RAJAGIRI COLLEGE OF SOCIAL SCIENCES (AUTONOMOUS),KALAMASSERY

SEMESTER 2

JAVA RECORD

POORNIMA K R

MSc CS (DA)

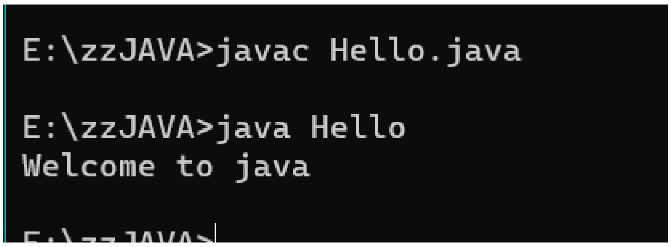
Section 1:

1. Write a program to print ‘Welcome to Java’.

   Code:

```
import java.io.*;
import java.lang.*;
class Hello{
        public static void main(String args[]){
                System.out.println("Welcome to java");
}}
```

   Output:



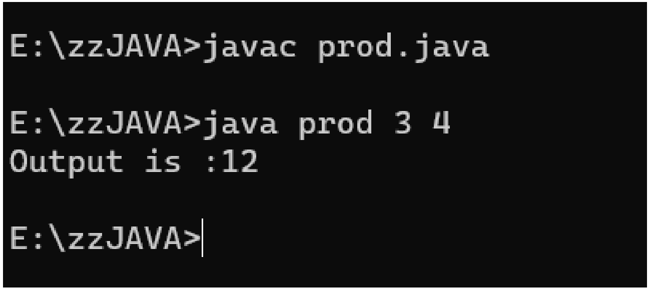
```
E:\zzJAVA>javac Hello.java

E:\zzJAVA>java Hello
Welcome to java

E:\zzJAVA>
```

2. WAP to display two numbers received as command line argument, and print its product

   Code:

```
        import java.io.*;
        import java.lang.*;
        class prod{
                public static void main(String args[]){
                        int a=Integer.parseInt(args[0]);
                        int b=Integer.parseInt(args[1]);
                        int s=a*b;
                        System.out.println("Output is :"+s);
}
}
```

   Output:

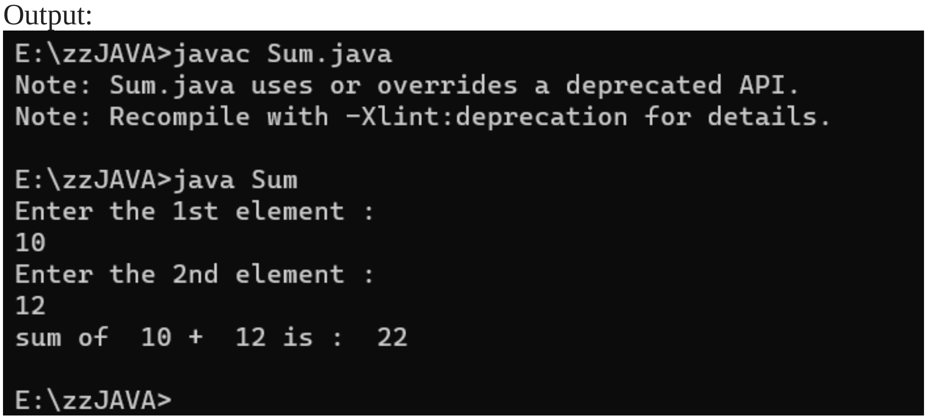


```
E:\zzJAVA>javac prod.java

E:\zzJAVA>java prod 3 4
Output is :12

E:\zzJAVA>
```

3. WAP to read two numbers and display the output in the form of ‘Sum of 2 and 3 is 5
   Code:

```
import java.io.*;
import java.lang.*;

class Sum{

        public static void main(String args[]){

                try{

                        int a,b,c;

                        DataInputStream dir=new
                DataInputStream(System.in);

                        System.out.println("Enter the 1st element :");

                        a=Integer.parseInt(dir.readLine());

                        System.out.println("Enter the 2nd element :");

                        b=Integer.parseInt(dir.readLine());

                        c=a+b;

                        System.out.println("sum of  " + a + " +  " + b
                + " is :  " + c );

}

        catch(Exception e){

                System.out.println("error"+e);

                }

        }

}
```

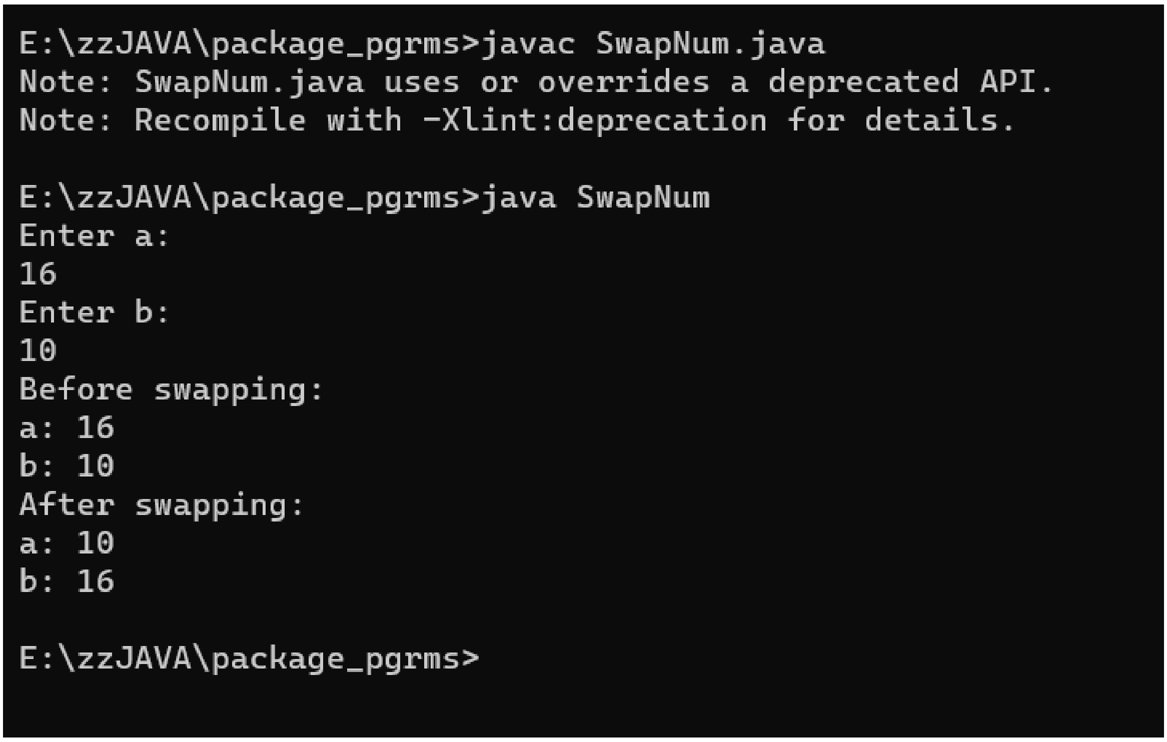
Output:



```
E:\zzJAVA>javac Sum.java
Note: Sum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java Sum
Enter the 1st element :
10
Enter the 2nd element :
12
sum of  10 +  12 is :  22

E:\zzJAVA>
```

4. WAP to accept two numbers from the keyboard and swap them.

```
import java.io.*;
public class SwapNum
{
public static void main(String args[])
{
try
{
String s;

System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int a=Integer.parseInt(s);

System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);

System.out.println("Before swapping: ");
System.out.println("a: "+a);
System.out.println("b: "+b);

int t=a;
a=b;
b=t;

System.out.println("After swapping: ");
System.out.println("a: "+a);
System.out.println("b: "+b);

}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



```
E:\zzJAVA\package_pgrms>javac SwapNum.java
Note: SwapNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA\package_pgrms>java SwapNum
Enter a:
16
Enter b:
10
Before swapping:
a: 16
b: 10
After swapping:
a: 10
b: 16

E:\zzJAVA\package_pgrms>
```

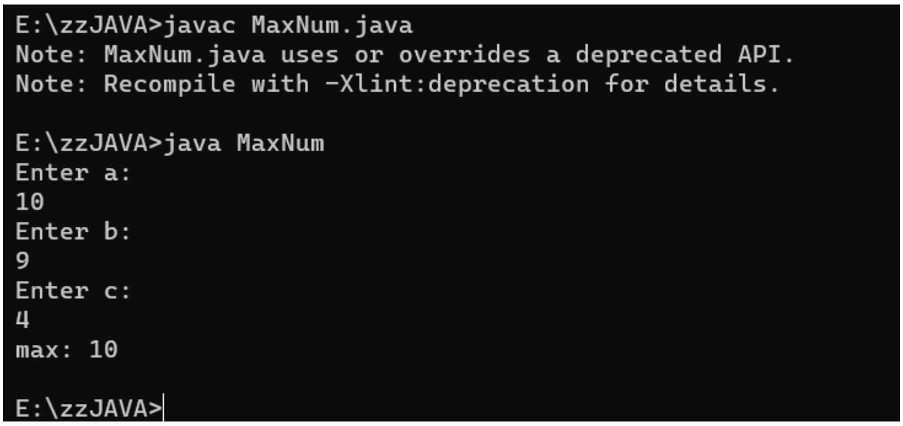
SECTION 2

1.WAP to read three numbers and the maximum.

```
import java.io.*;
import java.lang.*;

public class MaxNum
{
public static void main(String args[])
{
try
{
String s;
```

```
System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int a=Integer.parseInt(s);

System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);

System.out.println("Enter c: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int c=Integer.parseInt(s);

int max;
if(a>=b&&a>=c)
{
System.out.println("max: "+a);
}
else if(b>=a&&b>=c)
{
System.out.println("max: "+b);
}
else
{
System.out.println("max: "+c);
}
}
catch(Exception e)
{
```

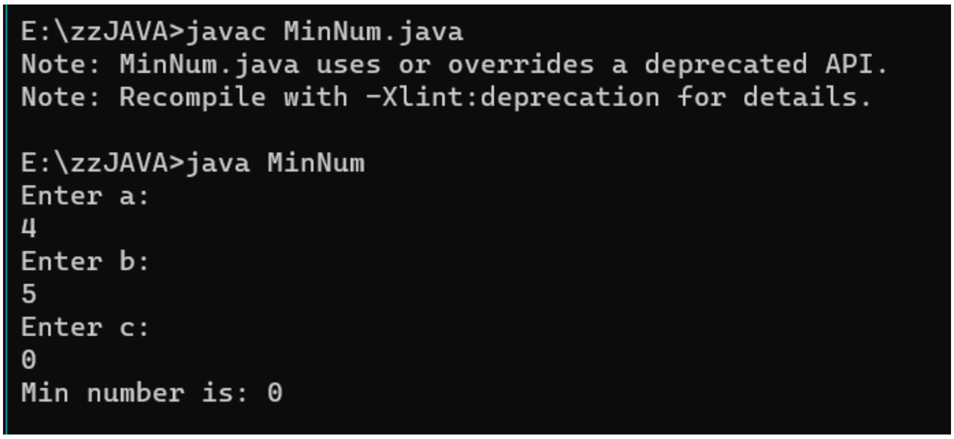
```
System.out.println(e);
}
}
}
```



```
E:\zzJAVA>javac MaxNum.java
Note: MaxNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java MaxNum
Enter a:
10
Enter b:
9
Enter c:
4
max: 10

E:\zzJAVA>
```

2. Find the minimum of three numbers using a single statement.

```
import java.io.*;
import java.lang.*;

public class MinNum
{
public static void main(String args[])
{
try
{
String s;

System.out.println("Enter a: ");
DataInputStream dis=new DataInputStream(System.in);
s=dis.readLine();
int a=Integer.parseInt(s);

System.out.println("Enter b: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int b=Integer.parseInt(s);
System.out.println("Enter c: ");
dis=new DataInputStream(System.in);
s=dis.readLine();
int c=Integer.parseInt(s);

int min=Math.min(Math.min(a,b),c);
//int min=(a<b) ? ((a<c)?a:c) : ((b<c)?b:c);
System.out.println("Min number is: "+min);

}
catch(Exception e)
{
System.out.println(e);
}
}
}
```



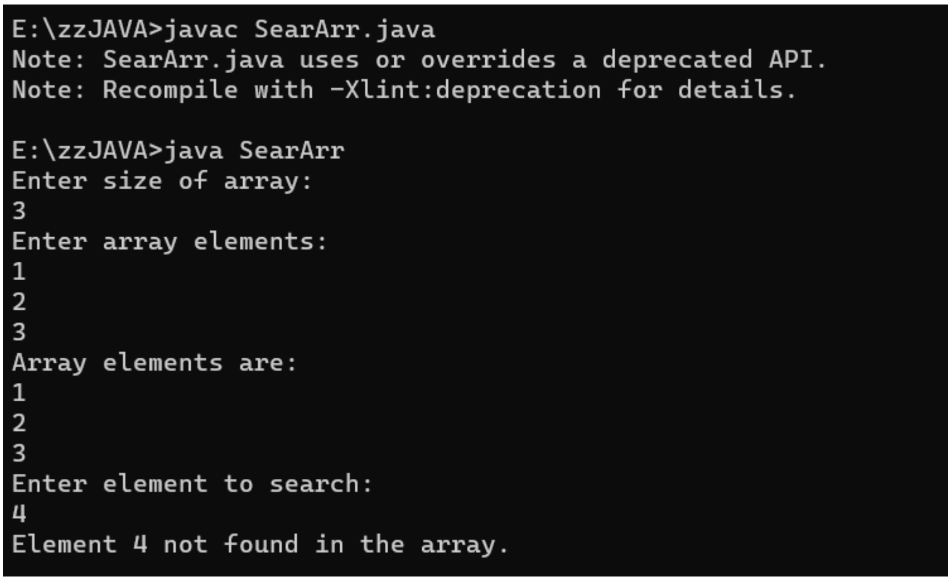
```
E:\zzJAVA>javac MinNum.java
Note: MinNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java MinNum
Enter a:
4
Enter b:
5
Enter c:
0
Min number is: 0
```

3.WAP to search for a given element in an array.

```
import java.io.*;

public class SearArr {
    public static void main(String args[]) {
        try {
            int a[];
```

```
String s;
int i;

System.out.println("Enter size of array: ");
DataInputStream dis = new DataInputStream(System.in);
s = dis.readLine();
int n = Integer.parseInt(s);
a = new int[n];

System.out.println("Enter array elements: ");
for (i = 0; i < n; i++) {
    s = dis.readLine();
    a[i] = Integer.parseInt(s);
}

System.out.println("Array elements are: ");
for (i = 0; i < n; i++) {
    System.out.println(a[i]);
}

System.out.println("Enter element to search: ");
s = dis.readLine();
int searchElement = Integer.parseInt(s);

int flag = 0;
for (i = 0; i < n; i++) {
    if (a[i] == searchElement) {
     flag = 1;
     System.out.println("Element " + searchElement + " found at index " + i);
        break;
    }
}
```

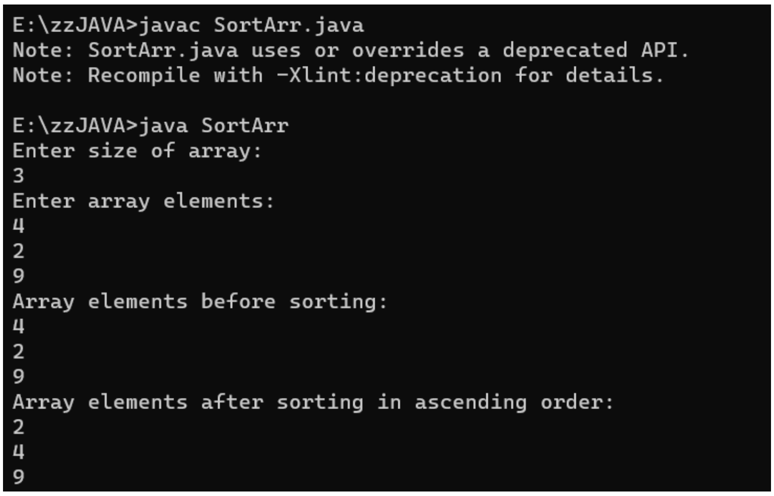
```
            if (flag == 0) {
            System.out.println("Element " + searchElement + " not found in the array.");
            }

        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac SearArr.java
Note: SearArr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SearArr
Enter size of array:
3
Enter array elements:
1
2
3
Array elements are:
1
2
3
Enter element to search:
4
Element 4 not found in the array.
```

4. WAP to sort elements in an array in ascending order.

```
import java.io.*;

public class SortArr {
    public static void main(String args[]) {
        try {
            int a[];
            String s;
            int i;

            System.out.println("Enter size of array: ");
            DataInputStream dis = new DataInputStream(System.in);
            s = dis.readLine();
            int n = Integer.parseInt(s);
            a = new int[n];

            System.out.println("Enter array elements: ");
            for (i = 0; i < n; i++) {
                s = dis.readLine();
                a[i] = Integer.parseInt(s);
            }

            System.out.println("Array elements before sorting: ");
            for (i = 0; i < n; i++) {
                System.out.println(a[i]);
            }

            // Call the bubbleSort method to sort the array
            bubbleSort(a);

            System.out.println("Array elements after sorting in ascending order:");
            for (i = 0; i < n; i++) {
                System.out.println(a[i]);
            }

        } catch (Exception e) {
            System.out.println(e);
        }
    }

    // Bubble Sort method
    static void bubbleSort(int[] arr) {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j] > arr[j + 1]) {
                    int temp = arr[j];
                    arr[j] = arr[j + 1];
                    arr[j + 1] = temp;
                }
            }
        }
    }
}
```



```
E:\zzJAVA>javac SortArr.java
Note: SortArr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SortArr
Enter size of array:
3
Enter array elements:
4
2
9
Array elements before sorting:
4
2
9
Array elements after sorting in ascending order:
2
4
9
```

5. Write a program to print the row wise and column wise sum of a 2D array. 1 2 3 | 6

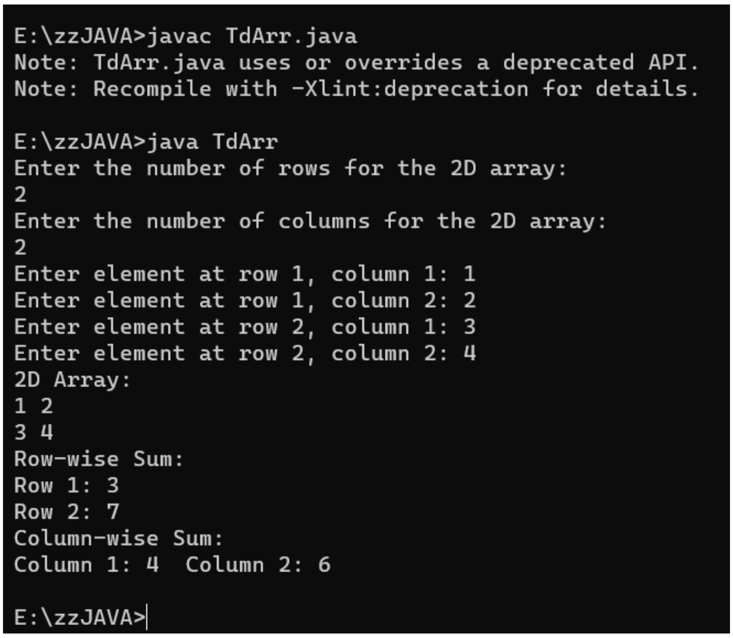
2 1 1 |4.   .   .

3 3 4

```
import java.io.*;


public class TdArr
{


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


        try {
```

```
int rows, cols;
int[][] array;
String s;

System.out.println("Enter the number of rows for the 2D array:");
s = i.readLine();
rows = Integer.parseInt(s);

System.out.println("Enter the number of columns for the 2D array:");
s = i.readLine();
cols = Integer.parseInt(s);

array = new int[rows][cols];

for (int j = 0; j < rows; j++) {
    for (int k = 0; k < cols; k++) {
        System.out.print("Enter element at row " + (j + 1) + ", column " + (k + 1) + ": ");
        s = i.readLine();
        array[j][k] = Integer.parseInt(s);
    }
}

System.out.println("2D Array:");
for (int j = 0; j < rows; j++) {
    for (int k = 0; k < cols; k++) {
        System.out.print(array[j][k] + " ");
    }
    System.out.println();
}

System.out.println("Row-wise Sum:");
for (int j = 0; j < rows; j++) {
```

```
                int rowSum = 0;
                for (int k = 0; k < cols; k++) {
                    rowSum += array[j][k];
                }
                System.out.println("Row " + (j + 1) + ": " + rowSum);
            }

