קורס: מבוא למדעי המחשב בשפת Java

סטודנטית 1: דליה ויליאם

סטודנט 2: גיא רחמים

Assignment 8:

//Dalya William & Guy Rahamim

//Assignment 8

**import** java.util.Scanner;

**public** **class** Assignment8

{

//initializing a global scanner.

**public** **static** Scanner *input* = **new** Scanner (System.***in***);

**public** **static** **void** main(String[] args)

{

//--------------------------------------------Question 2----------------------------------------------

//initializing variables

**int** size=4;

**int**[] array1;

**int**[] array2;

//initializing array1 and array2 using initializeArray function.

System.***out***.println("Please enter values for array number 1:");

array1= *initializeArray*(size);

System.***out***.println("Please enter values for array number 2:");

array2= *initializeArray*(size);

//print both arrays using printArray function.

System.***out***.println("The arrays in question are: \n");

*printArray*(array1);

System.***out***.println();

*printArray*(array2);

//decides if the arrays form a mirror pair using mirrorPair.

System.***out***.println("\nDo these two arrays form a mirror pair? the answer is "

+*mirrorPair*(array1,array2));

//--------------------------------------------Question 3----------------------------------------------

System.***out***.println("\n\n");

//initializing variables.

**boolean** rowColChecker=**false**;

**int** rows,cols;

**int** fullArraysizes=6;

**char** max;

**char** [][] charArray = **new** **char** [fullArraysizes][fullArraysizes], tempArray;

System.***out***.println("Please enter how many rows and coloumns "

+ "would you like to use from the available 6 : ");

**do**

{

rows=*input*.nextInt();

cols=*input*.nextInt();

**if** (rows >=1 && cols >=1 && rows <=6 && cols <=6)

rowColChecker=**true**;

**else**

System.***out***.println("Bad input! please make sure both numbers are in the range of 1 - 6");

}

**while** (!rowColChecker);

//initializing charArray with user input.

charArray=*initialize2DCharArray*(charArray,rows,cols);

//cropping the 6X6 array and storing the biggest char

//value in the array.

tempArray=*cropCharArray*(charArray,rows,cols);

max = *maxCharInArray*(tempArray);

//print the entire 6X6 array and the biggest char value.

*print2DCharArray*(charArray);

System.***out***.println("\n the biggest character in the array is: " +max);

//--------------------------------------------Question 4----------------------------------------------//

//initialize variables

**int** rowSize=5,colSize=4;

**int**[][] array;

//initialize array with function.

array=*initialize2DArray*(rowSize,colSize);

//print the array using a function.

//and then determine if it is positive using the positiveArray function.

*print2DArray*(array);

System.***out***.println("is the array positive? the answer is " + *positiveArray*(array));

*input*.close();

}

//-------------!!! FUNCTIONS FOR INTEGER ARRAYS !!! ----------------//

//a function that prints an array.

**public** **static** **void** printArray (**int**[] array)

{

**for** (**int** i=0; i<array.length;i++)

{

System.***out***.print(array[i]+ "\t");

}

}

//a function that prints a 2D array.

**public** **static** **void** print2DArray (**int**[][] array)

{

**for** (**int** i=0; i<array.length;i++)

{

*printArray*(array[i]);

System.***out***.println();

}

}

//a function that initializes an array.

**public** **static** **int**[] initializeArray (**int** arraySize)

{

**int** [] array = **new** **int** [arraySize];

**for** (**int** i=0; i<array.length;i++)

{

System.***out***.println("Please enter a number to store in array cell number " +i);

array[i]=*input*.nextInt();

}

**return** array;

}

//a function that initializes a 2D array

**public** **static** **int**[][] initialize2DArray (**int** rowNum, **int** colNum)

{

**int**[][] array = **new** **int** [rowNum][colNum];

**for** (**int** i=0; i<array.length;i++)

{

array[i]=*initializeArray*(colNum);

}

**return** array;

}

//-------------!!! FUNCTIONS FOR QUESTIONS 2,3 !!! ----------------//

//a functions that determines if 2 arrays are mirroring each other.

