**ASSIGNMENT-3**

**NAME : SANJOG PATEL**

**EMPLOYEE ID: 2607661**

**BATCH : 240**

**Problem Statement 1: Arrays API**

1.**Write a program that takes a String through Command Line argument and display the length of the string. Also display the string into uppercase and check whether it is a palindrome or not. (Refer Java API Documentation.)**

**package** day3

**public** **class** StringArgumentLength {

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

**if**(args.length!=1){

System.**out**.println("provide a single string as a command line argument.");

**return**;

}

String input = args[0];

**int** length =input.length();

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

String uppercaseString = input.toUpperCase();

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

**boolean** isPalindrome =isPalindrome(input);

**if**(isPalindrome){

System.**out**.println("the string is a palindrome");

}**else**{

System.**out**.println("the string is not a palindrome");

}

}

**private** **static** **boolean** isPalindrome(String input){

**int** left =0;

**int** right = input.length()-1;

**while**(left<right){

**if**(input.charAt(left)!= input.charAt(right)){

**return** **false**;

}

left ++;

right--;

}

**return** **true**;

}

}

2.**Write a program that accepts two numbers in the range from 1 to 40 from the Command Line. Then compares these numbers against a single dimension array of five integer elements ranging in value from 1 to 40. The program displays the message “Bingo” if the two inputted values are found in the array element**

**package** day3;

**public** **class** numbers {

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

**if**(args.length!=2){

System.***out***.println("provide a two numbers as a command line arguments");

**return**;

}

**int** firstnumber =Integer.*parseInt*(args[0]);

**int** secondnumber = Integer.*parseInt*(args[1]);

**if**(firstnumber<1 || firstnumber>40 || secondnumber<1 || secondnumber>40){

System.***out***.println("both the number must be the range of 1 to 40.");

**return**;

}

**int**[] array={7,30,25,6,18};

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

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

**boolean** firstnumberFound = **false** ;

**boolean** secondnumberFound = **false** ;

**for**(**int** number: array){

**if**(number==firstnumber){

firstnumberFound = **true** ;

}

**if**(number == secondnumber){

secondnumberFound = **true** ;

}

}

**if**(firstnumberFound&&secondnumberFound){

System.***out***.println("it's bingo!");

}**else**{

System.***out***.println("not found");

}

System.***out***.print("the array was");

**for**(**int** number:array){

System.***out***.print(number+ "");

}

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

}

}

3.**Write a program that allows you to create an integer array of 18 elements with the following values: int A[] = {3, 2, 4, 5, 6, 4, 5, 7, 3, 2, 3, 4, 7, 1, 2, 0, 0, 0}. Perform the following computations,**

**• Compute the sum of elements from index 0 to 14 and stores it at element 15.**

**• Compute the average of all numbers and stores it at element 16.**

**• Identifies the smallest value from the array and stores it at element 1**

**package** day3;

**public** **class** Integerarray {

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

**int**[]A ={3,2,4,5,6,4,5,7,3,2,3,4,7,1,2,0,0,0};

**int** sum =0;

**for**(**int** i=0; i<=1;i++){

sum+=A[i];

}

A[15]=sum;

**int** totalsum=0;

//Including the newly computed sum at index 15

**for**(**int** i=0;i<=15;i++){

totalsum+=A[i];

}

**double** average =(**double**) totalsum/15;

A[16]=(**int**)average;

**int** smallest=A[0];

**for**(**int** i=0;i<=15;i++){

**if**(A[i]<smallest){

smallest =A[i];

}

}

A[17]=smallest;

System.***out***.println("updated array:");

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

System.***out***.print(A[i]+"");

}

}

}

4.**Jaffer was done with the online shopping, and the next step was to complete the bill payment, for which he must provide his name, credit card number, expiry month/year and CVV code. The system should validate the inputs as mentioned below:**

**• Inputs cannot be empty if so, provide appropriate messages.**

**• Credit CardNumber, expiry month/year and CVV code must be numeric.**

**• Credit card should be 16 characters long and include a dash (-) after each four-digitnumber.**

**• CVV Code should be of length 3.**

**• The month and year in which the cardwill expire should be displayed as MM/YY**

**package day3;**

**import java.util.Scanner;**

**import java.util.regex.Pattern;**

**import java.util.regex.Matcher;**

**public class Paymentvalidation {**

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

**Scanner scanner = new Scanner(System.*in*);**

**System.*out*.print("Enter Your Name: ");**

**String name = scanner.nextLine();**

**System.*out*.print("Enter Card Number: ");**

**String cardNumber = scanner.nextLine();**

**System.*out*.print("Enter Expiry Date MMYY: ");**

**String expiryDate = scanner.nextLine();**

**System.*out*.print("Enter CVV Code: ");**

**String cvvCode = scanner.nextLine();**

**if (name.isEmpty()) {**

**System.*out*.println("Name cannot be empty.");**

**} else if (!*isValidCardNumber*(cardNumber)) {**

**System.*out*.println("Invalid card number. It should be 16 digits long and include a dash (-) after each four digits.");**

**} else if (!*isValidExpiryDate*(expiryDate)) {**

**System.*out*.println("Invalid expiry date. It should be in MMYY format.");**

**} else if (!*isValidCVV*(cvvCode)) {**

**System.*out*.println("Invalid CVV code. It should be 3 digits long.");**

**} else {**

**String formattedCardNumber = *formatCardNumber*(cardNumber);**

**String formattedExpiryDate = *formatExpiryDate*(expiryDate);**

**System.*out*.println("\nDetails Entered are");**

**System.*out*.println("Name: " + name);**

**System.*out*.println("Card Number: " + formattedCardNumber);**

**System.*out*.println("Expiry Date: " + formattedExpiryDate);**

**System.*out*.println("CVV Code: " + cvvCode);**

**}**

**}**

**private static boolean isValidCardNumber(String cardNumber) {**

**String regex = "\\d{16}";**

**Pattern pattern = Pattern.*compile*(regex);**

**Matcher matcher = pattern.matcher(cardNumber);**

**return matcher.matches();**

**}**

**private static String formatCardNumber(String cardNumber) {**

**return cardNumber.replaceAll("(.{4})", "$1-").substring(0, 19);**

**}**

**private static boolean isValidExpiryDate(String expiryDate) {**

**String regex = "^(0[1-9]|1[0-2])\\d{2}$"; // MMYY format with MM between 01 and 12**

**Pattern pattern = Pattern.*compile*(regex);**

**Matcher matcher = pattern.matcher(expiryDate);**

**return matcher.matches();**

**}**

**private static String formatExpiryDate(String expiryDate) {**

**return expiryDate.substring(0, 2) + "/" + expiryDate.substring(2, 4);**

**}**

**private static boolean isValidCVV(String cvvCode) {**

**String regex = "\\d{3}";**

**Pattern pattern = Pattern.*compile*(regex);**

**Matcher matcher = pattern.matcher(cvvCode);**

**return matcher.matches();**

**}**

**}**

5.**Create an interface MedicineInfo to represent a drug manufactured by a pharmaceutical company. Provide an abstract method displayLabel() in this interface to print Name and address of the company.**

**Do following tasks,**

**• Implement MedicineInfo interface with Tablet, Syrup and Ointment classes.**

**• Override the displayLabel() method in each of these classes to print information suitable to the type of medicine. For example, in case of tablets, it could be “store in a cool dry place”, in case of ointments it could be “for external use only” etc. Create a class TestMedicine . Write main function to do the following:**

**• Declare an array of Medicine references of size 10.**

**• Create a medicine object of the type as decided by a randomly generated integer in the range 1 to 3.(Refer Java API Documentation to find out random generation feature.)**

**• Check the polymorphic behavior of the displayLabel() method.**

**package** day3;

**public** **interface** MedicineInfo {

**void** displayLabel();

}

**package** day;

**public** **class** tablet **implements** MedicineInformation {

@Override

**public** **void** displayLabel(){

System.***out***.println("Store in a cool dry place.");

}

}

**package** day3;

**public** **class** syrup **implements** MedicineInformation {

@Override

**public** **void** displayLabel(){

System.***out***.println("Shake well before use.");

}

}

**package** day3;

**public** **class** ointement **implements** MedicineInformation {

@Override

**public** **void** displayLabel(){

System.***out***.println("for external use only.");

}

}

**package** day3;

**import** java.util.Random;

**public** **class** testMedicine {

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

MedicineInformation[] medicines =**new** MedicineInformation[10];

Random rand =**new** Random();

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

**int** type =rand.nextInt(3)+ 1;

**switch**(type){

**case** 1: medicines[i]=**new** tablet();

**break**;

**case** 2: medicines[i]=**new** syrup();

**break**;

**case** 3: medicines[i]=**new** ointement();

**break**;

}

}

**for**(MedicineInformation medicine:medicines){

medicine.displayLabel();

}

}

}

6.**Sort the given array in descending order, i.e., arrange the elements from largest to smallest.**

**Input: Array = {2, 6, 23, 98, 24, 35, 78}**

**Output: [98, 78, 35, 24, 23, 6, 2]**

Input: Array = {1, 2, 3, 4, 5}

Output: [5, 4, 3, 2, 1]

**package** day3;

**import** java.util.Arrays;

**import** java.util.Collections;

**import** java.util.Scanner;

**public** **class** descendingorder {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of elements: ");

**int** n = sc.nextInt();

Integer[] array = **new** Integer[n];

System.***out***.println("Enter " + n + " integers:");

**for** (**int** i = 0; i < n; i++) {

array[i] = sc.nextInt();

}

Arrays.*sort*(array, Collections.*reverseOrder*());

System.***out***.println("Array sorted in descending order:");

**for** (**int** num : array) {

System.***out***.print(num + " ");

}

sc.close();

}

}

7.**Given a sorted array arr[] of size N, the task is to remove the duplicate elements from the array.**

**Input: arr[] = {2, 2, 2, 2, 2}**

**Output: arr[] = {2}**

**Input: arr[] = {1, 2, 2, 3, 4, 4, 4, 5, 5}**

**Output: arr[] = {1, 2, 3, 4, 5]**

**package** day3;

**import** java.util.Scanner;

**import** java.util.Arrays;

**public** **class** Removeduplicateelements {

**public** **static** **int**[]removeduplicates(**int**[]arr){

**if**(arr==**null** || arr.length==0){

**return** arr;

}

**int** i=0;

**for**(**int** j=1;j<arr.length;j++){

**if**(arr[j]!=arr[i]){

i++;

arr[i]=arr[j];

}

}

**int**[] result =**new** **int**[i+1];

System.*arraycopy*(arr, 0, result, 0, i+1);

**return** result;

}

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

Scanner sc =**new** Scanner(System.***in***);

System.***out***.print("enter the number of elements:");

**int** n =sc.nextInt();

**if**(n<=0){

System.***out***.println("invalid number of elements");

**return**;

}

**int**[]arr=**new** **int**[n];

System.***out***.println("enter the sorted elements:");

**for**(**int** i=0;i<n;i++){

arr[i]=sc.nextInt();

}

sc.close();

**int**[] uniqueArr= *removeduplicates*(arr);

System.***out***.println("unique elements:");

**for**(**int** num:uniqueArr){

System.***out***.print(num +"");

}

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

}

)

