

Experiment:2

1. Define a class Array with following member Field:

```
int data[];
```

Function:

```
Array( ) //create array data of size 10
```

```
Array(int size) // create array of size size
```

```
Array(int data[]) // initialize array with parameter array
```

```
void Reverse _an _array () //reverse element of an array
```

```
int Maximum _of _array () // find maximum element of array
```

```
int Average_of _array() //find average of element of array
```

```
void Sorting () //sort element of array
```

```
void display() //display element of array
```

```
int search(int no) //search element and return index else return -1
```

```
int size(); //return size of an arrayUse all the function in main method. Create different objects with different constructors.
```

Source code:

```
import java.util.Scanner;

public class ArrayLab2_1 {
    int[] arr;
    public ArrayLab2_1() {
```

```
        arr = new int[10];
    }
    public ArrayLab2_1(int n) {
        arr = new int[n];
    }
    public ArrayLab2_1(int... data) {
        arr = new int[data.length];
        int i = 0;
        for (int val : data) {
            arr[i] = val;
            i++;
        }
    }
    public void diplayArray() {
        for (int val : arr) {
            System.out.print(val + " ");
        }
    }
    public void Reverse_an_array() {
        System.out.print("\nReverse Array is : ");
        for (int i = arr.length - 1; i >= 0; i--) {
            System.out.print(arr[i] + " ");
        }
    }
    public int Maximum_of_array() {
```

```
        int max = arr[0];
        for (int i = 1; i < arr.length; i++) {
            if (max < arr[i]) {
                max = arr[i];
            }
        }
        return max;
    }

    public int Average_of_array() {
        int sum = 0;
        for (int val : arr) {
            sum += val;
        }
        return sum / arr.length;
    }

    public void Sorting() {
        int[] SortArray = arr.clone();
        for (int i = 0; i < SortArray.length; i++) {
            int min = i;
            for (int j = i + 1; j < SortArray.length; j++) {
                if (SortArray[j] < SortArray[min]) {
                    min = j;
                }
            }
            int temp = SortArray[i];
```

```
        SortArray[i] = SortArray[min];
        SortArray[min] = temp;
    }
    System.out.print("\nSort array : ");
    for (int val : SortArray) {
        System.out.print(val + " ");
    }
}

public int search(int n) {
    for (int i = 0; i < arr.length; i++) {
        if (arr[i] == n) {
            return i;
        }
    }
    return -1;
}

public int Size() {
    return arr.length;
}

public static void main(String[] args) {
    Scanner sc = new Scanner(System.in);
    System.out.print("Enter Size of Array : ");
    int n = sc.nextInt();
    ArrayLab2_1 arr1 = new ArrayLab2_1(n);
    System.out.print("Enter array element : ");
```

```
        for (int i = 0; i < n; i++) {
            arr1.arr[i] = sc.nextInt();
        }

        System.out.println("Array Size is : " +
arr1.Size());

        System.out.print("Display Array element : ");
        arr1.dipplayArray();
        arr1.Reverse_an_array();

        System.out.print("\nMaximum element in array : " +
arr1.Maximum_of_array());

        System.out.print("\nAverage is : " +
arr1.Average_of_array());

        arr1.Sorting();

        System.out.print("\nEnter Element for Search : ");
        int se=sc.nextInt();

        System.out.print("Element "+se+" having index is :
"+arr1.search(se));

        sc.close();
    }
}
```

Output:

```
C:\12302130501036>javac ArrayLab2_1.java
C:\12302130501036>java ArrayLab2_1
Enter Size of Array : 3
Enter array element : 1 2 3
Array Size is : 3
Display Array element : 1 2 3
Reverse Array is : 3 2 1
Maximum element in array : 3
Average is : 2
Sort array : 1 2 3
Enter Element for Search : 2
Element 2 having index is : 1
```

2. Define a class Matrix with following Field:

int row, column;

float mat[][]

Function:

Matrix(int a[][])

Matrix()

Matrix(int rwo, int col)

void readMatrix() //read element of array

float [][] transpose() //find transpose of first matrix

float [][] matrixMultiplication(Matrix second) //multiply two matrices and return result

void displayMatrix(float [][]a) //display content of argument array

void displayMatrix() //display content

float maximum_of_array() // return maximum element of first array

float average_of_array() // return average of first array

create three object of Matrix class with different constructors in main

and test all the functions in main

Source code:

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

```
public class MATRIX {
```

```
int row, column;
float[][] mat;
Scanner sc;
public MATRIX() {
    row = column = 0;
    this.mat = new float[row][column];
}
public MATRIX(int row, int column) {
    this.row = row;
    this.column = column;
    this.mat = new float[row][column];
}
public MATRIX(int[][] a) {
    row = a.length;
    column = a[0].length;
    mat = new float[row][column];
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            this.mat[i][j] = (float) a[i][j];
        }
    }
}
public void readMATRIX() {
    sc = new Scanner(System.in);
    System.out.println("Enter a MATRIX Element");
```

```
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < this.column; j++) {
                mat[i][j] = sc.nextFloat();
            }
        }
    }

    public float[][] transpose() {
        float[][] transposed = new float[row][column];
        for (int i = 0; i < row; i++) {
            for (int j = 0; j < column; j++) {
                transposed[j][i] = this.mat[i][j];
            }
        }
        return transposed;
    }

    public float[][] MATRIXMultiplication(MATRIX second) {
        float[][] result = new float[this.row][second.column];
        float mul = 0.0f;
        for (int i = 0; i < this.row; i++) {
            for (int j = 0; j < second.column; j++) {
                for (int k = 0; k < this.column; k++) {
                    mul += this.mat[i][k] * second.mat[k][j];
                }
                result[i][j] = mul;
                mul = 0.0f;
            }
        }
    }
}
```



```
    }  
    }  
    return result;  
}  
  
public void displayMATRIX(float[][] a) {  
    for (int i = 0; i < row; i++) {  
        for (int j = 0; j < column; j++) {  
            System.out.print(a[i][j] + " ");  
        }  
        System.out.println(" ");  
    }  
}  
  
public void displayMATRIX() {  
    System.out.println("MATRIX Element are: ");  
    for (int i = 0; i < row; i++) {  
        for (int j = 0; j < column; j++) {  
            System.out.print(this.mat[i][j] + " ");  
        }  
        System.out.println();  
    }  
}  
  
public float maximum_of_array() {  
    float max = this.mat[0][0];  
    for (int i = 0; i < this.row; i++) {  
        for (int j = 0; j < this.column; j++) {
```

```
        if (max < this.mat[i][j]) {
            max = this.mat[i][j];
        }
    }
}

return max;
}

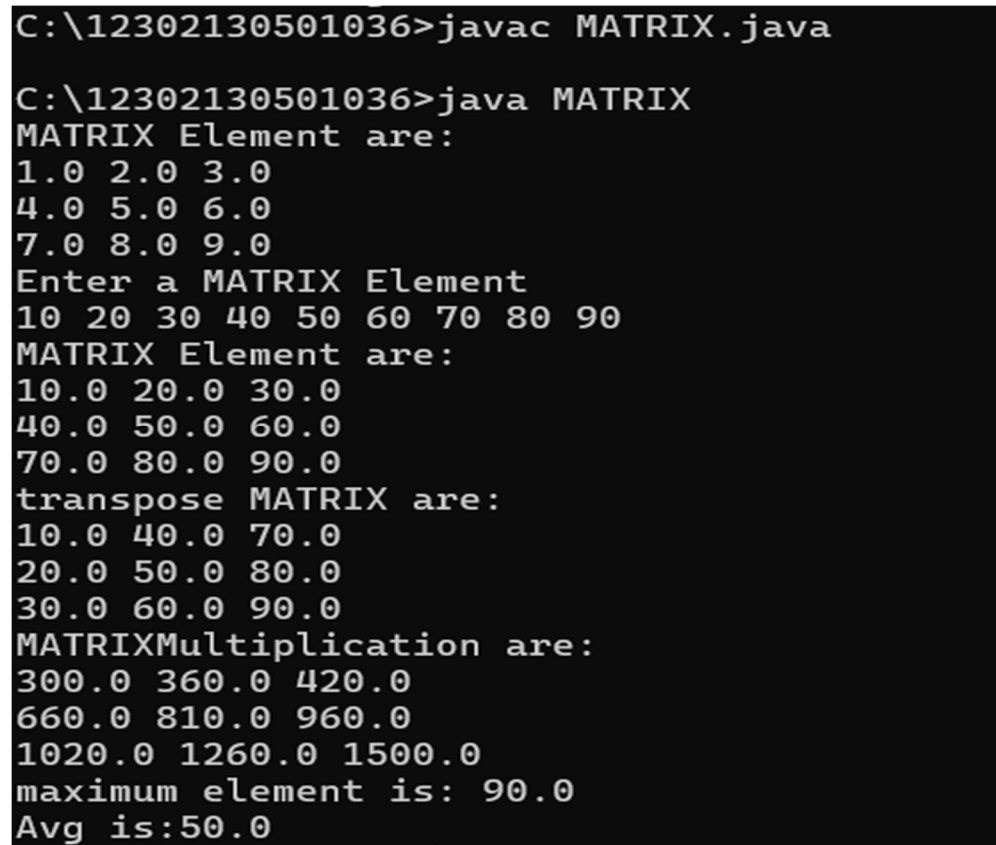
public float average_of_array() {
    float sum = 0.0f;
    for (int i = 0; i < row; i++) {
        for (int j = 0; j < column; j++) {
            sum += this.mat[i][j];
        }
    }

    float totalElement=row*column;
    return sum / totalElement;
}
```

```
public static void main(String[] args) {
    MATRIX MATRIX = new MATRIX();
    MATRIX MATRIX1 = new MATRIX(3, 3);
    MATRIX MATRIX2 = new MATRIX(new int[][]{{1, 2, 3}, {4, 5, 6}, {7, 8, 9}});
    MATRIX2.displayMATRIX();

    MATRIX1.readMATRIX();
}
```

```
MATRIX1.displayMATRIX();  
float[][] transpose = MATRIX1.transpose();  
System.out.println("transpose MATRIX are:");  
MATRIX1.displayMATRIX(transpose);  
float [][] MATRIXMulti=MATRIX1.MATRIXMultiplication(MATRIX2);  
System.out.println("MATRIXMulti are:");  
MATRIX1.displayMATRIX(MATRIXMulti);  
System.out.println("maximum element is:  
"+MATRIX1.maximum_of_array());  
System.out.print("Avg is:" + MATRIX1.average_of_array());  
}  
}
```