            System.out.println("Column-wise Sum:");
            for (int k = 0; k < cols; k++) {
                int colSum = 0;
                for (int j = 0; j < rows; j++) {
                    colSum += array[j][k];
                }
                System.out.print("Column " + (k + 1) + ": " + colSum + "  ");
            }
            System.out.println();

        } catch (Exception e) {
            System.out.println("Error "+e);
        }
    }
}
```

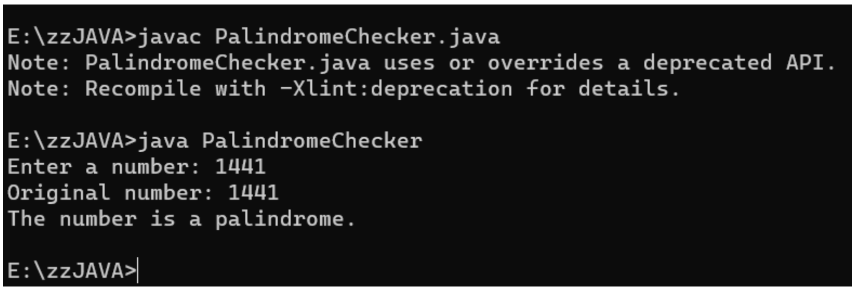


```
E:\zzJAVA>javac TdArr.java
Note: TdArr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java TdArr
Enter the number of rows for the 2D array:
2
Enter the number of columns for the 2D array:
2
Enter element at row 1, column 1: 1
Enter element at row 1, column 2: 2
Enter element at row 2, column 1: 3
Enter element at row 2, column 2: 4
2D Array:
1 2
3 4
Row-wise Sum:
Row 1: 3
Row 2: 7
Column-wise Sum:
Column 1: 4  Column 2: 6

E:\zzJAVA>
```

SECTION 3

1. WAP with two functions to check for an integer palindrome. (Function1 should reverse the integer. Function2 should return 1,if it is a palindrome or else 0.)

```
import java.io.*;
public class PalindromeChecker {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            String input = dis.readLine();
            int num = Integer.parseInt(input);
            System.out.println("Original number: " + num);
            if (isPalindrome(num)) {
                System.out.println("The number is a palindrome.");
            } else {
                System.out.println("The number is not a palindrome.");
            }
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static int reverseInteger(int num) {
        int reversed = 0;
        while (num != 0) {
            int digit = num % 10;
            reversed = reversed * 10 + digit;
            num /= 10;
        }
        return reversed;
    }
    public static boolean isPalindrome(int num) {
        return num == reverseInteger(num);
    }
}
```

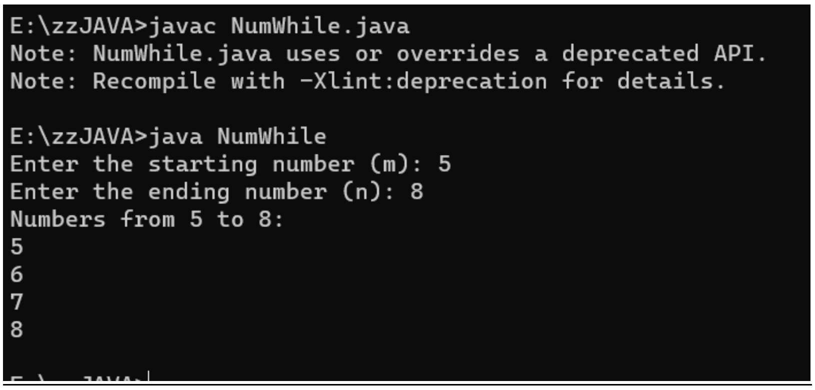


```
E:\zzJAVA>javac PalindromeChecker.java
Note: PalindromeChecker.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java PalindromeChecker
Enter a number: 1441
Original number: 1441
The number is a palindrome.

E:\zzJAVA>
```

2. WAP to display numbers from m to n using single while loop.

```
import java.io.*;
public class NumWhile {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the starting number (m): ");
            String mStr = dis.readLine();
            int m = Integer.parseInt(mStr);

            System.out.print("Enter the ending number (n): ");

            String nStr = dis.readLine();

            int n = Integer.parseInt(nStr);

            System.out.println("Numbers from " + m + " to " + n + ":");

            while (m <= n) {

                System.out.println(m);

                m++;

            }

        } catch (Exception e) {

            System.out.println(e);

        }

    }

}
```

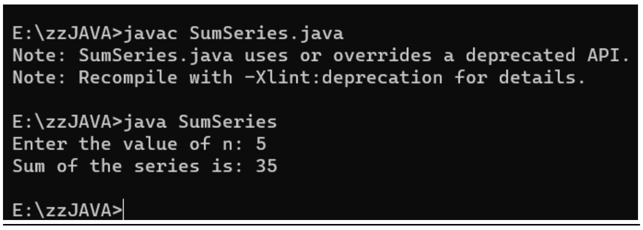
Output



```
E:\zzJAVA>javac NumWhile.java
Note: NumWhile.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java NumWhile
Enter the starting number (m): 5
Enter the ending number (n): 8
Numbers from 5 to 8:
5
6
7
8
```

2. WAP to find the sum of the series 1+(1+2)+(1+2+3)+............ +(1+2+3+…+n) using a single while loop.

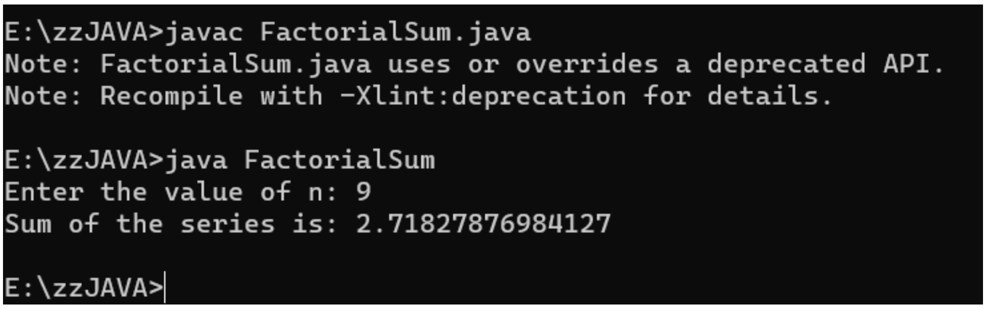
```
import java.io.*;

public class SumSeries {

    public static void main(String args[]) {

        try {

            DataInputStream dis = new DataInputStream(System.in);

            System.out.print("Enter the value of n: ");

            int n = Integer.parseInt(dis.readLine());

            int sum = 0;
            int innerSum = 0;
            int i = 1;
            while (i <= n) {
                innerSum += i;
                sum += innerSum;
                i++;
            }
            System.out.println("Sum of the series is: " + sum);
        } catch (Exception e) {
          System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac SumSeries.java
Note: SumSeries.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SumSeries
Enter the value of n: 5
Sum of the series is: 35

E:\zzJAVA>
```

4. WAP to find the sum of 1+2/2!+3/3!+4/4!++n/n! using a single for loop.

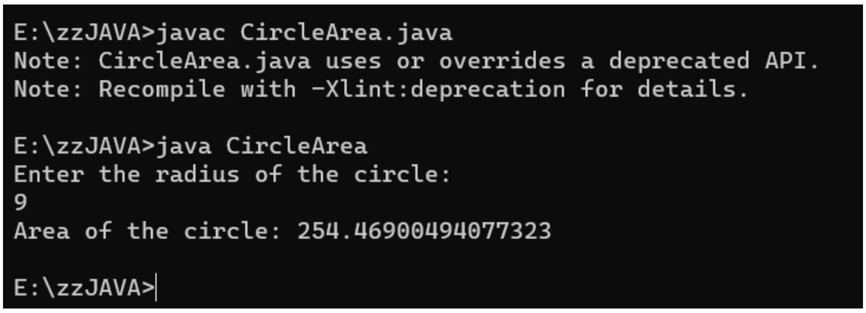
```
import java.io.*;
public class FactorialSum {
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the value of n: ");
            int n = Integer.parseInt(dis.readLine());
            double sum = 0;
            double factorial = 1;
            for (int i = 1; i <= n; i++) {
                factorial *= i;
                sum += i / factorial;
            }
            System.out.println("Sum of the series is: " + sum);
        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac FactorialSum.java
Note: FactorialSum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java FactorialSum
Enter the value of n: 9
Sum of the series is: 2.71827876984127

E:\zzJAVA>
```

5. WAP to calculate area of a circle (functions with no argument and no return type.)

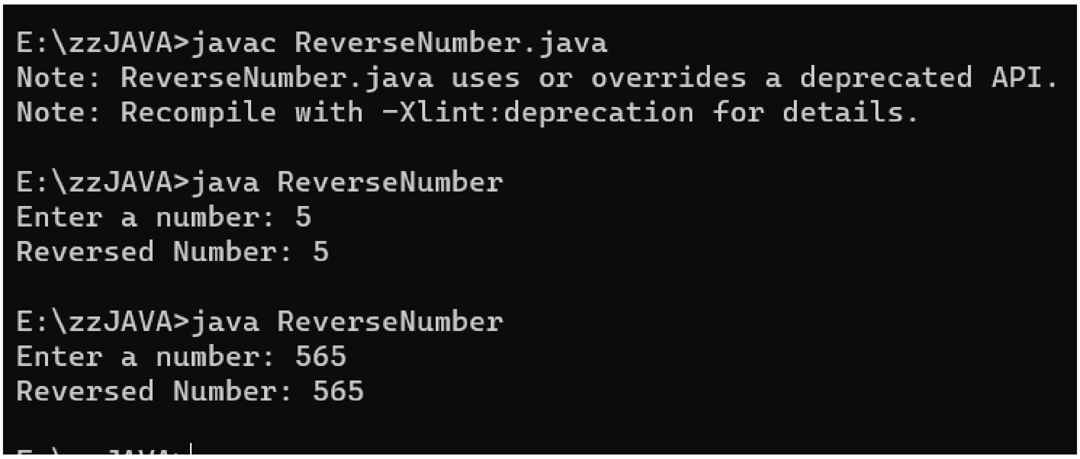
```
import java.io.*;
public class CircleArea {
    static void calculateArea() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.println("Enter the radius of the circle: ");
            double radius = Double.parseDouble(dis.readLine());
            double area = Math.PI * radius * radius;
            System.out.println("Area of the circle: " + area);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
    public static void main(String args[]) {
        calculateArea();
    }
}
```