8.**Given two arrays and our task is to find their common elements.**

**Input: Array1 = [“Article”, “for”, “Apple”, “for”, “Grapes”],**

**Array2 = [“Article”, “Apple”, “Grape”]**

**Output: [Article, Apple]**

**Input: Array1 = [“a”, “b”, “c”, “d”, “e”, “f”],**

**Array2 = [“b”, “d”, “e”, “h”, “g”, “c”]**

**Output: [b, c, d, e]**

**package** day3;

**import** java.util.Scanner;

**import** java.util.ArrayList;

**public** **class** CommonelementsInArrays {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter the number of elements in Array1: ");

**int** size1 = sc.nextInt();

sc.nextLine();

System.***out***.print("Enter the number of elements in Array2: ");

**int** size2 = sc.nextInt();

sc.nextLine();

String[] array1 = **new** String[size1];

System.***out***.println("Enter the elements of Array1:");

**for** (**int** i = 0; i < size1; i++) {

array1[i] = sc.nextLine();

}

String[] array2 = **new** String[size2];

System.***out***.println("Enter the elements of Array2:");

**for** (**int** i = 0; i < size2; i++) {

array2[i] = sc.nextLine();

}

ArrayList<String> commonElements = **new** ArrayList<>();

**for** (**int** i = 0; i < size1; i++) {

**for** (**int** j = 0; j < size2; j++) {

**if** (array1[i].equals(array2[j]) && !commonElements.contains(array1[i])) {

commonElements.add(array1[i]);

**break**;

}

}

}

System.***out***.println("Common elements: " + commonElements);

sc.close();

}

}

9.**Given a 2D array of order N X M and a column number K ( 1<=K<=m). Our task is to sort the 2D array according to values in Column K.**

**Input: 2D array is given as (Order 4X4)**

**39 27 11 42**

**10 93 91 90**

**54 78 56 89**

**24 64 20 65**

**Output:**

**39 27 11 42**

**24 64 20 65**

**54 78 56 89**

**10 93 91 90**

**package** day3;

**import** java.util.Scanner;

**import** java.util.Arrays;

**import** java.util.Comparator;

**public** **class** SortingColumnNumber {

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

Scanner sc = **new** Scanner(System.***in***);

**int** rows =4;

**int** cols =4;

**int**[][] array =**new** **int**[rows][cols];

System.***out***.println("enter the elements of 2D array:");

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

array[i][j]=sc.nextInt()-1;

}

}

System.***out***.println("enter the column number:");

**int** columnK =sc.nextInt()-1;

Arrays.*sort*(array, **new** Comparator<**int**[]>(){

**public** **int** compare(**int**[]row1,**int**[]row2){

**return** Integer.*compare*(row1[columnK],row2[columnK]);

}

});

System.***out***.print("Array sorted by column:");

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

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

}

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

}

sc.close();

}

}

10.**Write the Java Program to Find Transpose of Matrix given matrix.**

**package** day3;

**import** java.util.Scanner;

**public** **class** TransposeMatrix {

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

Scanner sc =**new** Scanner(System.***in***);

System.***out***.print("enter the no.of rows:");

**int** rows =sc.nextInt();

System.***out***.print("enter the no.of columns:");

**int** cols =sc.nextInt();

**int**[][]matrix= **new** **int**[rows][cols];

System.***out***.println("enter the elements of the matrix:");

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

matrix[i][j]=sc.nextInt();

}

}

**int**[][]transpose = *transposeMatrix*(matrix,rows,cols);

System.***out***.println("transpose of the matrix");

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

System.***out***.print(transpose[i][j] +"");

}

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

}

sc.close();

}

**public** **static** **int**[][]transposeMatrix(**int**[][]matrix,**int** rows, **int** cols){

**int**[][]transpose=**new** **int**[cols][rows];

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

transpose[j][i]=matrix[i][j];

}

}

**return** transpose;

}

}

11.**Given a matrix of size Row x Col Print the boundary elements of the matrix. Boundary elements are those elements which are not surrounded by elements on all four directions, i.e. elements in the first row, first column, last row and last column.**

**Input :**

**1 2 3**

**4 5 6**

**7 8 9**

**Output:**

**1 2 3**

**4 6**

**7 8 9**

**package** day3;

**public** **class** PrintingBoundaryElements {

**public** **static** **void** printBoundaryElements(**int**[][]matrix){

**if**(matrix==**null**||matrix.length==0)

**return**;

**int** rows=matrix.length;

**int** cols=matrix[0].length;

**for**(**int** i=0;i<rows;i++){

**for**(**int** j=0;j<cols;j++){

**if**(i==0||i==rows-1||j==0||j==cols-1){

System.***out***.print(matrix[i][j] +"");

}**else**{

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

}

}

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

}

}

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

**int**[][]matrix={ {1,2,3},{4,5,6} ,{7,8,9}};

*printBoundaryElements*(matrix);

}

}

**12.For a given 2D square matrix of size N\*N, the task is to find the sum of elements in the Principal and Secondary diagonals.**

**Input 1:**

**6 7 3 4**

**8 9 2 1**

**1 2 9 6**

**6 5 7 2**

**Output 1: Principal Diagonal: 26**

**Secondary Diagonal: 14**

**package** day3;

**public** **class** PrincipalAndSecondaryDiagonal {

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

**int**[][] matrix={

{6,7,3,4},

{8,9,2,1},

{1,2,9,6},

{6,5,7,2},

};

**int** principalDiagonalSum=0;

**int** secondaryDiagonalSum=0;

**int** N =matrix.length;

**for**(**int** i=0;i<N;i++){

principalDiagonalSum+=matrix[i][i];

secondaryDiagonalSum+=matrix[i][N-i-1];

}

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

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

}

}

**Problem Statement 2: String Classes**

13**.Write a program that takes String as input and count the last ‘n’ vowels of a given String. If the**

**number is greater than the vowels found, then print ‘Mismatch in Vowel Count**

**Input:**

**Testing**

**2**

**Output:**

**ei**

**package** day3;

**import** java.util.Scanner;

**import** java.util.List;

**import** java.util.ArrayList;

**public** **class** LastNVowles {

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

Scanner sc=**new** Scanner(System.***in***);

System.***out***.print("enter a string:");

String inputString =sc.nextLine();

System.***out***.print("enter the number of last vowels to extract:");

**int** n=sc.nextInt();

String result= *getLastNVowels*(inputString,n);

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

sc.close();

}

**public** **static** String getLastNVowels(String inputString,**int** n){

List<Character>vowels =**new** ArrayList<>();

String vowelsList="aeiouAEIOU";

**for**(**int** i =inputString.length()-1;i>=0;i--){

**char** currentChar =inputString.charAt(i);

**if**(vowelsList.indexOf(currentChar)!=-1){

vowels.add(currentChar);

}

}

**if**(vowels.size()<n){

**return**"mismatch in vowel count";

}

StringBuilder lastNVowels =**new** StringBuilder();

**for**(**int** i=0;i<n;i++){

lastNVowels.insert(0,vowels.get(i));

}

**return** lastNVowels.toString();

}

}

**14.Write a Java Program to reverse a string.**

**Input:**

**S = “abc”**

**Output:**

**S = “cba”**

**package** day3;

**import** java.util.Scanner;

**public** **class** ReverseString {

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

Scanner sc =**new** Scanner(System.***in***);

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

String input =sc.nextLine();

String reversestring = **new** StringBuilder(input).reverse().toString();

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

sc.close();

}

}

**15.Write the Java Program to Sort a String.**

Input string :

"JavaProgram"

Output string :

"aaagjmoprrv”

**package** day3;

**import** java.util.Arrays;

**import** java.util.Scanner;

**public** **class** sortingstring {

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

Scanner sc=**new** Scanner(System.***in***);

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

String input =sc.nextLine();

**char**[] charArray=input.toCharArray();

Arrays.*sort*(charArray);

String sortedString =**new** String(charArray);

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

sc.close();

}

}

16.**Given string str, the task is to write a Java program to swap the pairs of characters of a string. If the string contains an odd number of characters, then the last character remains as it is.**

**Input: str = “Java”**

**Output: aJav**

**Input: str = “Testing”**

**Output: eTtsnig**

**package** day3;

**import** java.util.Scanner;

**public** **class** swapsorting{

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

Scanner sc=**new** Scanner(System.***in***);

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

String input =sc.nextLine();

sc.close();

String result = *swapsort*(input);

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

}

**public** **static** String swapsort(String str){

**char**[] chars =str.toCharArray();

**int** length =chars.length;

**for**(**int** i=0;i<length-1;i+=2){

**char** temp =chars[i];

chars[i]=chars[i+1];

chars[i+1]=temp;

}

**return** **new** String(chars);

}

}

**17.Given string str, the task is to write Java Program check whether the given string is a pangram or not.A string is a pangram string if it contains all the character of the alphabets ignoring the case of the alphabets.**

**Input: str = “Abcdefghijklmnopqrstuvwxyz”**

**Output: Yes**

**Input: str = “welcome”**

**Output: No**

**package** day3;

**import** java.util.Scanner;

**public** **class** Panogram {

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

Scanner sc= **new** Scanner(System.***in***);

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

String input = sc.nextLine();

**if**(*isPangram*(input)){

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

}**else**{

System.***out***.println("no");

}

}

**public** **static** **boolean** isPangram(String str){

str =str.toLowerCase().replaceAll("[^a-z]","");

**return**

str.chars().distinct().count()==26;

}

}

18. **Write the java program to extract the first letter of each word in it.**

**Input : Practice Java Program**

**Output : PJP**

**Input : United Kingdom**

**Output : Uk**

**package** day3;

**import** java.util.Scanner;

**public** **class** firstletter {

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

Scanner sc =**new** Scanner(System.***in***);

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

String input = sc.nextLine();

sc.close();

String result =*extractfirstletter*(input);

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

}

**public** **static** String extractfirstletter(String str){

String[]words=str.split("\\s+");

StringBuilder result =**new** StringBuilder();

**for**(String word :words){

**if**(!word.isEmpty()){

result.append(word.charAt(0));

}

}

**return** result.toString();

}

}

19.**Given a String, the task is to insert another string in between the given String at a particular specified index in Java.**

**Input: originalString = "Computer Portal",**

**stringToBeInserted = "Science ",**

**index = 8**

**Output: "Computer Science Portal"**

package day3;

import java.util.Scanner;

public class InsertinString {

public static void main(String[]args){

Scanner sc = new Scanner(System.*in*);

System.*out*.print("enter the original string: ");

String originalString = sc.nextLine();

System.*out*.print("enter the string to be inserted: ");

String stringToBeInserted = sc.nextLine();

System.*out*.print("enter the index: ");

int index = sc.nextInt();

sc.close();

String result = *insertStringAt*(originalString,stringToBeInserted,index);

System.*out*.println("output:" +result);

}

public static String insertStringAt(String original,String toInsert,int index) {

StringBuilder sb = new StringBuilder(original);

sb.insert(index, toInsert + " ");

return sb.toString();

}

}

20.**Write a Java program to print Even length words in a String.**

**Input: s = "This is a java language"**

**Output: This**

**is**

**java**

**language**

**Input: s = "i am Tester"**

**Output: am**

Tester

**package** day3;

**import** java.util.Scanner;

**public** **class** EvenlengthFinding {

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

Scanner sc = **new** Scanner(System.***in***);

System.***out***.print("Enter a string:");

String input = sc.nextLine();

String result = *getEvenlength*(input);

System.***out***.println("output:"+result);

sc.close();

}

**public** **static** String getEvenlength(String str) {

String[] words = str.split(" ");

StringBuilder result = **new** StringBuilder();

**for**(String word : words) {

**if**(word.length()%2 == 0) {

result.append(word).append(" ");

}

}

**return** result.toString().trim();

}

}