Output:

```
C:\12302130501036>javac MATRIX.java  
C:\12302130501036>java MATRIX  
MATRIX Element are:  
1.0 2.0 3.0  
4.0 5.0 6.0  
7.0 8.0 9.0  
Enter a MATRIX Element  
10 20 30 40 50 60 70 80 90  
MATRIX Element are:  
10.0 20.0 30.0  
40.0 50.0 60.0  
70.0 80.0 90.0  
transpose MATRIX are:  
10.0 40.0 70.0  
20.0 50.0 80.0  
30.0 60.0 90.0  
MATRIXMultiplication are:  
300.0 360.0 420.0  
660.0 810.0 960.0  
1020.0 1260.0 1500.0  
maximum element is: 90.0  
Avg is:50.0
```

3. Write a Program to demonstrate usage of different methods of Wrapper class.

Source code:

```
public class ClassWrapperdemo {  
    public static void main(String[] args) {  
        Character ch='c';  
        Integer i=100;  
        Float f=10.18f;  
        Boolean b=true;  
        Byte by=100;  
        Short s=99;  
        String st="Name";  
        Double d=100.50;  
  
        System.out.print("Integer:"+i+"\n");  
        System.out.print("Float:"+f+"\n");  
        System.out.print("Character:"+ch+'\n');  
        System.out.print("Boolean:"+b+'\n');  
        System.out.print("Byte:"+by+'\n');  
        System.out.print("Short:"+s.toString()+'\n');  
        System.out.print("String:"+st+'\n');  
        System.out.print("Double:"+d+'\n');  
    }  
}
```

Output:

```
C:\12302130501036>javac ClassWrapperdemo.java  
  
C:\12302130501036>java ClassWrapperdemo  
Integer:100  
Float:10.18  
Character:c  
Boolean:true  
Byte:100  
Short:99  
String:Name  
Double:100.5
```