```
E:\zzJAVA>javac CircleArea.java
Note: CircleArea.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java CircleArea
Enter the radius of the circle:
9
Area of the circle: 254.46900494077323

E:\zzJAVA>
```

6. WAP to reverse a number (functions with argument and no return type.)

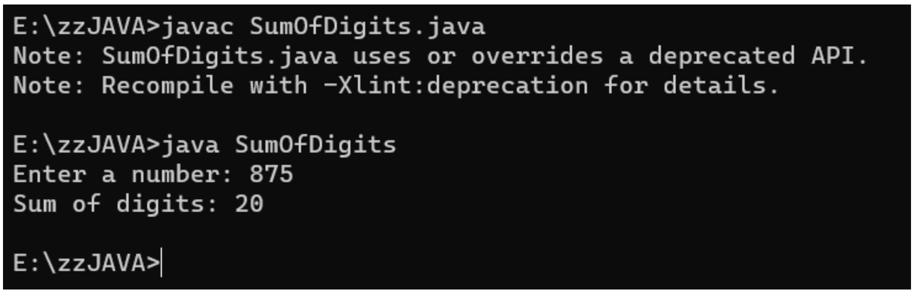
```
import java.io.*;

public class ReverseNumber {
    static void reverseNumber(int number) {
        int reversedNumber = 0;
        while (number != 0) {
            int digit = number % 10;
            reversedNumber = reversedNumber * 10 + digit;
            number /= 10;
        }
        System.out.println("Reversed Number: " + reversedNumber);
    }

    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());
            reverseNumber(inputNumber);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac ReverseNumber.java
Note: ReverseNumber.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java ReverseNumber
Enter a number: 5
Reversed Number: 5

E:\zzJAVA>java ReverseNumber
Enter a number: 565
Reversed Number: 565

E:\zzJAVA>
```

7. WAP to calculate sum of digits of a number (functions with argument and return type.)

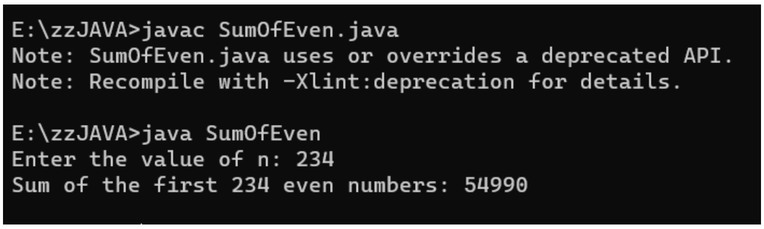
```
import java.io.*;
public class SumOfDigits {
    static int calculateSumOfDigits(int number) {
        int sum = 0;
        while (number != 0) {
            int digit = number % 10;
            sum += digit;
            number /= 10;
        }
        return sum;
    }
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());
            int sum = calculateSumOfDigits(inputNumber);
            System.out.println("Sum of digits: " + sum);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac SumOfDigits.java
Note: SumOfDigits.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SumOfDigits
Enter a number: 875
Sum of digits: 20

E:\zzJAVA>
```

8. WAP to calculate sum of n even numbers (functions with no argument and return type.)

```
import java.io.*;
public class SumOfEven {
    static void calculateSumOfEvenNumbers() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the value of n: ");
            int n = Integer.parseInt(dis.readLine());
            int sum = 0;
            int count = 0;
            int number = 2; // Starting from the first even number
            while (count < n) {
                sum += number;
                number += 2; // Move to the next even number
                count++;
            }
            System.out.println("Sum of the first " + n + " even numbers: " + sum);
        } catch (Exception e) {
         System.out.println(e);
        }
    }
    public static void main(String args[]) {
        calculateSumOfEvenNumbers();
    }
}
```

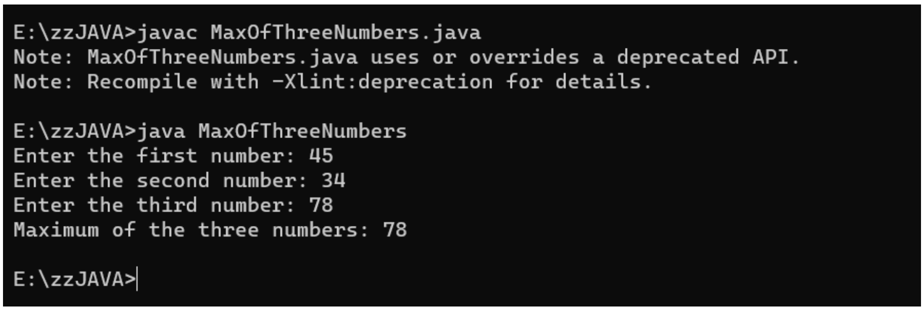


```
E:\zzJAVA>javac SumOfEven.java
Note: SumOfEven.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SumOfEven
Enter the value of n: 234
Sum of the first 234 even numbers: 54990
```

SECTION 4

1.WAP with nested functions to find the maximum of three numbers. Function1 should take in two arguments and find the maximum. Function2 should take in the third number and the maximum from function1 to find the maximum)

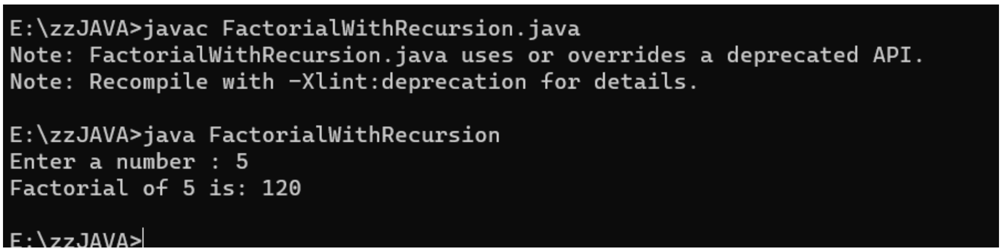
```
import java.io.*;
public class MaxOfThreeNumbers {
    static int findMax(int num1, int num2) {
        return (num1 > num2) ? num1 : num2;
    }
    static int findMaxOfThree(int num1, int num2, int num3) {
        int maxOfFirstTwo = findMax(num1, num2);
        return findMax(maxOfFirstTwo, num3);
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the first number: ");
            int num1 = Integer.parseInt(dis.readLine());
            System.out.print("Enter the second number: ");
            int num2 = Integer.parseInt(dis.readLine());
            System.out.print("Enter the third number: ");
            int num3 = Integer.parseInt(dis.readLine());
            int max = findMaxOfThree(num1, num2, num3);
            System.out.println("Maximum of the three numbers: " + max);
        } catch (Exception e) {
            System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac MaxOfThreeNumbers.java
Note: MaxOfThreeNumbers.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java MaxOfThreeNumbers
Enter the first number: 45
Enter the second number: 34
Enter the third number: 78
Maximum of the three numbers: 78

E:\zzJAVA>
```

3. WAP to find the factorial of n, using recursion.

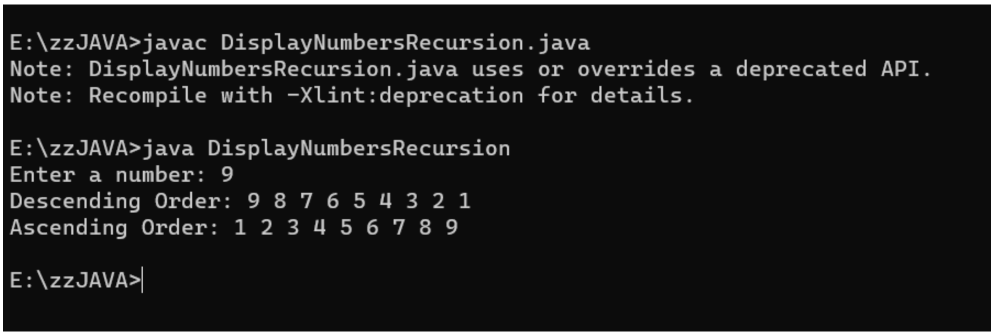
```
import java.io.*;
public class FactorialWithRecursion {
    static int factorial(int n) {
        if (n == 0 || n == 1) {
            return 1;
        } else {
            return n * factorial(n - 1);
        }
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number : ");
            int inputNumber = Integer.parseInt(dis.readLine());
            int result = factorial(inputNumber);
            System.out.println("Factorial of " + inputNumber + " is: " + result);
        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac FactorialWithRecursion.java
Note: FactorialWithRecursion.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java FactorialWithRecursion
Enter a number : 5
Factorial of 5 is: 120

E:\zzJAVA>
```

3. WAP to display numbers from n to 1 and vice versa, using recursion.

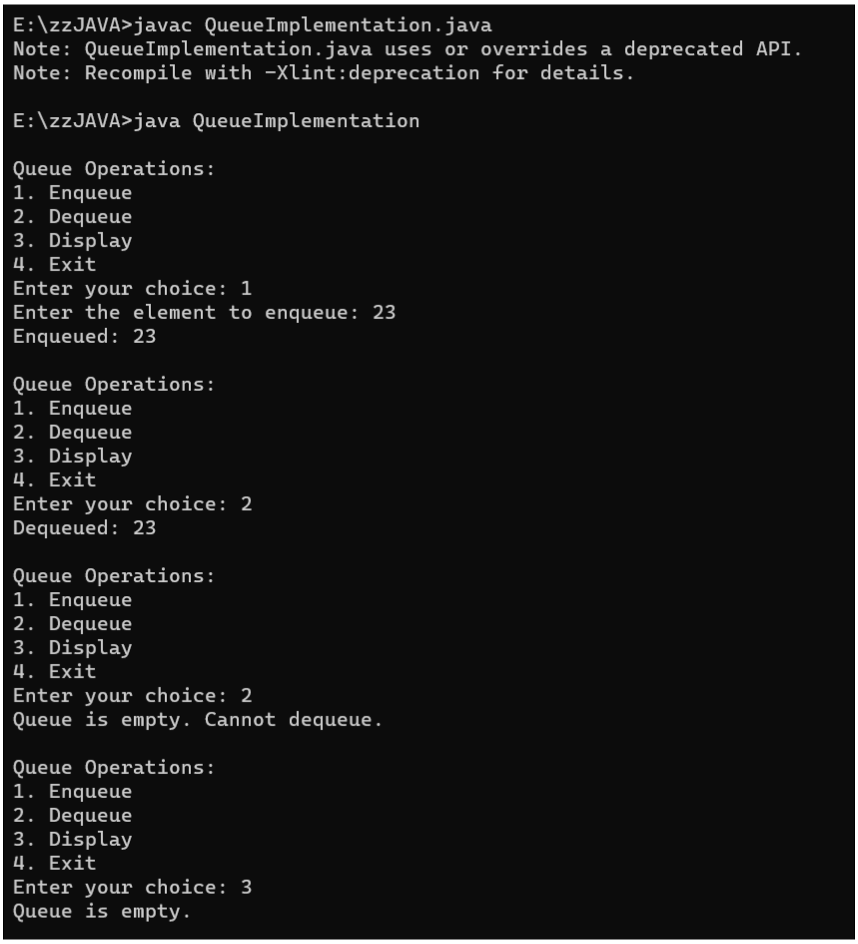
```
import java.io.*;
public class DisplayNumbersRecursion {
    static void displayDescending(int n) {
        if (n >= 1) {
            System.out.print(n + " ");
            displayDescending(n - 1);
        }
    }
    static void displayAscending(int n) {
        if (n >= 1) {
            displayAscending(n - 1);
            System.out.print(n + " ");
        }
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            int inputNumber = Integer.parseInt(dis.readLine());
            System.out.print("Descending Order: ");
            displayDescending(inputNumber);
            System.out.println(); // Move to the next line
            System.out.print("Ascending Order: ");
            displayAscending(inputNumber);
            System.out.println(); // Move to the next line