**public** **static** **boolean** mirrorPair(**int**[] array1, **int**[] array2)

{

**for** (**int** i = 0; i<array1.length;i++)

{

**if** (array1[i]!=array2[array2.length-i-1])

**return** **false**;

}

**return** **true**;

}

//a function that finds the biggest char among the

//outer frame of a given 2D array.

**public** **static** **char** maxCharInArray(**char** [][]array)

{

**char** max= array[0][0];

**for** (**int** i=0; i<array.length;i++)

{

**for** (**int** j=0; j<array.length;j++)

{

**if** (i==0 || i== array[i].length-1 || j==0 || j==array.length-1)

{

System.***out***.println(array.length);

System.***out***.println("location: "+ i +" , " +j + " comparing " + max +" with " +array[i][j]);

max = max>array[i][j] ? max : array[i][j];

}

}

}

**return** max;

}

//a function that crops a small 2D array out of a larger

//2D character array.

**public** **static** **char**[][] cropCharArray(**char**[][] arrayToCrop, **int** rowNum, **int** colNum)

{

**char**[][] array= **new** **char**[rowNum+1][colNum+1];

**for** (**int** i = 0; i < array.length; i++)

{

**for** (**int** j = 0; j < array.length; j++)

{

array[i][j]=arrayToCrop[i][j];

}

}

**return** array;

}

//a function that prints an array of characters.

**public** **static** **void** printCharArray (**char**[] array)

{

**for** (**int** i=0; i<array.length;i++)

{

System.***out***.print(array[i]+ "\t");

}

}

//a function that prints a 2D array of characters.

**public** **static** **void** print2DCharArray (**char**[][] array)

{

**for** (**int** i=0; i<array.length;i++)

{

*printCharArray*(array[i]);

System.***out***.println("");

}

}

//a function that initializes a 2D array of characters.

**public** **static** **char**[][] initialize2DCharArray(**char**[][] charArray, **int** rows, **int** cols)

{

**char** exclemation='!';

**for** (**int** i=0;i<charArray.length;i++)

{

**for** (**int** j=0; j<charArray.length;j++)

{

**if** (i<rows && j<cols)

{

System.***out***.println("Please enter a char to be stored at row: "+i + "coloumn: " +j);

charArray[i][j]=*input*.next().charAt(0);

}

**else**

charArray[i][j]=exclemation;

}

System.***out***.println();

}

**return** charArray;

}

//a function that checks if an array is considered "positive"

**public** **static** **boolean** positiveArray(**int**[][] array)

{

**int** rowSum=0, colSum=0;

**for** (**int** row = 0; row < array.length; row++)

{

**for** (**int** col = 0; col < array[row].length; col++)

{

**if** (col%2!=0)

{

colSum+=array[row][col];

}

**if** ((row%2)==0)

{

rowSum+=array[row][col];

}

}

}

**if** (colSum>rowSum)

**return** **true**;

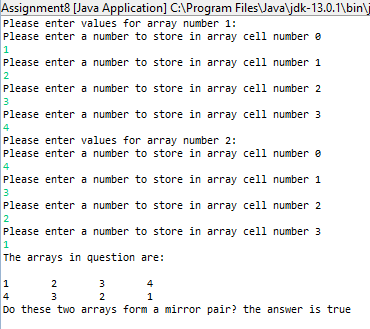
**else**

**return** **false**;

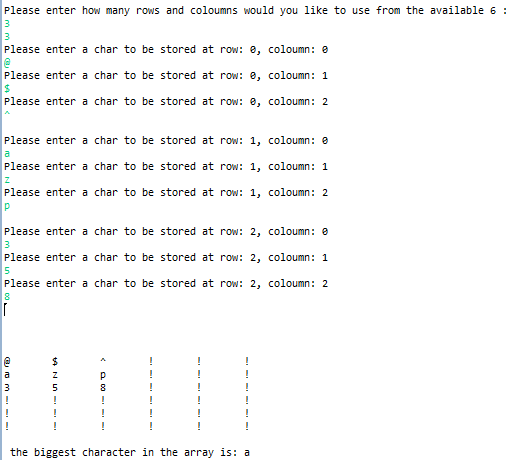
}

}

Output for question 2:



Output for question 3:



Output for question 4:

