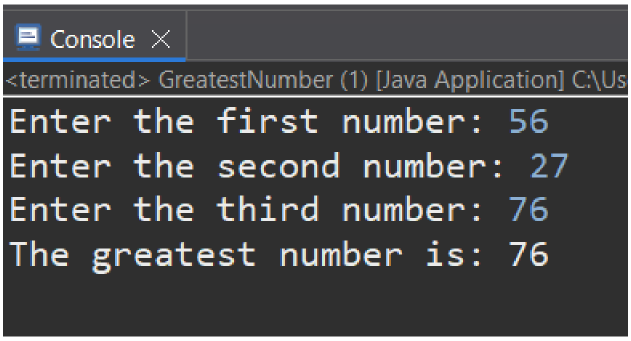
1) W.A.J.P to Take three numbers from the user and print the greatest number.

```
package Assignments;
import java.util.Scanner;
public class GreatestNumber {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the first number: ");
        int num1 = sc.nextInt();
        System.out.print("Enter the second number: ");
        int num2 = sc.nextInt();
        System.out.print("Enter the third number: ");
        int num3 = sc.nextInt();
        int greatest;
        if (num1 >= num2 && num1 >= num3) {
            greatest = num1;
        }
        else if (num2 >= num1 && num2 >= num3) {
            greatest = num2;
        }
        else {
            greatest = num3;
        }
        System.out.println("The greatest number is: " + greatest);
        sc.close();
        }
}
```

OUTPUT:



Console

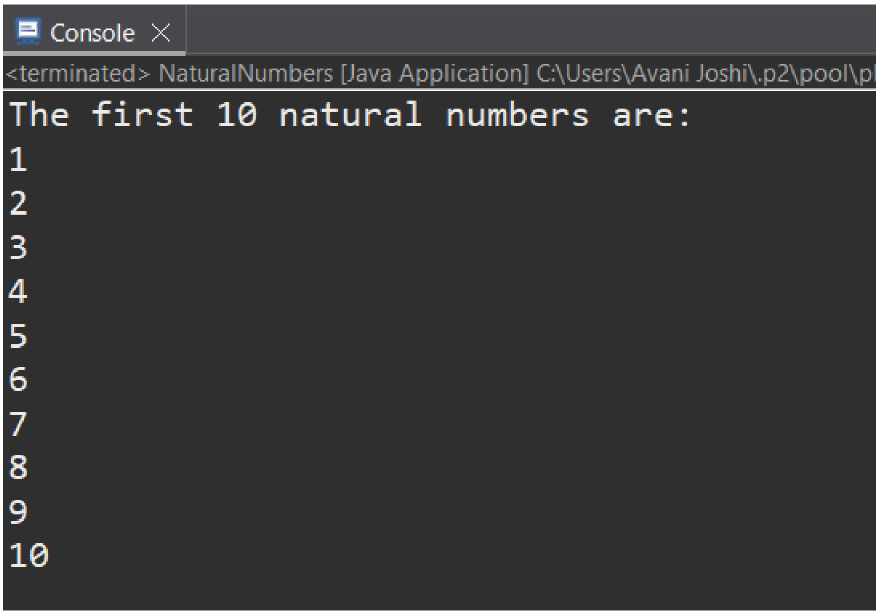
<terminated> GreatestNumber (1) [Java Application] C:\Us

```
Enter the first number: 56
Enter the second number: 27
Enter the third number: 76
The greatest number is: 76
```

2) W.A.J.P in Java to display the first 10 natural numbers using while loop.

```
package Assignments;
public class NaturalNumbers {
        public static void main(String[] args) {
                int i = 1;
                System.out.println("The first 10 natural numbers are:");
                while (i <= 10) {
                        System.out.println(i);
                        i++;
                }
        }
}
```

OUTPUT:

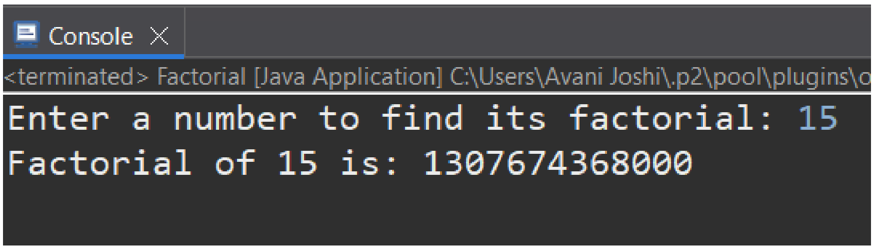


```
<terminated> NaturalNumbers [Java Application] C:\Users\Avani Joshi\.p2\pool\p
The first 10 natural numbers are:
1
2
3
4
5
6
7
8
9
10
```

3) W.A.J.P to find factorial for Given Number.

```
package Assignments;
import java.util.Scanner;
public class Factorial {
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter a number to find its factorial: ");
                int number = sc.nextInt();
                long factorial = 1;
                for (int i = 1; i <= number; i++) {
                        factorial *= i;
                }
                System.out.println("Factorial of " + number + " is: " + factorial);
                sc.close();
        }
}
```

OUTPUT:

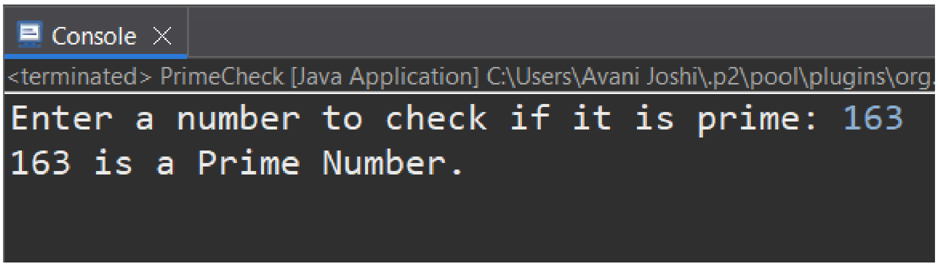


```
<terminated> Factorial [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\o
Enter a number to find its factorial: 15
Factorial of 15 is: 1307674368000
```

4) W.A.J.P to check given number is Prime or not?

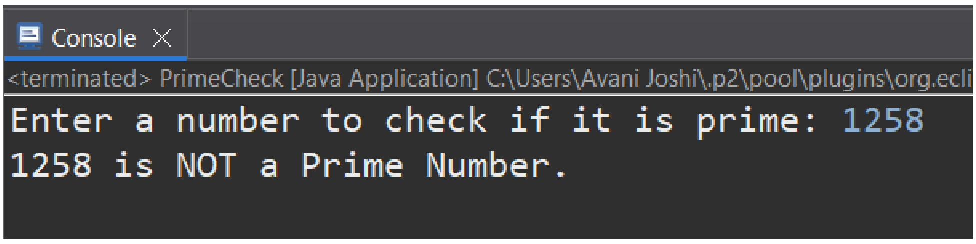
```
package Assignments;
import java.util.Scanner;
public class PrimeCheck {
        public static void main(String[] args) {
                Scanner sc = new Scanner (System.in);
                System.out.print("Enter a number to check if it is prime: ");
                int num = sc. nextInt ();
                int count = 0;
                for (int i=1; i<=num; i++) {
                        if (num%i == 0) {
                                count++;
                        }
                }
                if (count == 2) {
                        System.out.println(num + " is a Prime Number.");
                }
                else {
                        System.out.println(num + " is NOT a Prime Number.");
                }
                sc.close();
        }
}
```

OUTPUT:



<terminated> PrimeCheck [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.

```
Enter a number to check if it is prime: 163
163 is a Prime Number.
```



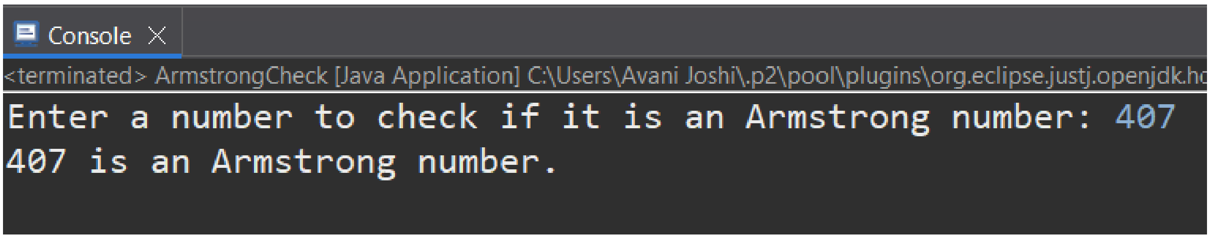
<terminated> PrimeCheck [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.ecli

```
Enter a number to check if it is prime: 1258
1258 is NOT a Prime Number.
```

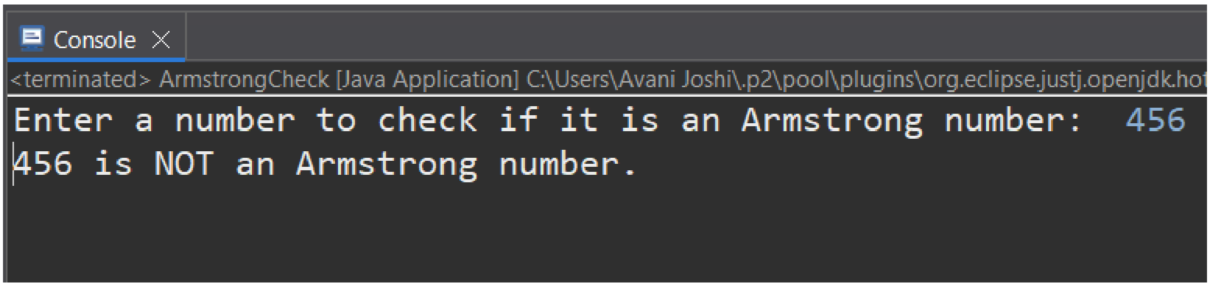
5) W.A.J.P to check given number is Armstrong or not?

```
package Assignments;
import java.util.Scanner;
public class ArmstrongCheck {
        public static void main(String[] args) {
                Scanner sc = new Scanner (System.in);
                System.out.print("Enter a number to check if it is an Armstrong number: ");
                int num = sc. nextInt ();
                int sum =0;
                int tmp = num;
                while (num>0) {
                    int digit = num%10;
                    sum = sum + (digit * digit * digit);
                    num = num/10;
                }
                if (sum == tmp) {
                    System.out.println(tmp + " is an Armstrong
number.");
                }
                else
                {
                    System.out.println(tmp + " is NOT an Armstrong
number.");
                }
                sc.close();
            }
}
```

OUTPUT:



```
<terminated> ArmstrongCheck [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hc
Enter a number to check if it is an Armstrong number: 407
407 is an Armstrong number.
```



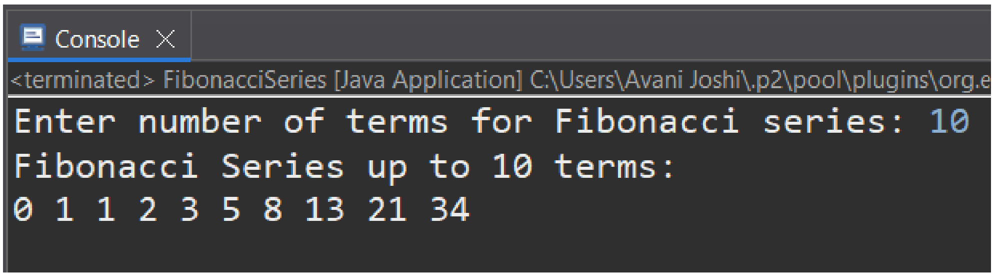
```
<terminated> ArmstrongCheck [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hot
Enter a number to check if it is an Armstrong number:  456
456 is NOT an Armstrong number.
```

6) W.A.J.P for create Fibonacci Series.

```
package Assignments;
import java.util.Scanner;
public class FibonacciSeries {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter number of terms for Fibonacci series: ");
		int n = sc.nextInt();
		int a = 0, b = 1;
		System.out.println("Fibonacci Series up to " + n + " terms:");
		for (int i = 1; i <= n; i++) {
			System.out.print(a + " ");
			int c = a + b;
			a = b;
			b = c;
		}
		sc.close();
	}
```

}

OUTPUT:



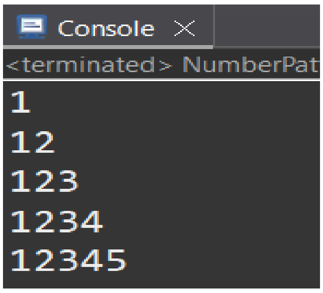
```
<terminated> FibonacciSeries [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.e
Enter number of terms for Fibonacci series: 10
Fibonacci Series up to 10 terms:
0 1 1 2 3 5 8 13 21 34
```

7) W.A.J.P to Print pattern Given Below.

i. Pattern-1

```
package Assignments;
public class NumberPattern1 {
	public static void main(String[] args) {
		for (int i = 1; i <= 5; i++) {
			for (int j = 1; j <= i; j++) {
				System.out.print(j);
			}
			System.out.println();
		}
	}
}
```

OUTPUT:

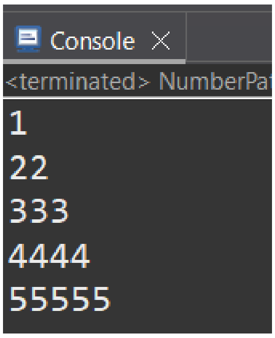


```
<terminated> NumberPat
1
12
123
1234
12345
```

ii. Pattern-2

```
package Assignments;
public class NumberPattern3 {
        public static void main(String[] args) {
                for (int i = 1; i <= 5; i++) {
                        for (int j = 1; j <= i; j++) {
                                System.out.print(i + "");
                        }
                        System.out.println();
                }
        }
}
```

OUTPUT:



Console

<terminated> NumberPa

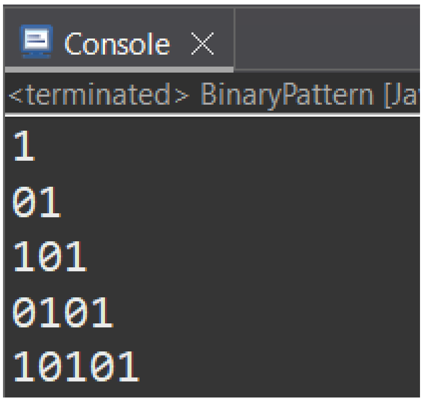
```
1
22
333
4444
55555
```

iii. Pattern-3

```
package Assignments;
public class BinaryPattern {
        public static void main(String[] args) {
                for (int i = 1; i <= 5; i++) {
                        for (int j = 1; j <= i; j++) {
```

```
                if ((i + j) % 2 == 0) {
                    System.out.print("1");
                }
                else {
                    System.out.print("0");
                }
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

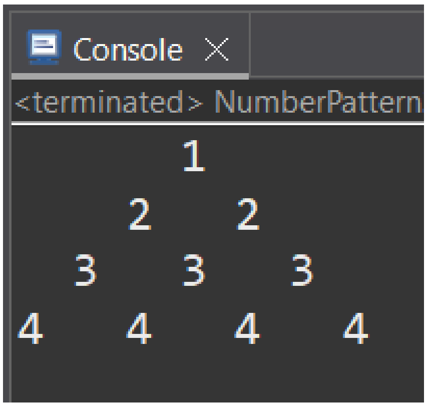


```
1
01
101
0101
10101
```

iv. Pattern-4

```
package Assignments;
public class NumberPattern2 {
    public static void main(String[] args) {
        for (int i = 1; i <= 4; i++) {
            for (int space = 1; space <= 4 - i; space++) {
                System.out.print("  ");
            }
            for (int j = 1; j <= i; j++) {
                System.out.print(i + "   ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

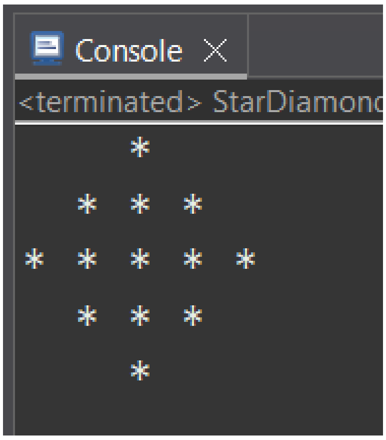


```
<terminated> NumberPattern
      1
    2   2
  3   3   3
4   4   4   4
```

v. Pattern-5

```
package Assignments;
public class StarDiamondPattern {
    public static void main(String[] args) {
        for (int i = 1; i <= 3; i++) {
            for (int space = 1; space <= 3 - i; space++) {
                System.out.print("  ");
            }
            for (int star = 1; star <= (2 * i - 1); star++) {
                System.out.print("* ");
            }
            System.out.println();
        }
        for (int i = 3 - 1; i >= 1; i--) {
            for (int space = 1; space <= 3 - i; space++) {
                System.out.print("  ");
            }
            for (int star = 1; star <= (2 * i - 1); star++) {
                System.out.print("* ");
            }
            System.out.println();
        }
    }
}
```

OUTPUT:

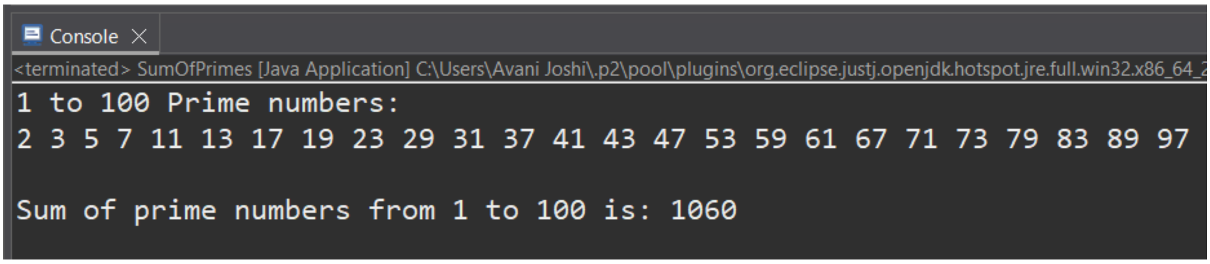


```
<terminated> StarDiamonc
    *
  * * *
* * * * *
  * * *
    *
```

8) WAP to compute the sum of the first 100 prime numbers.

```
package Assignments;
public class SumOfPrimes {
	public static void main(String[] args) {
		int sum = 0;
		System.out.println("1 to 100 Prime numbers: ");
		for (int num = 2; num<=100; num++) {
			int count =0;
			for (int i=1; i<=num; i++) {
				if (num % i == 0) {
					count++;
				}
			}
			if (count == 2) {
				System.out.print(num + " ");
				sum = sum + num;
			}
		}
		System.out.println("\n\nSum of prime numbers from 1 to 100 is: " + sum);
	}
}
```

OUTPUT:

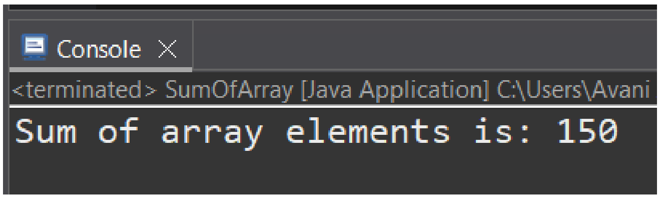


```
<terminated> SumOfPrimes [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_2
1 to 100 Prime numbers:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

Sum of prime numbers from 1 to 100 is: 1060
```

9) WAP to sum values of an array.

```
package Assignments;
public class SumOfArray {
	public static void main(String[] args) {
		int[] a = {10, 20, 30, 40, 50};
		int sum = 0;
		for (int i = 0; i < a.length; i++) {
			sum = sum + a[i];
		}
		System.out.println("Sum of array elements is: " + sum);
	}
}
```

OUTPUT:

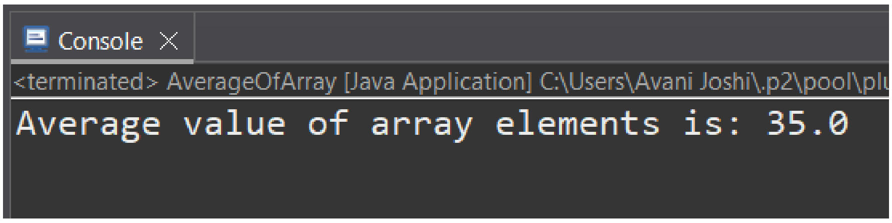


```
<terminated> SumOfArray [Java Application] C:\Users\Avani
Sum of array elements is: 150
```

10) WAP to calculate the average value of array elements.

```
package Assignments;
public class AverageOfArray {
	public static void main(String[] args) {
		int[] a = {10, 20, 30, 40, 50, 60};
		int sum = 0;
		for (int i = 0; i < a.length; i++) {
			sum = sum + a[i];
		}
		double avg = (double) sum / a.length;
		System.out.println("Average value of array elements is: " + avg);
	}
}
```

OUTPUT:



<terminated> AverageOfArray [Java Application] C:\Users\Avani Joshi\.p2\pool\pl

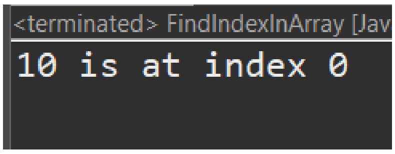
```
Average value of array elements is: 35.0
```

11) WAP to find the index of an array element.

```
package Assignments;
public class FindIndexInArray {
	public static void main(String[] args) {
		int a [] = {10, 20, 30, 40, 50};
		int index = 10;
```

```
        for (int i=0; i<a. length; i++) {
            if (a[i] == index) {
                System.out.println(index + " is at index " + i);
            }
        }
    }
}
```

OUTPUT:

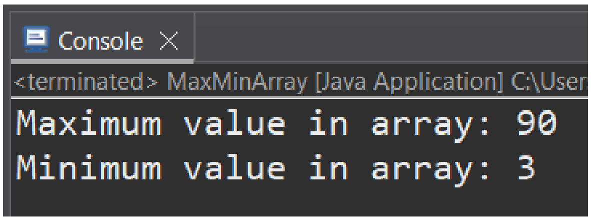


```
10 is at index 0
```

12) WAP to find the maximum and minimum value of an array.

```
package Assignments;
public class MaxMinArray {
    public static void main(String[] args) {
        int[] numbers = {25, 78, 12, 90, 3, 55};
        int max = numbers[0];
        int min = numbers[0];
        for (int i = 1; i < numbers.length; i++) {
            if (numbers[i] > max) {
                max = numbers[i];
            }
            if (numbers[i] < min) {
                min = numbers[i];
            }
        }
    System.out.println("Maximum value in array: " + max);
    System.out.println("Minimum value in array: " + min);
    }
}
```

OUTPUT:



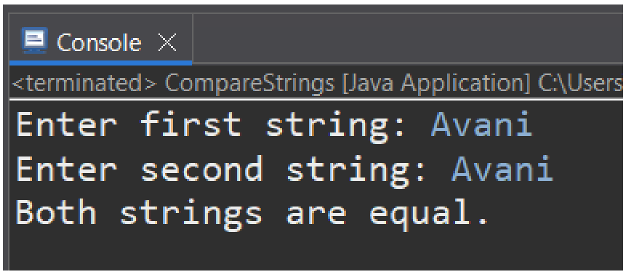
```
Maximum value in array: 90
Minimum value in array: 3
```

13) WAP to Compare Two String.