        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac DisplayNumbersRecursion.java
Note: DisplayNumbersRecursion.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java DisplayNumbersRecursion
Enter a number: 9
Descending Order: 9 8 7 6 5 4 3 2 1
Ascending Order: 1 2 3 4 5 6 7 8 9

E:\zzJAVA>
```

4. Using constructors, implement the operations of a queue.

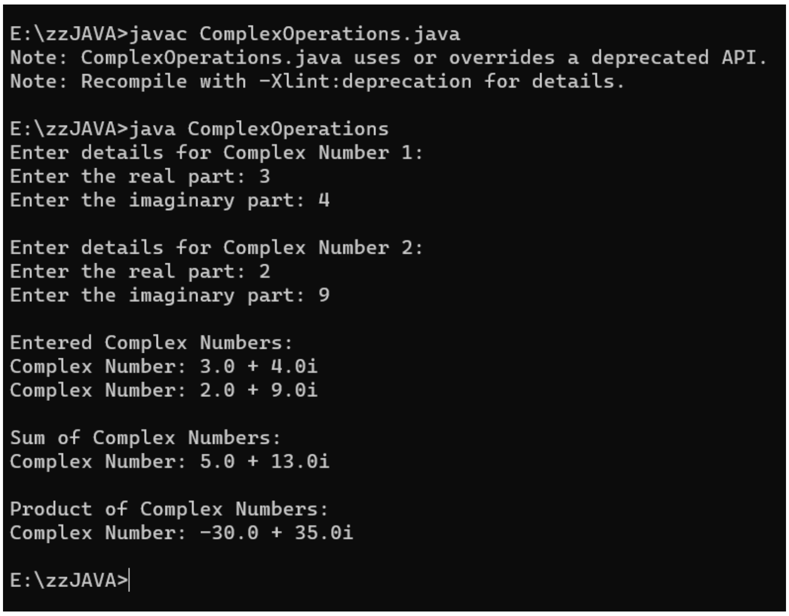
```
import java.io.*;
class Queue {
    private static final int MAX_SIZE = 5; // Maximum size of the queue
    private int[] array;
    private int front, rear, size;
    public Queue() {
        array = new int[MAX_SIZE];
        front = rear = -1;
        size = 0;
    }
    public boolean isEmpty() {
        return size == 0;
    }
    public boolean isFull() {
        return size == MAX_SIZE;
    }
    public void enqueue(int element) {
        if (isFull()) {
            System.out.println("Queue is full. Cannot enqueue.");
        } else {
            if (isEmpty()) {
                front = 0;
            }
            rear = (rear + 1) % MAX_SIZE;
            array[rear] = element;
            size++;
            System.out.println("Enqueued: " + element);
        }
    }
    public void dequeue() {
        if (isEmpty()) {
            System.out.println("Queue is empty. Cannot dequeue.");
        } else {
            int removedElement = array[front];
            if (front == rear) {
                front = rear = -1;
            } else {
                front = (front + 1) % MAX_SIZE;
            }
            size--;
            System.out.println("Dequeued: " + removedElement);
        }
    }
    public void display() {
        if (isEmpty()) {
            System.out.println("Queue is empty.");
        } else {
            System.out.print("Queue: ");
            int i = front;
            do {
                System.out.print(array[i] + " ");
                i = (i + 1) % MAX_SIZE;
            } while (i != (rear + 1) % MAX_SIZE);
            System.out.println();
        }
    }
}
public class QueueImplementation {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            Queue myQueue = new Queue();
            while (true) {
                System.out.println("\nQueue Operations:");
                System.out.println("1. Enqueue");
                System.out.println("2. Dequeue");
                System.out.println("3. Display");
        System.out.println("4. Exit");
        System.out.print("Enter your choice: ");
        int choice = Integer.parseInt(dis.readLine());
        switch (choice) {
          case 1:
            System.out.print("Enter the element to enqueue: ");
            int element = Integer.parseInt(dis.readLine());
            myQueue.enqueue(element);
            break;

          case 2:
            myQueue.dequeue();
            break;

          case 3:
            myQueue.display();
            break;

          case 4:
            System.out.println("Exiting the program.");
            System.exit(0);
            break;

          default:
            System.out.println("Invalid choice. Please enter a valid option.");
        }
      }
    } catch (Exception e) {
      System.out.println(e);
    }
  }
}
```



```
E:\zzJAVA>javac QueueImplementation.java
Note: QueueImplementation.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java QueueImplementation

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 1
Enter the element to enqueue: 23
Enqueued: 23

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 2
Dequeued: 23

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 2
Queue is empty. Cannot dequeue.

Queue Operations:
1. Enqueue
2. Dequeue
3. Display
4. Exit
Enter your choice: 3
Queue is empty.
```

5. Create a class complex having a real and imaginary part. Provide functions for read, display, add and multiplying two complex numbers

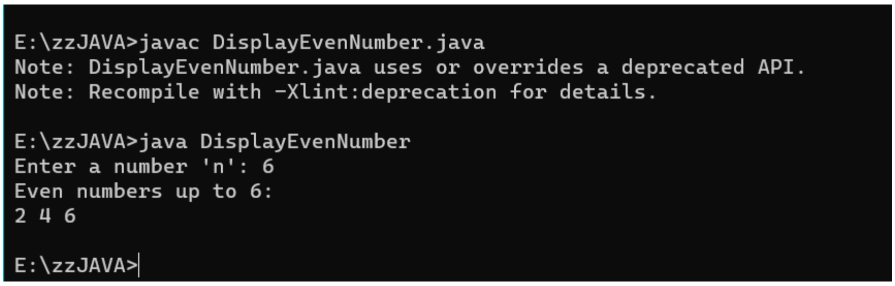
```
import java.io.*;
class Complex {
    private double real;
    private double imaginary;
    public Complex(double real, double imaginary) {
        this.real = real;
        this.imaginary = imaginary;
    }
    public void read() {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the real part: ");
            real = Double.parseDouble(dis.readLine());
            System.out.print("Enter the imaginary part: ");
            imaginary = Double.parseDouble(dis.readLine());
        } catch (Exception e) {
           System.out.println(e);
        }  }
    public void display() {
        System.out.println("Complex Number: " + real + " + " + imaginary + "i");
    }
    public Complex add(Complex other) {
        double resultReal = this.real + other.real;
        double resultImaginary = this.imaginary + other.imaginary;
        return new Complex(resultReal, resultImaginary);
    }
    public Complex multiply(Complex other) {
        double resultReal = (this.real * other.real) - (this.imaginary * other.imaginary);
        double resultImaginary = (this.real * other.imaginary) + (this.imaginary * other.real);
        return new Complex(resultReal, resultImaginary);
    }}
public class ComplexOperations {
    public static void main(String[] args) {
        Complex complex1 = new Complex(0, 0);
        Complex complex2 = new Complex(0, 0);
        System.out.println("Enter details for Complex Number 1:");
        complex1.read();
        System.out.println("\nEnter details for Complex Number 2:");
        complex2.read();
        System.out.println("\nEntered Complex Numbers:");
        complex1.display();
        complex2.display();
        Complex sumResult = complex1.add(complex2);

        System.out.println("\nSum of Complex Numbers:");

        sumResult.display();

        Complex productResult = complex1.multiply(complex2);

        System.out.println("\nProduct of Complex Numbers:");

        productResult.display();

    }

}
```



```
E:\zzJAVA>javac ComplexOperations.java
Note: ComplexOperations.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java ComplexOperations
Enter details for Complex Number 1:
Enter the real part: 3
Enter the imaginary part: 4

Enter details for Complex Number 2:
Enter the real part: 2
Enter the imaginary part: 9

Entered Complex Numbers:
Complex Number: 3.0 + 4.0i
Complex Number: 2.0 + 9.0i

Sum of Complex Numbers:
Complex Number: 5.0 + 13.0i

Product of Complex Numbers:
Complex Number: -30.0 + 35.0i

E:\zzJAVA>
```

6. WAP to display even numbers upto ‘n’ using a static function

```
import java.io.*;

public class DisplayEvenNumber {

    static void displayEvenNumbers(int n) {

        System.out.println("Even numbers up to " + n + ":");

        for (int i = 2; i <= n; i += 2) {

            System.out.print(i + " ");

        }

        System.out.println();

    }
    public static void main(String args[]) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number 'n': ");
            int inputNumber = Integer.parseInt(dis.readLine());
            displayEvenNumbers(inputNumber);
        } catch (Exception e) {
          System.out.println(e);
        }
    }
}
```

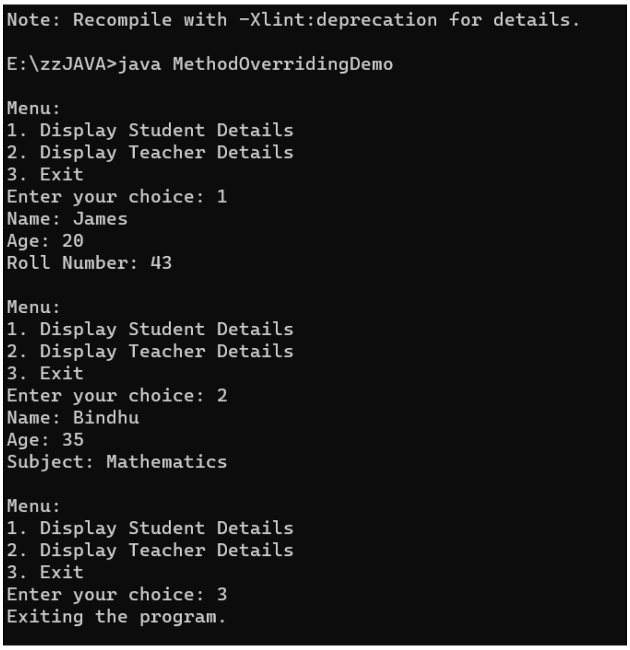


```
E:\zzJAVA>javac DisplayEvenNumber.java
Note: DisplayEvenNumber.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java DisplayEvenNumber
Enter a number 'n': 6
Even numbers up to 6:
2 4 6

E:\zzJAVA>
```

SECTION 5

1. WAP (menu driven) to demonstrate method overriding in java, by displaying details of a student, and a teacher.

```
import java.io.*;
class Person {
    private String name;
    private int age;
    public Person(String name, int age) {
        this.name = name;
        this.age = age;
    }
    public void displayDetails() {
        System.out.println("Name: " + name);
        System.out.println("Age: " + age);
    }
}
class Student extends Person {
    private int rollNumber;
    public Student(String name, int age, int rollNumber) {
        super(name, age);
        this.rollNumber = rollNumber;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Roll Number: " + rollNumber);
    }
}
class Teacher extends Person {
    private String subject;
    public Teacher(String name, int age, String subject) {
        super(name, age);
        this.subject = subject;
    }
    public void displayDetails() {
        super.displayDetails();
        System.out.println("Subject: " + subject);
    }
}
public class MethodOverridingDemo {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            while (true) {
                System.out.println("\nMenu:");
                System.out.println("1. Display Student Details");
                System.out.println("2. Display Teacher Details");
                System.out.println("3. Exit");
                System.out.print("Enter your choice: ");
                int choice = Integer.parseInt(dis.readLine());
                switch (choice) {
                    case 1:
                        Student student = new Student("James", 20, 43);
                        student.displayDetails();
                        break;
                    case 2:
                        Teacher teacher = new Teacher("Bindhu", 35, "Mathematics");
                        teacher.displayDetails();
                        break;
                    case 3:
                        System.out.println("Exiting the program.");
                        System.exit(0);
                        break;
                    default:
                        System.out.println("Invalid choice. Please enter a valid option.");
                }
            }
        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```

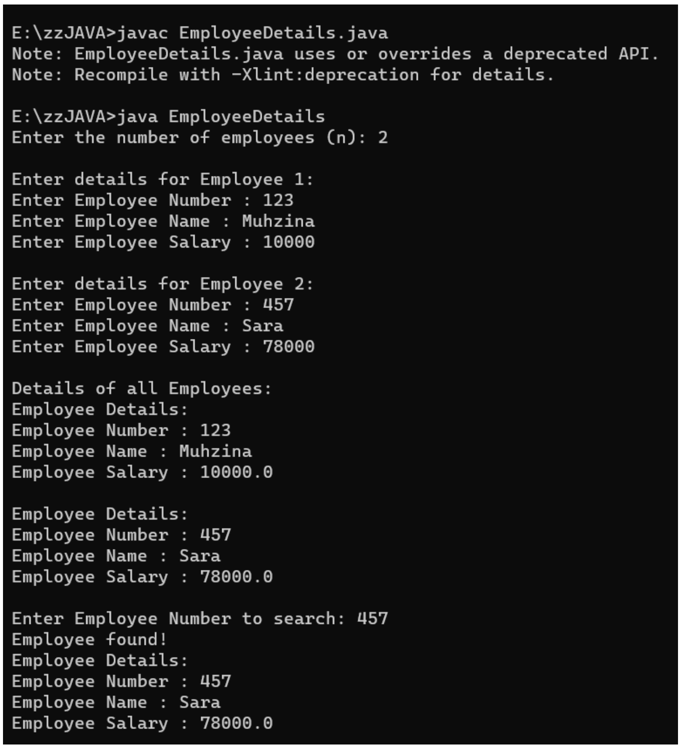


```
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java MethodOverridingDemo

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 1
Name: James
Age: 20
Roll Number: 43

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 2
Name: Bindhu
Age: 35
Subject: Mathematics

Menu:
1. Display Student Details
2. Display Teacher Details
3. Exit
Enter your choice: 3
Exiting the program.
```

2. Create a class for employee having eno,ename and esal as data members. Provide functions for reading and displaying employee details. (Accept information of n employees in the main function, display the same and search for an emp (using eno)).

```
import java.io.*;

class Employee {

    private int eno;

    private String ename;

    private double esal;

    public void readDetails() {

        try {

            DataInputStream dis = new DataInputStream(System.in);

            System.out.print("Enter Employee Number : ");

            eno = Integer.parseInt(dis.readLine());

            System.out.print("Enter Employee Name : ");

            ename = dis.readLine();

            System.out.print("Enter Employee Salary : ");

            esal = Double.parseDouble(dis.readLine());

        } catch (Exception e) {

           System.out.println(e);
```

```java
        }
    }
    public void displayDetails() {
        System.out.println("Employee Details:");
        System.out.println("Employee Number : " + eno);
        System.out.println("Employee Name : " + ename);
        System.out.println("Employee Salary : " + esal);
    }
    public boolean searchEmployee(int searchEno) {
        return eno == searchEno;
    }
}
public class EmployeeDetails {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the number of employees (n): ");
            int n = Integer.parseInt(dis.readLine());
            Employee[] employees = new Employee[n];
            for (int i = 0; i < n; i++) {
                employees[i] = new Employee();
                System.out.println("\nEnter details for Employee " + (i + 1) + ":");
                employees[i].readDetails();
            }
            System.out.println("\nDetails of all Employees:");
            for (int i = 0; i < n; i++) {
                employees[i].displayDetails();
                System.out.println();
            }
            System.out.print("Enter Employee Number to search: ");
            int searchEno = Integer.parseInt(dis.readLine());
            boolean found = false;
            for (int i = 0; i < n; i++) {
                if (employees[i].searchEmployee(searchEno)) {
                    System.out.println("Employee found!");
                    employees[i].displayDetails();
                    found = true;
                    break;
                }
            }
            if (!found) {
                System.out.println("Employee not found.");
            }
        } catch (Exception e) {
          System.out.println(e);
        }
    }
}
```

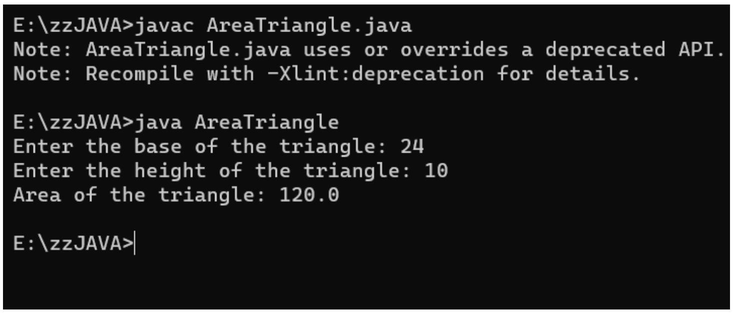


```
E:\zzJAVA>javac EmployeeDetails.java
Note: EmployeeDetails.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java EmployeeDetails
Enter the number of employees (n): 2

Enter details for Employee 1:
Enter Employee Number : 123
Enter Employee Name : Muhzina
Enter Employee Salary : 10000

Enter details for Employee 2:
Enter Employee Number : 457
Enter Employee Name : Sara
Enter Employee Salary : 78000

Details of all Employees:
Employee Details:
Employee Number : 123
Employee Name : Muhzina
Employee Salary : 10000.0

Employee Details:
Employee Number : 457
Employee Name : Sara
Employee Salary : 78000.0

Enter Employee Number to search: 457
Employee found!
Employee Details:
Employee Number : 457
Employee Name : Sara
Employee Salary : 78000.0
```

SECTION 6

1. Program to implement run time polymorphism in Java using interface, wrt calculating area of a triangle.

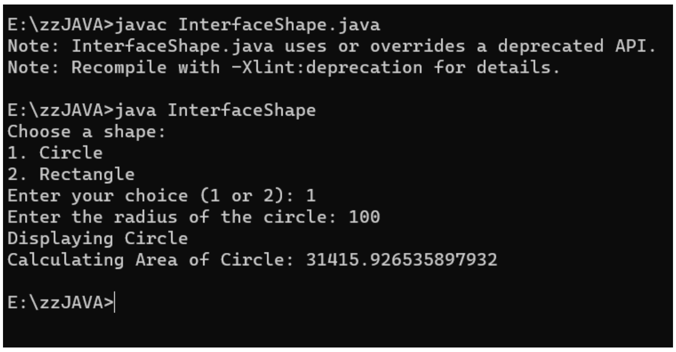
```
import java.io.*;
interface Shape {
    double calculateArea();
}
class Triangle implements Shape {
    private double base;
    private double height;
    public Triangle(double base, double height) {
        this.base = base;
        this.height = height;
    }
    public double calculateArea() {
        return 0.5 * base * height;
    }
}
public class AreaTriangle {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the base of the triangle: ");
            double base = Double.parseDouble(dis.readLine());
            System.out.print("Enter the height of the triangle: ");
            double height = Double.parseDouble(dis.readLine());
            Triangle triangle = new Triangle(base, height);
            System.out.println("Area of the triangle: " + triangle.calculateArea());
        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac AreaTriangle.java
Note: AreaTriangle.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java AreaTriangle
Enter the base of the triangle: 24
Enter the height of the triangle: 10
Area of the triangle: 120.0

E:\zzJAVA>
```

2. Create an interface Shape having two prototypes disp() and calc(), to diplay the shape and calculate volume respectively. Create two classes circle and rectangle which implements the above interface. In the main function create a reference of Shape depending on the user-choice.