4. Write a program to demonstrate usage of String and StringBuffer class.

Source code:

```
public class StringBuffer1Demo {  
    public static void main(String[] args) {  
        String s = "FullName";  
        String s1 = " Sname  ";  
        System.out.print("UpperCase: " + s.toUpperCase() + "\n");  
        System.out.print("LowerCase: " + s.toLowerCase() + "\n");  
        System.out.print("trim: " + s1.trim() + "\n");  
        System.out.print("SubString: " + s.substring(1, 5) + "\n");  
        System.out.print("Replace: " + s.replace('u', 'a') + "\n");  
        System.out.print("StartWith: " + s.startsWith("F") + "\n");  
        System.out.print("EndsWith: " + s.endsWith("e") + "\n");  
        System.out.print("CharAt: " + s.charAt(2) + "\n");  
        System.out.print("IndexOf: " + s.indexOf('u') + "\n");  
        System.out.print("Equals: " + s.equals(s1) + "\n");  
        System.out.println("\n\nString Buffer");  
        StringBuilder sb = new StringBuilder(s);  
        sb.append(" Sname");  
        System.out.println("After Append: " + sb.toString());  
        sb.insert(sb.length(), " Fname");  
        System.out.println("After Insert: " + sb);  
        sb.delete(12, sb.length());  
        System.out.println("After Delete: " + sb);  
        sb.reverse();  
        System.out.println("After Revese: " + sb);  
    }  
}
```

Output:

```
PS C:\12302130501036> javac StringBuffer1Demo.java
PS C:\12302130501036> java StringBuffer1Demo
UpperCase: FULLNAME
LowerCase: fullname
trim: Sname
SubString: ullN
Replace: FallName
StartWith: true
EndsWith: true
CharAt: l
IndexOf: 1
Equals: false

String Buffer
After Append: FullName Sname
After Insert: FullName Sname Fname
After Delete: FullName Sna
After Reverse: anS emaNlluF
PS C:\12302130501036>
```

5. Define a class Cipher with following data Field:

String plainText; int key

Functions:

Cipher(String plaintext,int key)

String Encryption()

String Decryption()

Read string and key from command prompt and replace every character of string with character which is key place down from current character. Example plainText = "GCET"

Key = 3

Encryption function written following String

"JFHW"

Decryption function will convert encrypted string to original form "GCET"

Source code:

```
import java.util.Scanner;

public class ClassCipher {
    String plainText;
    int key;

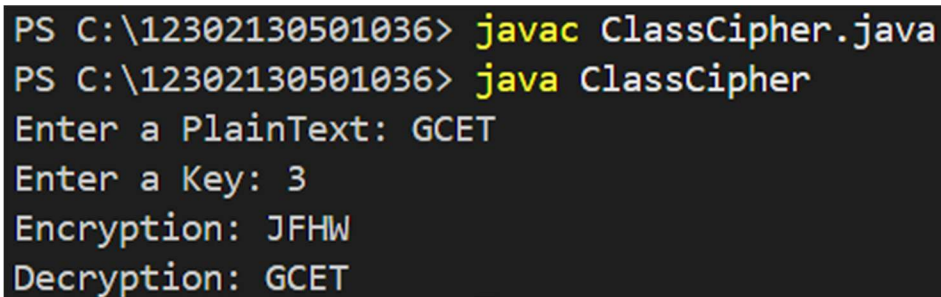
    public ClassCipher(String plainText, int key) {
        this.plainText = plainText;
        this.key = key;
    }

    public String Encryption() {
        char[] plainArray = plainText.toCharArray();
        for (int i = 0; i < plainArray.length; i++) {
            plainArray[i] = (char) (plainArray[i] + this.key);
        }
        plainText = new String(plainArray);
        return plainText;
    }

    public String Decryption() {
        char[] plainArray = plainText.toCharArray();
        for (int i = 0; i < plainArray.length; i++) {
            plainArray[i] = (char) (plainArray[i] - this.key);
        }
        plainText = new String(plainArray);
    }
}
```

```
        return plainText;
    }

    public static void main(String[] args) {
        String text;
        int key;
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter a PlainText: ");
        text = sc.nextLine();
        System.out.print("Enter a Key: ");
        try {
            key = sc.nextInt();
            ClassCipher cipher = new ClassCipher(text, key);
            System.out.print("Encryption: " + cipher.Encryption());
            System.out.print("\nDecryption: " + cipher.Decryption());
        } catch (Exception e) {
            System.out.println("Please Enter Interger Value");
        }
    }
}
```

Output:

```
PS C:\12302130501036> javac ClassCipher.java
PS C:\12302130501036> java ClassCipher
Enter a PlainText: GCET
Enter a Key: 3
Encryption: JFHW
Decryption: GCET
```