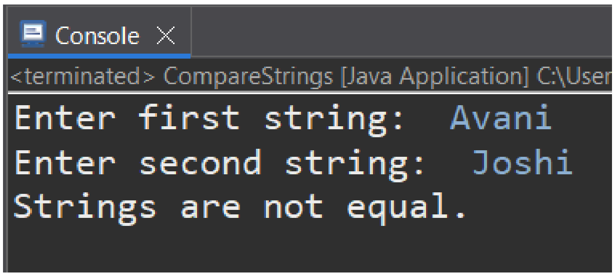
```
package Assignments;
import java.util.Scanner;
public class CompareStrings {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter first string: ");
        String str1 = sc.nextLine();

        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();
        if (str1.equals(str2)) {
            System.out.println("Both strings are equal.");
        }
```

```
		else {
			System.out.println("Strings are not equal.");
		}
		sc.close();
	}
}
```

OUTPUT:



```
<terminated> CompareStrings [Java Application] C:\Users
Enter first string: Avani
Enter second string: Avani
Both strings are equal.
```



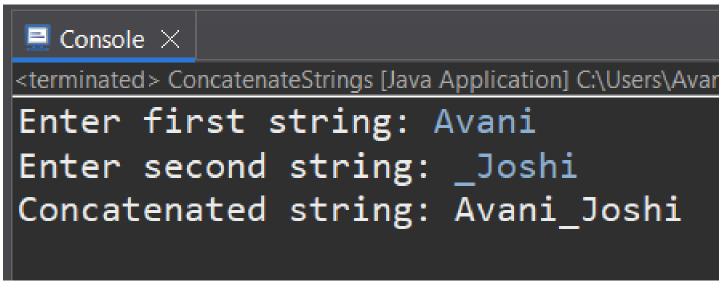
```
<terminated> CompareStrings [Java Application] C:\User
Enter first string:  Avani
Enter second string:  Joshi
Strings are not equal.
```

14) WAP to concatenate a given string to the end of another string.

```
package Assignments;
import java.util.Scanner;
public class ConcatenateStrings {
	public static void main(String[] args) {
		Scanner sc = new Scanner(System.in);
		System.out.print("Enter first string: ");
		String str1 = sc.nextLine();
```

```
        System.out.print("Enter second string: ");
        String str2 = sc.nextLine();
        String result = str1 + str2;
        System.out.println("Concatenated string: " + result);
        sc.close();
    }
}
```

OUTPUT:



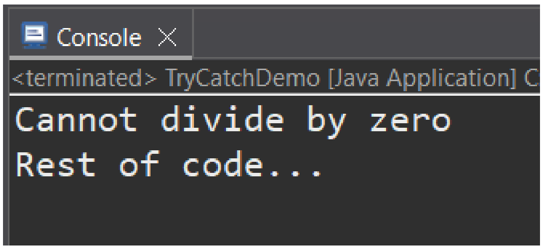
```
<terminated> ConcatenateStrings [Java Application] C:\Users\Avar
Enter first string: Avani
Enter second string: _Joshi
Concatenated string: Avani_Joshi
```

15) WAP to demonstrate try catch block.

```
package Assignments;
public class TryCatchDemo {
    public static void main(String[] args) {
        try {
            int a = 10;
            int b= 0;
            int res = a/b;
            System.out.println("Result is: " +res);
        }
        catch (ArithmeticException e) {
            System.out.println("Cannot divide by zero");
```

```
                }
                System.out.println("Rest of code...");
        }
}
```

OUTPUT:

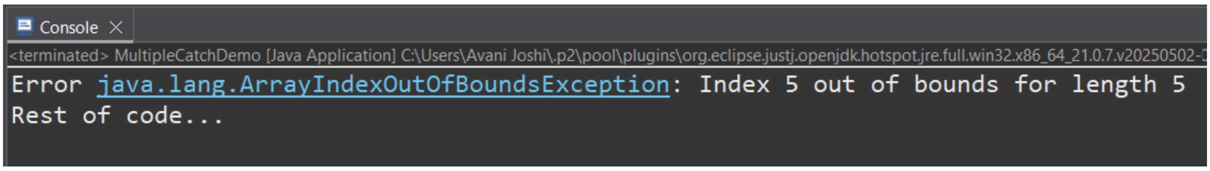


```
Cannot divide by zero
Rest of code...
```

16) WAP to demonstrate multiple catch blocks.

```
package Assignments;
public class MultipleCatchDemo {
        public static void main(String[] args) {
                int arr [] = new int [5];
                try {
                        arr [5] = 10;
                        System.out.println("Hello ");
                }
                catch (ArithmeticException e) {
                        System.out.println("Error " + e);
                }
                catch (Exception e) {
                        System.out.println("Error " + e);
                }
        System.out.println("Rest of code...");
    }
}
```

OUTPUT:

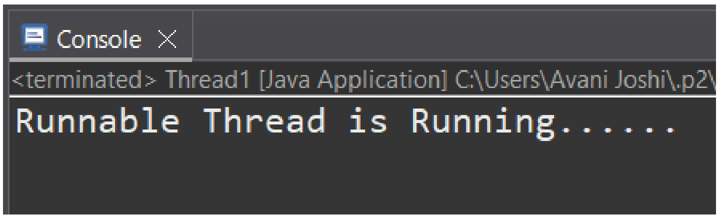


```
<terminated> MultipleCatchDemo [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-0
Error java.lang.ArrayIndexOutOfBoundsException: Index 5 out of bounds for length 5
Rest of code...
```

17) WAP to create one thread by implementing Runnable interface in Class.

```
package Assignments;
public class Thread1 implements Runnable {
    public void run () {
        System.out.println("Runnable Thread is Running......");
    }
    public static void main(String[] args) {
        Thread1 t1 = new Thread1();
        Thread m1 = new Thread(t1);
        m1. start ();
    }
}
```

OUTPUT:

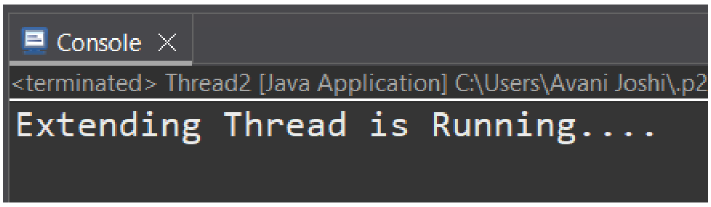


```
<terminated> Thread1 [Java Application] C:\Users\Avani Joshi\.p2\
Runnable Thread is Running......
```

18) WAP to create one thread by extending Thread class in another Class.

```
package Assignments;
public class Thread2 extends Thread {
        public void run () {
                System.out.println("Extending Thread is Running....");
        }
        public static void main(String[] args) {
                Thread2 t2 = new Thread2();
                t2. start ();
        }
}
```

OUTPUT:

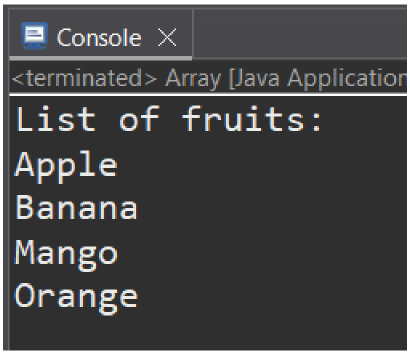


```
<terminated> Thread2 [Java Application] C:\Users\Avani Joshi\.p2
Extending Thread is Running....
```

19) WAP to iterate through all elements in an array list.

```
package Assignments;
import java.util.ArrayList;
public class Array {
        public static void main(String[] args) {
                ArrayList<String> fruits = new ArrayList<>();
                fruits.add("Apple");
                fruits.add("Banana");
                fruits.add("Mango");
                fruits.add("Orange");
                System.out.println("List of fruits:");
                for (String s:fruits) {
                    System.out.println(s);
                }
            }
}
```

OUTPUT:

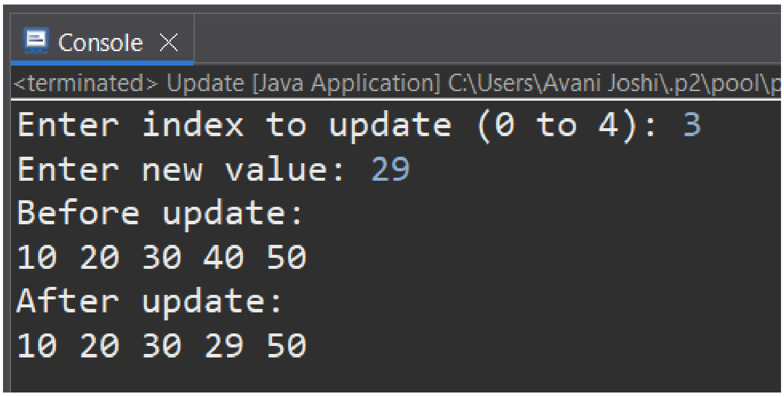


```
List of fruits:
Apple
Banana
Mango
Orange
```

20) WAP to update specific array element by given element.

```
package Assignments;
import java.util.Scanner;
public class Update {
    public static void main(String[] args) {
        int[] numbers = {10, 20, 30, 40, 50};
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter index to update (0 to 4): ");
        int index = sc.nextInt();
        if(index < 0 || index >= numbers.length) {
                System.out.println("Invalid index!");
            }
            else {
                System.out.print("Enter new value: ");
                int newValue = sc.nextInt();
                System.out.println("Before update:");
                for (int num : numbers) {
                    System.out.print(num + " ");
                }
                numbers[index] = newValue;
                System.out.println("\n After update: ");
                for (int num : numbers) {
                    System.out.print(num +  " ");
                }
            }
            sc.close();
        }
}
```

OUTPUT:

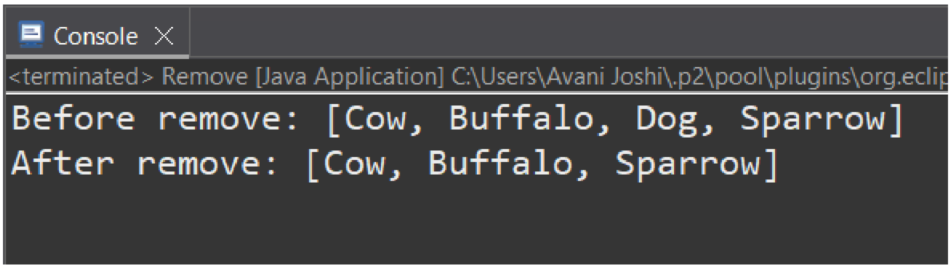


```
<terminated> Update [Java Application] C:\Users\Avani Joshi\.p2\pool\p
Enter index to update (0 to 4): 3
Enter new value: 29
Before update:
10 20 30 40 50
After update:
10 20 30 29 50
```

21) WAP to remove the third element from an array list.

```
package Assignments;
import java.util.ArrayList;
public class Remove {
	public static void main(String[] args) {
		ArrayList<String> st = new ArrayList<String>();
		st.add("Cow");
		st.add("Buffalo");
		st.add("Dog");
		st.add("Sparrow");
		System.out.println("Before remove: " + st);
		if (st.size() >= 3) {
			st.remove(2);
			System.out.println("After remove: " + st);
		}
		else {
			System.out.println("List has less than 3 elements.");
		}
	}
}
```

OUTPUT:

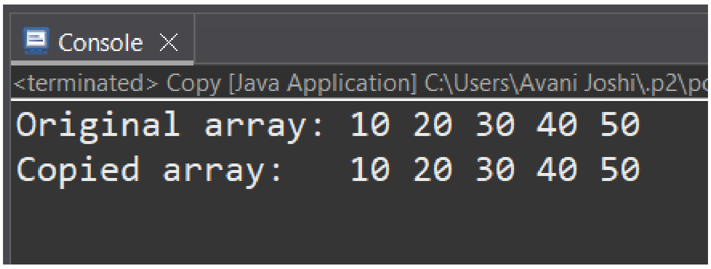


```
<terminated> Remove [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclip
Before remove: [Cow, Buffalo, Dog, Sparrow]
After remove: [Cow, Buffalo, Sparrow]
```

22) WAP to Copy one array into another.

```
package Assignments;
public class Copy {
	public static void main(String[] args) {
		int[] a = {10, 20, 30, 40, 50};
		int[] b = new int[a.length];
		for (int i = 0; i < a.length; i++) {
			b[i] = a[i];
		}
		System.out.print("Original array: ");
		for (int num : a) {
			System.out.print(num + " ");
		}
		System.out.print("\nCopied array:   ");
		for (int num : b) {
			System.out.print(num + " ");
		}
	}
}
```

OUTPUT:

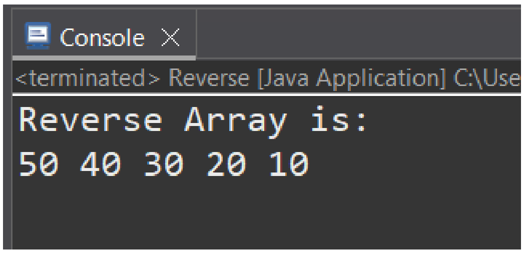


```
<terminated> Copy [Java Application] C:\Users\Avani Joshi\.p2\pc
Original array: 10 20 30 40 50
Copied array:   10 20 30 40 50
```

23) WAP to reverse an array of integer values.

```
package Assignments;
public class Reverse {
	public static void main(String[] args) {
		int a [] = {10,20,30,40,50};
		System.out.println("Reverse Array is: ");
		for (int i = 4; i>=0; i--) {
			System.out.print(a[i] + " ");
		}
	}
}
```

OUTPUT:



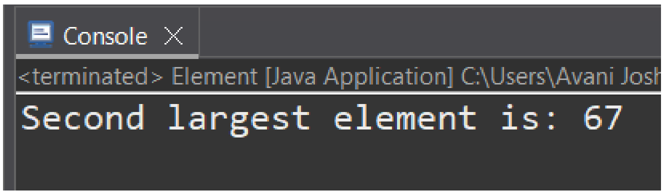
```
<terminated> Reverse [Java Application] C:\Use
Reverse Array is:
50 40 30 20 10
```

24) WAP to find the second largest element in an array.

```
package Assignments;
public class Element {
	public static void main(String[] args) {
		int [] arr = {45, 22, 89, 65, 12};
		int largest = arr [0];
		int second = arr [0];
		for (int i=1; i<arr. length; i++) {
```

```
                if(arr[i] > largest) {
                        largest = arr[i];
                }
        }
        for (int i=0; i<arr.length; i++) {
                if(arr[i] > second && arr[i] < largest) {
                        second = arr[i];
                }
        }
        System.out.println("Second Largest Number is: "+ second);
    }
}
```

OUTPUT:

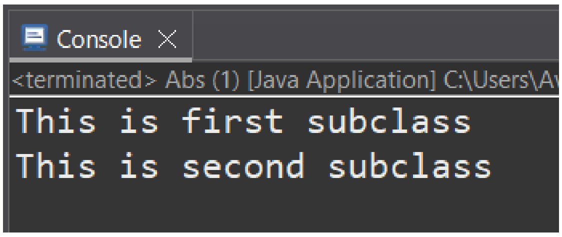


```
<terminated> Element [Java Application] C:\Users\Avani Josh
Second largest element is: 67
```

25) W.A.J.P. Create an abstract class 'Parent' with a method 'message'. It has two subclasses each having a method with the same name 'message' that prints "This is first subclass" and "This is second subclass" respectively. Call the methods 'message' by creating an object for each subclass.

```
package Assignments;
abstract class Parent {
    abstract void message();
}
class Firstclass extends Parent {
        void message() {
                System.out.println("This is first subclass");
        }
}
class Secondclass extends Parent {
        void message() {
                System.out.println("This is second subclass");
        }
}
public class Abs {
        public static void main(String[] args) {
                Parent p1 = new Firstclass();
                Parent p2 = new Secondclass();
                p1.message();
                p2.message();
        }
}
```

OUTPUT:



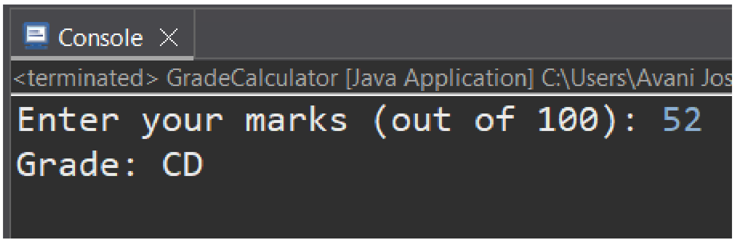
```
<terminated> Abs (1) [Java Application] C:\Users\A
This is first subclass
This is second subclass
```

26) W.A.J.P. which will ask the user to enter his/her marks (out of 100). Define a method that will display grades according to the marks entered as below:

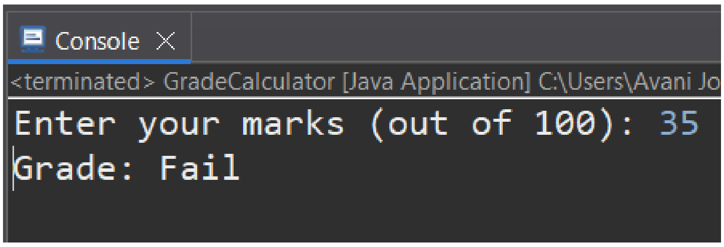
Marks	**Grade**
91-100	**AA**
81-90	**AB**
71-80	**BB**
61-70	**BC**
51-60	**CD**
41-50	**DD**
< = 40	**Fail**

```
package Assignments;
import java.util.Scanner;
public class GradeCalculator {
        static void displayGrade(int marks) {
                if (marks >= 91 && marks <= 100) {
                        System.out.println("Grade: AA");
                }
                else if (marks >= 81 && marks <= 90) {
                        System.out.println("Grade: AB");
                }
                else if (marks >= 71 && marks <= 80) {
                        System.out.println("Grade: BB");
                }
                else if (marks >= 61 && marks <= 70) {
                        System.out.println("Grade: BC");
                }
                else if (marks >= 51 && marks <= 60) {
                        System.out.println("Grade: CD");
                }
                else if (marks >= 41 && marks <= 50) {
                        System.out.println("Grade: DD");
                }
                else if (marks <= 40 && marks >= 0) {
                        System.out.println("Grade: Fail");
                }
                else {
                        System.out.println("Invalid marks! Please enter
marks between 0 and 100.");
                }
        }
        public static void main(String[] args) {
                Scanner sc = new Scanner(System.in);
                System.out.print("Enter your marks (out of 100): ");
                int marks = sc.nextInt();
                displayGrade(marks);
                sc.close();
        }
}
```

OUTPUT:



```
<terminated> GradeCalculator [Java Application] C:\Users\Avani Jos
Enter your marks (out of 100): 52
Grade: CD
```



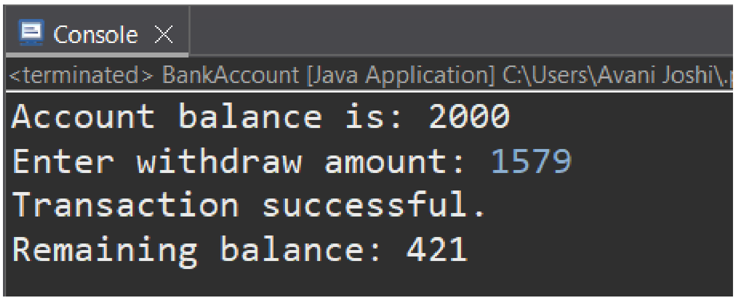
```
<terminated> GradeCalculator [Java Application] C:\Users\Avani Jo
Enter your marks (out of 100): 35
Grade: Fail
```

27) W.A.J.P. to create a custom exception if Customer withdraw amount which is greater than account balance then program will show custom exception otherwise amount will deduct from account balance. Account balance is:2000 Enter withdraw amount:2500 Sorry, insufficient balance, you need more 500 Rs. To perform this transaction.

```
package Assignments;
import java.util.Scanner;
class InsufficientBalanceException extends Exception {
        InsufficientBalanceException(int extraAmount) {
                super("Sorry, insufficient balance, you need more " + extraAmount + " Rs. to perform this transaction.");
        }
}
public class BankAccount {
        public static void main(String[] args) {
```

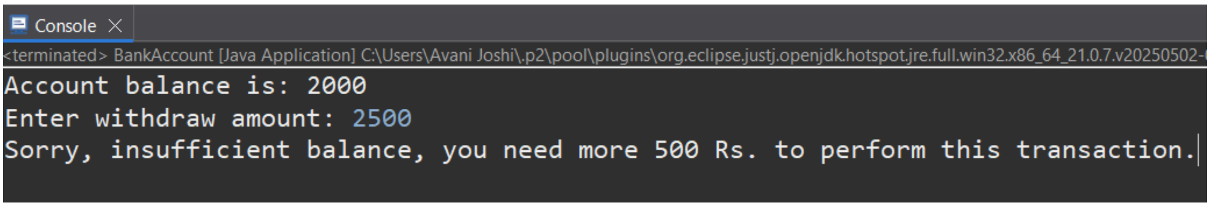
```
        int balance = 2000;
        Scanner sc = new Scanner(System.in);
        System.out.println("Account balance is: " + balance);
        System.out.print("Enter withdraw amount: ");
        int withdrawAmount = sc.nextInt();
        try {
            if (withdrawAmount > balance) {
                int shortage = withdrawAmount - balance;
            throw new InsufficientBalanceException(shortage);
            }
            else {
                balance = balance - withdrawAmount;
                System.out.println("Transaction successful.");
                System.out.println("Remaining balance: " +
balance);
            }
        }
        catch (InsufficientBalanceException e) {
            System.out.println(e.getMessage());
        }
        sc.close();
    }
}
```

OUTPUT:



Console
<terminated> BankAccount [Java Application] C:\Users\Avani Joshi\.

```
Account balance is: 2000
Enter withdraw amount: 1579
Transaction successful.
Remaining balance: 421
```



Console
<terminated> BankAccount [Java Application] C:\Users\Avani Joshi\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_21.0.7.v20250502-

```
Account balance is: 2000
Enter withdraw amount: 2500
Sorry, insufficient balance, you need more 500 Rs. to perform this transaction.
```