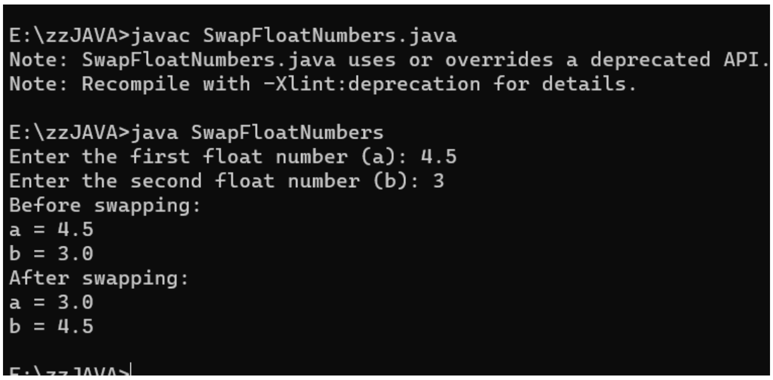
```
import java.io.*;
interface Shape {
    void disp();
    void calc();
}
class Circle implements Shape {
    private double radius;
    public Circle(double radius) {
        this.radius = radius;
    }
    public void disp() {
        System.out.println("Displaying Circle");
    }
    public void calc() {
        double area = Math.PI * radius * radius;
        System.out.println("Calculating Area of Circle: " + area);
    }
}
class Rectangle implements Shape {
    private double length;
    private double width;
    public Rectangle(double length, double width) {
        this.length = length;
        this.width = width;
    }
    public void disp() {
        System.out.println("Displaying Rectangle");
    }
    public void calc() {
        double area = length * width;
        System.out.println("Calculating Area of Rectangle: " + area);
    }
}
public class InterfaceShape {
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.println("Choose a shape:");
            System.out.println("1. Circle");
            System.out.println("2. Rectangle");
            System.out.print("Enter your choice (1 or 2): ");
            int choice = Integer.parseInt(dis.readLine());
            Shape shape;
            if (choice == 1) {
                System.out.print("Enter the radius of the circle: ");
                double radius = Double.parseDouble(dis.readLine());
                shape = new Circle(radius);
            } else if (choice == 2) {
                System.out.print("Enter the length of the rectangle: ");
                double length = Double.parseDouble(dis.readLine());
                System.out.print("Enter the width of the rectangle: ");
                double width = Double.parseDouble(dis.readLine());
                shape = new Rectangle(length, width);
            } else {
                System.out.println("Invalid choice. Exiting the program.");
                return;
            }
            shape.disp();
            shape.calc();
        } catch (Exception e) {
          System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac InterfaceShape.java
Note: InterfaceShape.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java InterfaceShape
Choose a shape:
1. Circle
2. Rectangle
Enter your choice (1 or 2): 1
Enter the radius of the circle: 100
Displaying Circle
Calculating Area of Circle: 31415.926535897932

E:\zzJAVA>
```

3. WAP to implement a function using call by value to swap two float numbers.

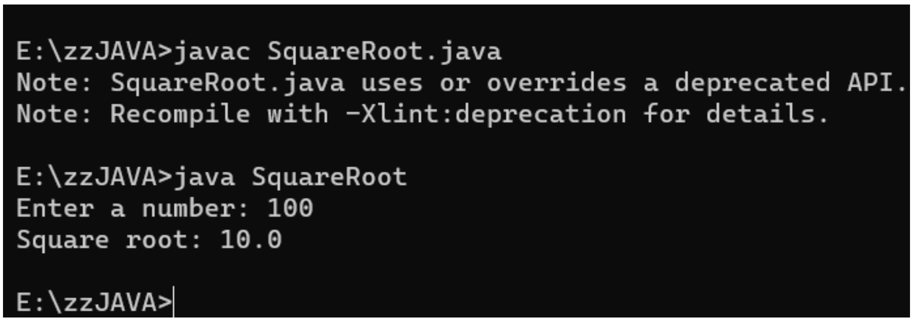
```
import java.io.*;
public class SwapFloatNumbers
 {
        static void swap(float a, float b) {
        System.out.println("Before swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
        float temp = a;
        a = b;
        b = temp;
        System.out.println("After swapping:");
        System.out.println("a = " + a);
        System.out.println("b = " + b);
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter the first float number (a): ");
            float a = Float.parseFloat(dis.readLine());
            System.out.print("Enter the second float number (b): ");
            float b = Float.parseFloat(dis.readLine());
            swap(a, b);
        } catch (Exception e) {
          System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac SwapFloatNumbers.java
Note: SwapFloatNumbers.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SwapFloatNumbers
Enter the first float number (a): 4.5
Enter the second float number (b): 3
Before swapping:
a = 4.5
b = 3.0
After swapping:
a = 3.0
b = 4.5

E:\zzJAVA>
```

4. WAP to implement a function using call by reference to find the square root of a given number.

```
import java.io.*;
public class SquareRoot {
    static void findSquareRoot(double[] number) {
        if (number[0] >= 0) {
            number[0] = Math.sqrt(number[0]);
        } else {
            System.out.println("Cannot find square root of a negative number.");
        }
    }
    public static void main(String[] args) {
        try {
            DataInputStream dis = new DataInputStream(System.in);
            System.out.print("Enter a number: ");
            double[] inputNumber = { Double.parseDouble(dis.readLine()) };
            findSquareRoot(inputNumber);
            System.out.println("Square root: " + inputNumber[0]);
        } catch (Exception e) {
           System.out.println(e);
        }
    }
}
```



```
E:\zzJAVA>javac SquareRoot.java
Note: SquareRoot.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java SquareRoot
Enter a number: 100
Square root: 10.0

E:\zzJAVA>
```

Section 7:

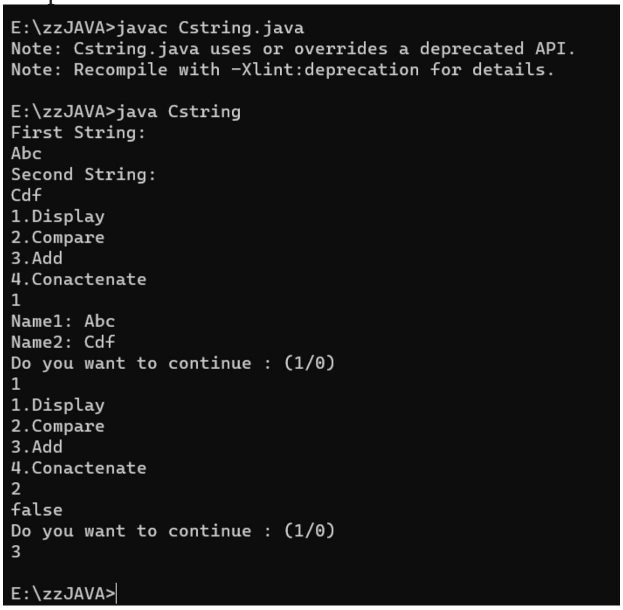
1. Create a class for Cstring having a string data member and provide functions for read , display, compare(return Boolean value),add and concatenate.

Code:

```
import java.io.*;
import java.util.Scanner;
import java.lang.*;

class Cstring{
static void display(String name1,String name2){
System.out.println("Name1: "+name1);
System.out.println("Name2: "+name2);
}

static void compare(String name1,String name2){
        System.out.println(name1==name2);
}
static void add(String name1,String name2){
        System.out.println(name1+name2);
}

static void concate(String name1,String name2){
        System.out.println(name1.concat(name2));
}


public static void main(String args[]){
try{
String name1,name2;
int c,ch=1;

DataInputStream dir=new DataInputStream(System.in);

Scanner console=new Scanner(System.in);
System.out.println("First String: ");
name1=console.nextLine();

System.out.println("Second String: ");
name2=console.nextLine();

do{
        System.out.println("1.Display\n2.Compare\n3.Add\n4.Conactenate");
        c=Integer.parseInt(dir.readLine());

        switch(c){

case 1:
display(name1,name2);break;

case 2:
compare(name1,name2);break;

case 3:
```

```
add(name1,name2);
break;

case 4:
concate(name1,name2);break;

}
System.out.println("Do you want to continue : (1/0)");

ch=Integer.parseInt(dir.readLine());
}while(ch==1);

}


catch(Exception e){
System.out.println("error : "+e);
}

}
}
```

Output:



```
E:\zzJAVA>javac Cstring.java
Note: Cstring.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java Cstring
First String:
Abc
Second String:
Cdf
1.Display
2.Compare
3.Add
4.Conactenate
1
Name1: Abc
Name2: Cdf
Do you want to continue : (1/0)
1
1.Display
2.Compare
3.Add
4.Conactenate
2
false
Do you want to continue : (1/0)
3

E:\zzJAVA>
```

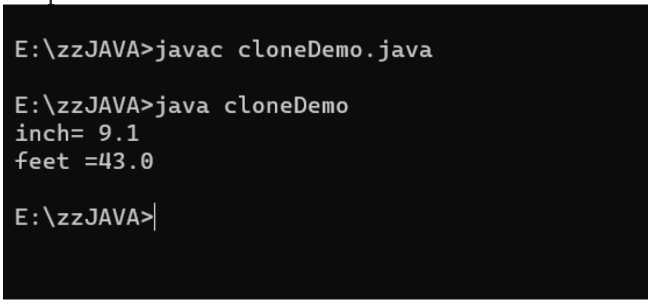
2. Write a program to implement object cloning for the class Distance which has inch and feet as data members.

Code:

```
import java.io.*;
import java.lang.*;
class Dist implements Cloneable{

        Double inch,feet;
        Dist(Double a,Double b){
        inch=a;
        feet=b;
}
```

```
        public Object clone(){
        try{
                return super.clone();
}
        catch(CloneNotSupportedException c){
        System.out.println("error : "+c);
return this;
}
}
}
class cloneDemo{
public static void main(String args[]){
        Dist obj1= new Dist(9.1,43.0);
        Dist obj2=(Dist)obj1.clone();
        System.out.println("inch= "+obj2.inch);
        System.out.println("feet ="+obj2.feet);
}
}
```

Output:



```
E:\zzJAVA>javac cloneDemo.java

E:\zzJAVA>java cloneDemo
inch= 9.1
feet =43.0

E:\zzJAVA>
```

3. Write a program to create a menu driven program for performing the following operations.

□ Length of a given string

□ Compare for equality

□ Extract a substring from a string.

