Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department.
OOPM / Odd Sem 2021-22 / Experiment 3.

YASH SARANG.
47 / D6AD.
EXPERIMENT - 3.

AIM: Program on 2D array, strings functions.

THEORY:
An array is a collection of elements of the same data type.
The following statement creates an array of integers:
int[] arr={38, 27, 43, 3, 9, 82, 10}.

The array variable arr is of the type int[], where int is the type of the elements.
This array is initialized with an initializer list. A list of values delimited by braces{}.
An alternate way of creating an array is using the new separator.
The following statement creates an array of length 7.
By default integer elements are initialized to be zero.
int[] arr = new int[7];

The string is basically an object that represents a sequence of char values,
for eg: char[] ch={'j', 'a', ''v', 'a', 'p', 'o', 'i', 'n', 't'}
String s= new String(ch) is same as String s = "Java point"

The Java String class provides a lot of methods to perform operations on strings,
like compare(), equals(), length(), compareTo(), subString(), etc.

The Java language string class implements serializable, comparable, char sequence interfaces and string objects can be created by a string literal.

METHODS OF STRING CLASS	DESCRIPTION
public boolean equals (Object anObject)	Compares the string to the specified object
public boolean equalsIgnoreCase (String another)	Compares the string to another string, ignoring case
public String concat (String str)	Concatenates the specified string to the end of this string
public int compareTo(String str)	Compares two strings and return int
public int compareToIgnoreCase(String str)	Compare two strings, ignoring the case differences.
public String substring(int beginIndex)	Returns a new string that is a substring of this string
public String substring(int beginIndex; int endIndex)	Returns a new string that is a substring of this string
public String toUpperCase()	Converts all characters in the string to uppercase.
public String toLowerCase()	Converts all characters in the string to lowercase.
public String trim()	Returns the copy of the string with whitespaces omitted.
public boolean startsWith(String prefix)	Test if this string starts with the specified matrix.
public boolean endsWith(String suffix)	Tests if this string ends with the specified matrix.
public char charAt(int index)	Returns the char value at the specified index
public int length()	Returns the length of the string.

CONCLUSION:

Through these programs, we learned about arrays, strings, and string functions.

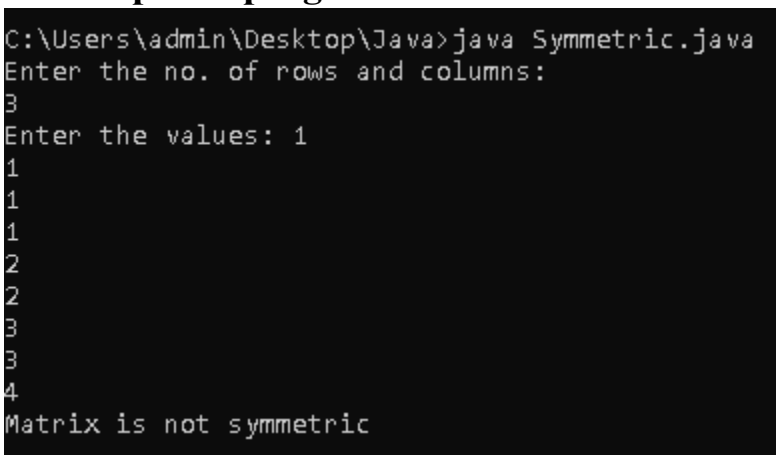
Program 1:

To check if the entered matrix is symmetric or not.

CODE:

```
import java.util.Scanner;

class symmetric{

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

          for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                  if(arr[i][j] != arr[j][i])
                    return 0;
            }
          }
           return 1;
        }

        public static void main(String[] args){

          Scanner in = new Scanner(System.in);

          System.out.println("Enter the no. of rows and columns: ");
          int n = in.nextInt();
          int[][] arr = new int[10][10];

          System.out.print("Enter the values: ");

          for(int i=0;i<n;i++){
            for(int j=0;j<n;j++){
                  arr[i][j] = in.nextInt();
            }
          }
          int flag = sym(arr,n);

          if(flag == 1)
                  System.out.println("Matrix is symmetric");
          else
                  System.out.println("Matrix is not symmetric");
        }
}
```

The output of program 1:



Program 2:

To perform Matrix Multiplication.

