', 'Declaration', and 'Console X'. The console output displays the message: 'the sum of digits of no 1234 is 10'. The status bar at the bottom indicates 'Writable', 'Smart Insert', and '14:2 [286]'.

## 6. Print reverse of a string using recursion

```
package com.example.main;

class StringReverse {
    public static String Reverse(String input) {
        if (input.length() == 0 || input.length() == 1) {
            return input;
        }
        return Reverse(input.substring(1)) + input.charAt(0);
    }

    public static void main(String[] args) {
        String str = "swapnali";
        System.out.println("the rev of " + str + "is " + Reverse(str));
    }
}
```



The screenshot shows the Eclipse IDE interface. The top part displays the source code for the `StringReverse` class. The bottom part shows the console output, which is the result of running the program. The console output is: `<terminated> StringReverse [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86_64\eclipse\plugins\org.e` followed by the string `the rev of swapnaliis ilanpaws`.

## 7. Program for length of a string using recursion

```
package com.example.main;

class StringLength {
    public static int Length(String str) {
        if (str.length() == 0) {
            return 0;
        }
        return Length(str.substring(1)) + 1;
    }

    public static void main(String[] args) {
        String str = "siddhant";
        System.out.println("the length of " + str + " is " +
Length(str));
    }
}
```

Problems @ Javadoc Declaration Console ×

<terminated> StringLength [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-x86\_64\eclipse\plugins\org.ec

the length of siddhant is 8



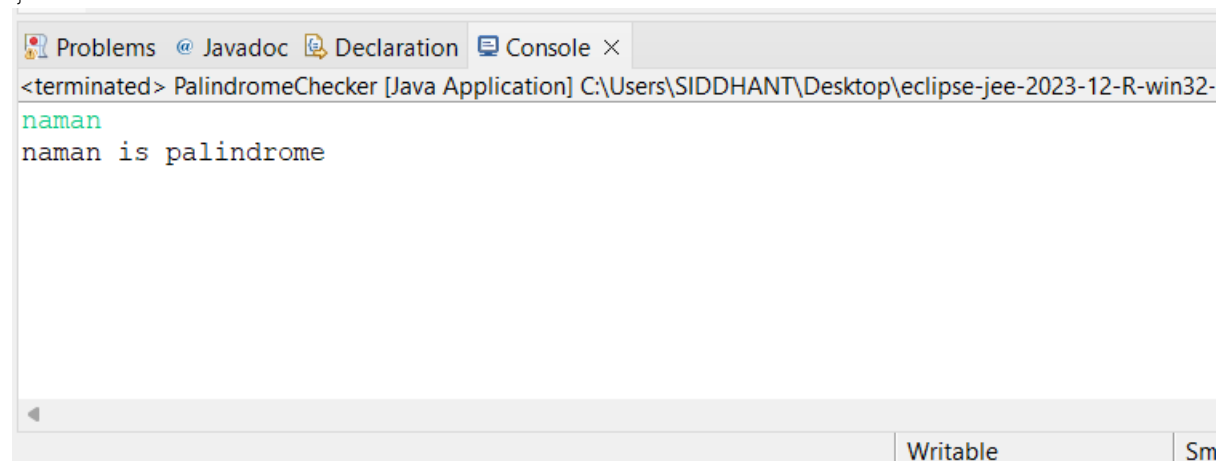
## 8. Recursive function to check if a string is palindrome

```
package com.example.main;

import java.util.Scanner;

class PalindromeChecker {
    public static boolean isPalindrome(String input) {
        if (input.length() == 0 || input.length() == 1) {
            return true;
        }
        if (input.charAt(0) != input.charAt(input.length() - 1)) {
            return false;
        }
        return isPalindrome(input.substring(1, input.length() - 1));
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        if (isPalindrome(str)) {
            System.out.println(str + " is palindrome ");
        } else {
            System.out.println(str + " is not palindrome ");
        }
    }
}
```



```
<terminated> PalindromeChecker [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-win32-
naman
naman is palindrome
```