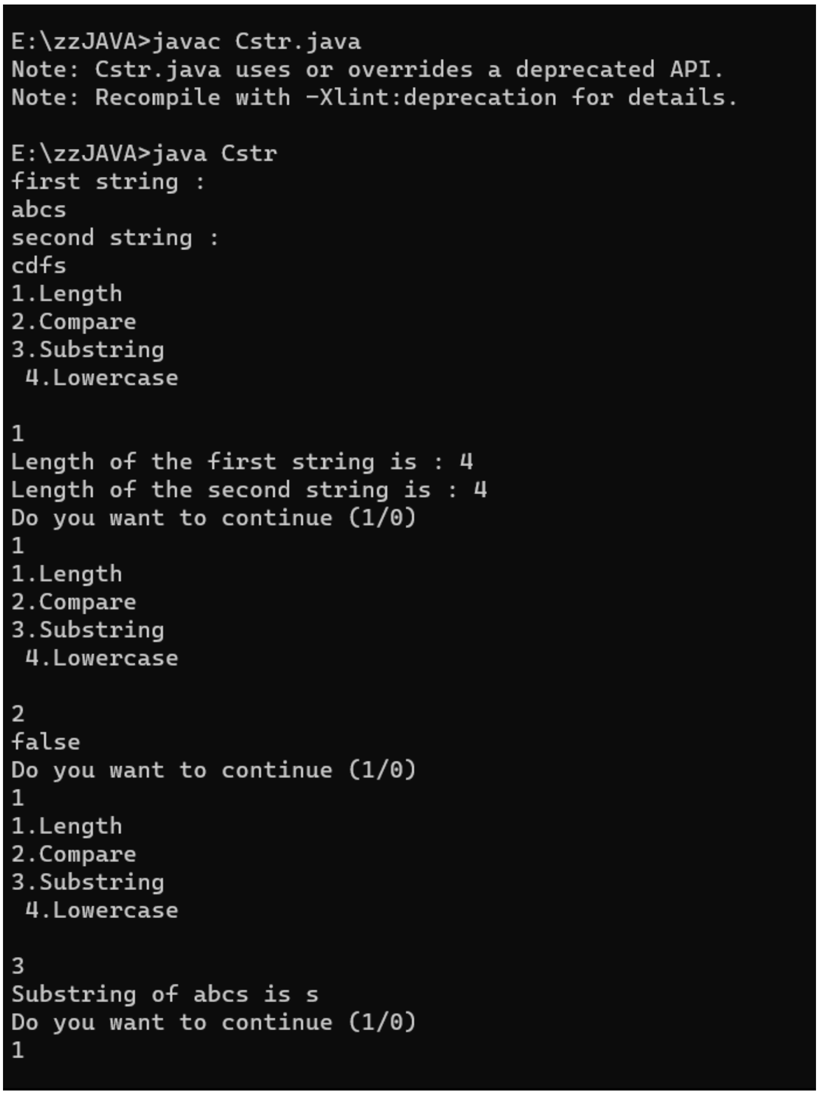
□ Convert to uppercase and lowercase

Code:

```
import java.io.*;

import java.util.Scanner;

import java.lang.*;


class Cstr{

        static void length(String name1,String name2){

                System.out.println("Length of the first string is : "+name1.length());

                System.out.println("Length of the second string is : "+name2.length());
```

```
}
static void compare(String name1,String name2){
        System.out.println(name1==name2);
}

static void substr(String name1){
        System.out.println("Substring of "+ name1+" is "+name1.substring(3));
}
static void lowercase(String name1,String name2){
System.out.println(name1.toLowerCase());
System.out.println(name2.toLowerCase());
}
public static void main(String args[]){
try{
String name1,name2;
int c,ch=1;

DataInputStream dir=new DataInputStream(System.in);
Scanner console=new Scanner(System.in);

System.out.println("first string : ");
name1 = console.nextLine();

System.out.println("second string : ");
name2 = console.nextLine();

do{
System.out.println("1.Length\n2.Compare\n3.Substring\n 4.Lowercase\n");
c=Integer.parseInt(dir.readLine());

switch(c){
        case 1:
                length(name1,name2);break;
        case 2:
                compare(name1,name2);break;
        case 3:
                substr(name1);break;
        case 4:
                lowercase(name1,name2);break;

}
System.out.println("Do you want to continue (1/0) ");
ch=Integer.parseInt(dir.readLine());
}while(ch==1);


}
catch(Exception e){
        System.out.println("error "+e);
}
}
}
```

Output:



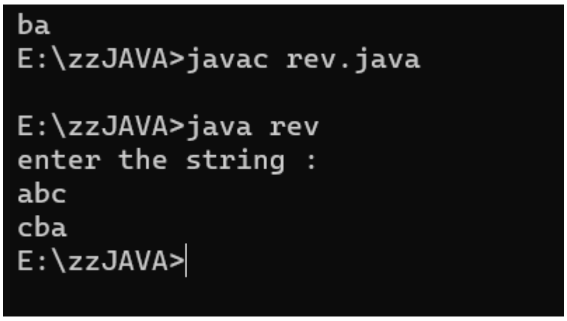
```
E:\zzJAVA>javac Cstr.java
Note: Cstr.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\zzJAVA>java Cstr
first string :
abcs
second string :
cdfs
1.Length
2.Compare
3.Substring
 4.Lowercase

1
Length of the first string is : 4
Length of the second string is : 4
Do you want to continue (1/0)
1
1.Length
2.Compare
3.Substring
 4.Lowercase

2
false
Do you want to continue (1/0)
1
1.Length
2.Compare
3.Substring
 4.Lowercase

3
Substring of abcs is s
Do you want to continue (1/0)
1
```

4. Write a program to reverse a string

Code:

```
import java.util.Scanner;

import java.lang.*;

class rev{

	public static void main(String args[]){

try{

int i;

Scanner console=new Scanner (System.in);

System.out.println("enter the string :");

String name=console.nextLine();

int l=name.length();
for(i=l-1;i>=0;i--){

        System.out.print(name.charAt(i));



}

}

catch(Exception e){

System.out.println("error : "+e);

}

}

}
```

Output:



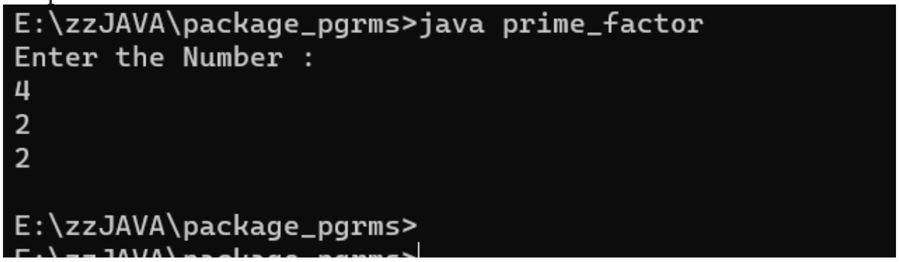
```
ba
E:\zzJAVA>javac rev.java

E:\zzJAVA>java rev
enter the string :
abc
cba
E:\zzJAVA>
```

5. Write a program to calculate the prime factors of a given number, using packages.
   Code:

```
package pack1;
import java.io.*;

public class p1{
public void disp(int temp){
        int i=2;
        while(temp!=0 && temp>0){
                if(temp==0 || i<=0){
                        break;
                        }
                else{
                        if(temp%i==0){
                                System.out.println(i);
                                temp=temp/i;
                        }
                        else{
                        i=i+1;
                        }
                }
        }
}
}
```

```
import java.io.*;
import java.lang.*;
import pack1.p1;

class prime_factor{
public static void main(String args[]){
try{
int m=1,d,temp;
DataInputStream dir=new DataInputStream(System.in);
System.out.println("Enter the Number :");
int num = Integer.parseInt(dir.readLine());
temp=num;
p1 obj=new p1();
obj.disp(temp);
}
catch(Exception e){
System.out.println("error : "+e);
}

}
}
```

Output:

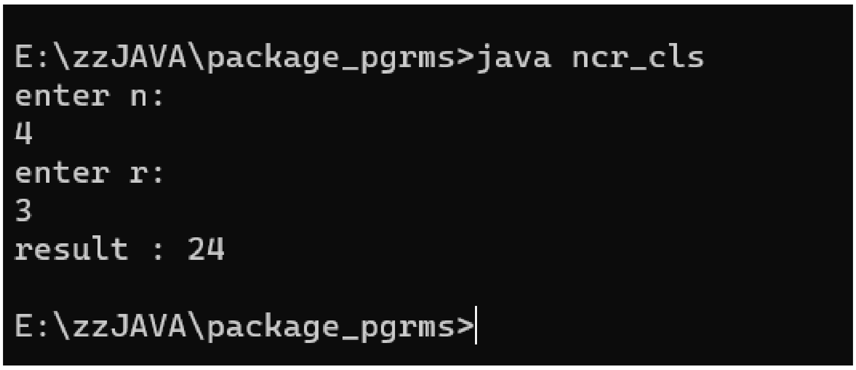


```
E:\zzJAVA\package_pgrms>java prime_factor
Enter the Number :
4
2
2

E:\zzJAVA\package_pgrms>
E:\zzJAVA\package_pgrms>
```

6. Write a program to calculate the value of nCr for given value of n & r, using packages.

Code:

```
package pack1;
import java.io.*;

public class p2{
public int ncr_method(int n,int d){
int i;
int result=1,result1=1,result2;
for(i=1;i<=n;i++){
result=result*i;
}
for(i=1;i<=d;i++){
result=result*i;
}
for(i=1;i<=d;i++){
result1=result1*i;
}
result2=result/result1;
return result2;
}
}

import java.io.*;
import pack1.*;
import java.lang.*;

class ncr_cls{

public static void main(String args[]){
try{
int result2,i;
DataInputStream dir=new DataInputStream(System.in);
System.out.println("enter n: ");
int n=Integer.parseInt(dir.readLine());

System.out.println("enter r: ");
int r=Integer.parseInt(dir.readLine());

int d=n-r;
p2 obj = new p2();
result2 =obj.ncr_method(n,d);
System.out.println("result : "+result2);
}
catch(Exception e){
            System.out.println(e);
}
}
}
```



```
E:\zzJAVA\package_pgrms>java ncr_cls
enter n:
4
enter r:
3
result : 24

E:\zzJAVA\package_pgrms>
```