

### **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);
}
}
}
```

### **Output**

```
E:\java2023>javac SwapNum.java
Note: SwapNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\java2023>java SwapNum
Enter a:
10
Enter b:
50
Before swapping:
a: 10
b: 50
After swapping:
a: 50
b: 10

E:\java2023>javac MaxNum.java
Note: MaxNum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\java2023>java MaxNum
Enter a:
120
Enter b:
40
Enter c:
66
max: 120
```

**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);
            System.out.println("Min number is: "+min);
        }
        catch(Exception e)
        {
            System.out.println(e);
        }
    }
}
```

```
}  
}  
}
```

### Output

```
E:\java2023>javac MinNum.java  
Note: MinNum.java uses or overrides a deprecated API.  
Note: Recompile with -Xlint:deprecation for details.  
  
E:\java2023>java MinNum  
Enter a:  
50  
Enter b:  
60  
Enter c:  
88  
Min number is: 50
```

### 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);
}
}
}

```

### Output

```

E:\java2023>javac SearchArray.java
Note: SearchArray.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\java2023>java SearchArray
Enter size of array:
4
Enter array elements:
5
9
4
6
Array elements are:
5
9
4
6
Enter element to search:
9
Element 9 found at index 1

```

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

```
import java.io.*;

public class SortArray {

    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]);
            }

            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);
        }
    }
}
```

```

    }
    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;
                }
            }
        }
    }
}

```

### Output

```

E:\java2023>javac SortArray.java
Note: SortArray.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\java2023>java SortArray
Enter size of array:
4
Enter array elements:
10
2
6
52
Array elements before sorting:
10
2
6
52
Array elements after sorting in ascending order:
2
6
10
52

```

**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 ArraySum
{
    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 :");
            s = i.readLine();
            rows = Integer.parseInt(s);
            System.out.println("Enter the number of columns :");
            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 : ");
                    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);
}
}
}

```

## Output

```
E:\java2023>javac ArraySum.java
Note: ArraySum.java uses or overrides a deprecated API.
Note: Recompile with -Xlint:deprecation for details.

E:\java2023>java ArraySum
Enter the number of rows :
2
Enter the number of columns :
2
Enter element : 5
Enter element : 4
Enter element : 2
Enter element : 6
2D Array:
5 4
2 6
Row-wise Sum:
Row 1: 9
Row 2: 8
Column-wise Sum:
Column 1: 7   Column 2: 10
```