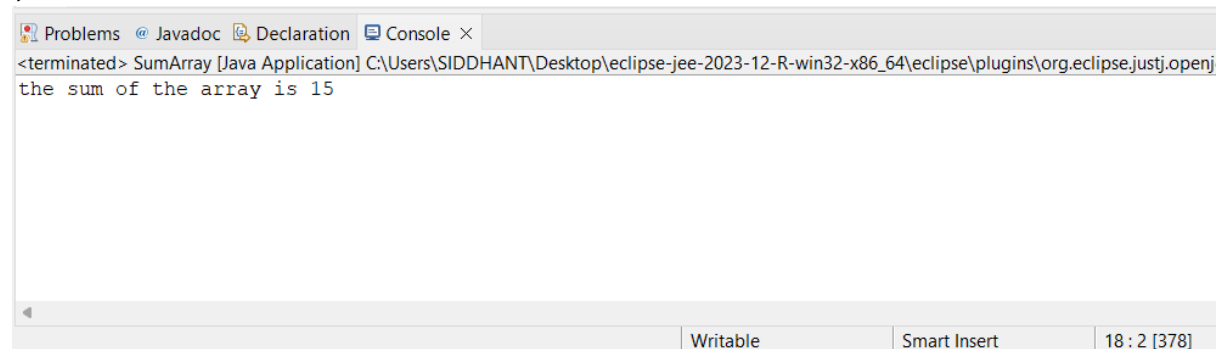
## 9. Tail recursion to calculate sum of array elements.

```
package com.example.main;

import java.util.Scanner;

class SumArray {
    public static int Sum(int[] arr, int n) {
        if (n == 0) {
            return 0;
        }
        int sSum = Sum(arr, n - 1);
        return sSum + arr[n - 1];
    }

    public static void main(String[] args) {
        int arr[] = { 1, 2, 3, 4, 5 };
        System.out.println("the sum of the array is " + Sum(arr,
arr.length));
    }
}
```



## 10. Print Fibonacci Series in reverse order using Recursion

```
package com.example.main;
```

```
import java.util.ArrayList;
```

```
import java.util.List;
```

```
import java.util.Scanner;
```

```
class PrintFibonacciReverse {
```

```
    public static List<Integer> list = new ArrayList<>();
```

```
    public static List<Integer> printFib(int n) {
```

```
        if (n == 1) {
```

```
            list.add(0);
```

```
            return list;
```

```
        }
```

```
        if (n == 2) {
```

```
            list.add(0);
```

```
            list.add(1);
```

```
            return list;
```

```
        }
```

```
        List<Integer> result = printFib(n - 1);
```

```
        int fLast = result.get(result.size() - 1);
```

```
        int sLast = result.get(result.size() - 2);
```

```
        int last = fLast + sLast;
```

```
        if (last < n) {
```

```
            result.add(last);
```

```
        }
```

```

        return result;
    }

    public static void printReverse(List<Integer> list) {
        if (list.size() == 0)
            return;

        Integer val = list.get(0);
        list.remove(val);
        printReverse(list);
        System.out.print(val + " ");
    }

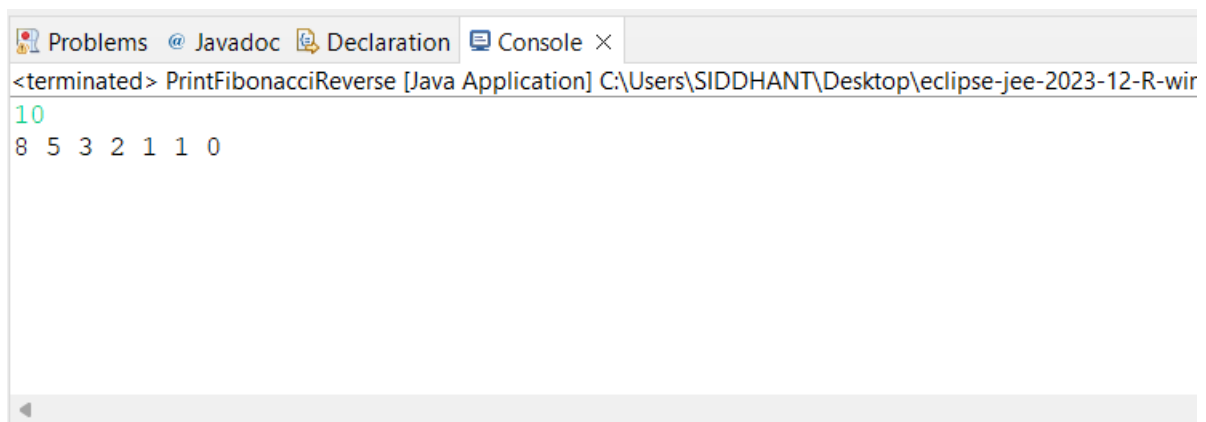
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        int n = sc.nextInt();

        List<Integer> res = printFib(n);

        printReverse(res);
    }
}

```



The screenshot shows the Eclipse IDE's console window. The title bar includes tabs for 'Problems', '@ Javadoc', 'Declaration', and 'Console'. The console text shows the application has terminated and printed the Fibonacci sequence in reverse order for n=10. The output is '10' on the first line and '8 5 3 2 1 1 0' on the second line.

```

<terminated> PrintFibonacciReverse [Java Application] C:\Users\SIDDHANT\Desktop\eclipse-jee-2023-12-R-wir
10
8 5 3 2 1 1 0

```