CODE:

```
import java.io.*;

class MatrixMultiplication {
    static void printMatrix(int M[][],
                            int rowSize,
                            int colSize)
    {
        for (int i = 0; i < rowSize; i++) {
            for (int j = 0; j < colSize; j++)
                System.out.print(M[i][j] + " ");

            System.out.println();
        }
    }
    static void multiplyMatrix(
        int row1, int col1, int A[][],
        int row2, int col2, int B[][])
    {
        int i, j, k;

        System.out.println("\nMatrix A:");
        printMatrix(A, row1, col1);
        System.out.println("\nMatrix B:");
        printMatrix(B, row2, col2);

        if (row2 != col1) {
            System.out.println(
                "\nMultiplication Not Possible");
            return;
        }
```

```
        int C[][] = new int[row1][col2];
        for (i = 0; i < row1; i++) {
            for (j = 0; j < col2; j++) {
                for (k = 0; k < row2; k++)
                    C[i][j] += A[i][k] * B[k][j];
            }
        }

        System.out.println("\nResultant Matrix:");
        printMatrix(C, row1, col2);
    }

    public static void main(String[] args)
    {

        int row1 = 4, col1 = 3, row2 = 3, col2 = 4;

        int A[][] = { { 1, 1, 1 },
                      { 2, 2, 2 },
                      { 3, 3, 3 },
                      { 4, 4, 4 } };

        int B[][] = { { 1, 1, 1, 1 },
                      { 2, 2, 2, 2 },
                      { 3, 3, 3, 3 } };

        multiplyMatrix(row1, col1, A,
                       row2, col2, B);
    }
}
```

The output of program 2:

```
C:\Users\admin\Desktop\Java>javac MatrixMultiply.java

C:\Users\admin\Desktop\Java>java MatrixMultiply

Matrix A:
1 1 1
2 2 2
3 3 3
4 4 4

Matrix B:
1 1 1 1
2 2 2 2
3 3 3 3

Resultant Matrix:
6 6 6 6
12 12 12 12
18 18 18 18
24 24 24 24
```

Program 3:

Reverse the string and decide whether it is palindrome or not and Capitalize the string.

CODE:

```
class PalindromeStrings
{
	public static int fun(String str)
	{
		int n = str.length()-1;
		for(int i=0;i<n/2;i++)
		{
			if(str.charAt(i) != str.charAt(n-i))
				return 0;
		}

		return 1;
	}

	public static void main(String[] args)
	{

		Scanner in = new Scanner(System.in);

		System.out.print("Enter a string: ");
		String str = in.next();

		int flag = fun(str);

		if(flag == 1)
			System.out.println(str+ " is a palindrome");
		else
			System.out.println(str+ " is not a palindrome");

		System.out.println("Capitalized string is "+ str.toUpperCase());
	}
}
```

The output of program 3:

```
C:\Users\admin\Desktop\Java>javac PalindromeStrings.java

C:\Users\admin\Desktop\Java>java PalindromeStrings
Enter a string: madam
madam is a palindrome
Capitalized string is MADAM

C:\Users\admin\Desktop\Java>java PalindromeStrings
Enter a string: sir
sir is not a palindrome
Capitalized string is SIR
```

Program 4:
Program to demonstrate built-in Functions of String Class.

CODE:

```
import java.lang. *;
import java.util.*;
class StringFunctions
{
	public static void main(String[] args)
	{
		Scanner in=new Scanner(System.in);			//Scanner Initialization

		System.out.print("\n\n Input the first string :");		//Vadapav
		String str = in.next();

		System.out.print("\n\n Input the second string : ");	//Katchori
		String str2 = in.next();

		System.out.print("\n Using equals() we get: ");					//1
		System.out.println(str.equals(str2));

		System.out.print("\n Using equalsIgnoreCase() we get: ");			//2
		System.out.println(str.equalsIgnoreCase(str2));

		String str3=str.concat(str2);
		System.out.print("\n Using concat() we get: ");					//3
		System.out.println(str3);

		System.out.print("\n Using compareTo() we get: ");				//4
		System.out.println(str.compareTo(str2));

		System.out.print("\n Using compareToIgnoreCase() we get: ");			//5
		System.out.println(str.compareToIgnoreCase(str2));

		System.out.print("\n Using substring(int beginIndex) we get: ");		//6
		System.out.println(str.substring(3));

		System.out.print("\n Using substring(int beginIndex, int endIndex) we get: ");	//7
		System.out.println(str.substring(3,6));

		System.out.print("\n Using toUpperCase() we get: ");				//8
		System.out.println(str.toUpperCase());
```

```
        System.out.print("\n Using to LowerCase() we get: ");                    //9
        System.out.println(str.toLowerCase());

        System.out.print("\n Using trim(): ");                                   //10
        System.out.println(str.trim());

        System.out.print("\n The length of string is: ");                        //11
        System.out.println(str.length());

        System.out.print("\n Using charAt(int index): ");                        //12
        System.out.println(str.charAt(2));
    }
}
```

The output of program 4:

```
C:\Users\admin\Desktop\Java>javac StringFunctions.java

C:\Users\admin\Desktop\Java>java StringFunctions

 Input the first string :Vadapav

 Input the second string : Katchori

 Using equals() we get: false

 Using equalsIgnoreCase() we get: false

 Using concat() we get: VadapavKatchori

 Using compareTo() we get: 11

 Using compareToIgnoreCase() we get: 11

 Using substring(int beginIndex) we get: apav

 Using substring(int beginIndex, int endIndex) we get: apa

 Using toUpperCase() we get: VADAPAV

 Using to LowerCase() we get: vadapav

 Using trim(): Vadapav

 The length of string is: 7

 Using charAt(int index): d

C:\Users\admin\Desktop\Java>
```
