Q1. Write a Java program to find the maximum and minimum element in an array.

**package** Array;

**public** **class** P1 {

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

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

**int** max=arr[0];

**int** min=arr[0];

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

**if**(max<arr[i]) {

max=arr[i];

}

**if**(min>arr[i]) {

min=arr[i];

}

}

System.***out***.println("Minimum element is "+min);

System.***out***.println("Maximum element is "+max);

}

}

Q2. Given an array of integers, find two numbers such that they add up to a specific target number.

**package** Array;

**public** **class** P2 {

**private** **static** **int** *sum*=0;

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

**int**[] array = {3, 2, 4,6};

**int** target = 5;

**int** p=0;

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

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

**if** (array[i] + array[j] == target) {

p = 1;

*sum* = array[i] + array[j];

**break**;

}

}

**if** (p != 0) {

**break**;

}

}

**if** (p != 0) {

System.***out***.println("It reached the target value that is " + *sum*);

} **else** {

System.***out***.println("Not reached the target value");

}

}

}

Q3. Write a function to remove duplicates from a sorted array in-place.

**package** Array;

**public** **class** P3 {

**public** **static** **int** removeDuplicateElements(**int** arr[], **int** n){

**if** (n==0 || n==1){

**return** n;

}

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

**int** j = 0;

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

**if** (arr[i] != arr[i+1]){

temp[j++] = arr[i];

}

}

temp[j++] = arr[n-1];

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

arr[i] = temp[i];

}

**return** j;

}

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

**int** arr[] = {10,20,20,30,30,40,50,50};

**int** length = arr.length;

length = *removeDuplicateElements*(arr, length);

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

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

}

}

Q4. Find the intersection of two arrays.

**package** Array;

**import** java.util.ArrayList;

**import** java.util.Arrays;

**public** **class** P4 {

**public** **static** ArrayList<Integer> intersection(**int**[] arr1, **int**[] arr2) {

ArrayList<Integer> result = **new** ArrayList<>();

**int** i = 0, j = 0;

**while** (i < arr1.length && j < arr2.length) {

**if** (arr1[i] == arr2[j]) {

result.add(arr1[i]);

i++;

j++;

} **else** **if** (arr1[i] < arr2[j]) {

i++;

} **else** {

j++;

}

}

**return** result;

}

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

**int**[] arr1 = {1, 2, 3, 4, 5};

**int**[] arr2 = {3, 4, 5, 6, 7};

ArrayList<Integer> intersection = *intersection*(arr1, arr2);

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

}

}

Q5. Write a program to reverse a string without using built-in functions.

**package** Array;

**public** **class** P5 {

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

String name="JAVA";

String temp="";

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

temp+=name.charAt(i);

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

}

}

Q6. Implement a method to perform basic string compression using the counts of repeated characters.

**package** Array;

**public** **class** P6 {

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

P6 obj = **new** P6();

obj.repeatedCharater();

}

**public** **void** repeatedCharater() {

String name="codecademy";

**char** Arr[]=name.toCharArray();

**int** count=0;

**int** total=0;

**int** repeat=0;

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

count=0;

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

**if**(Arr[j]==Arr[i]) {

count++;

}

}

**if** (count>0) {

total++;

}

}

repeat+=total;

System.***out***.println("The character Repeated at:"+repeat+" Time");

}

}

Q7. Given a string, find the length of the longest substring without repeating characters.

**package** Array;

**public** **class** P7 {

**public** **int** lengthOfLongestSubstring(String s) {

**int** n = s.length();

**int** result = 0;

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

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

**if** (checkRepetition(s, i, j)) {

result = Math.*max*(result, j - i + 1);

}

}

}

**return** result;

}

**private** **boolean** checkRepetition(String s, **int** start, **int** end) {

**int**[] chars = **new** **int**[128];

**for** (**int** i = start; i <= end; i++) {

**char** c = s.charAt(i);

chars[c]++;

**if** (chars[c] > 1) {

**return** **false**;

}

}

**return** **true**;

}

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

P7 ob = **new** P7();

String s = "Hello";

System.***out***.println(ob.lengthOfLongestSubstring(s));

}

}

Q8. Write a program to count the number of vowels and consonants in a given string.

**package** Array;

**public** **class** P8 {

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

**int** vCount = 0, cCount = 0;

String str = "Practice Session";

str = str.toLowerCase();

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

**if**(str.charAt(i) == 'a' || str.charAt(i) == 'e' || str.charAt(i) == 'i' || str.charAt(i) == 'o' || str.charAt(i) == 'u') {

vCount++;

}

**else** **if**(str.charAt(i) >= 'a' && str.charAt(i)<='z') {

cCount++;

}

}

System.***out***.println("Number of vowels: " + vCount);

System.***out***.println("Number of consonants: " + cCount);

}

}

Q9. Write a menu driven program to enter the names of students and delete the students name in the array. Example 1 for adding student 2. For deleting student 3. For exit.

**package** Array;

**import** java.util.Scanner;

**public** **class** P9 {

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

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

String[] students = **new** String[10];

**int** size = 0;

**while** (**true**) {

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

System.***out***.println("1. Add Student");

System.***out***.println("2. Delete Student");

System.***out***.println("3. Exit");

System.***out***.print("Enter your choice: ");

**int** choice = scanner.nextInt();

**switch** (choice) {

**case** 1:

**if** (size < students.length) {

System.***out***.print("Enter the name of the student to add: ");

String newStudent = scanner.next();

students[size++] = newStudent;

System.***out***.println(newStudent + " added successfully.");

} **else** {

System.***out***.println("Array is full. Cannot add more students.");

}

**break**;

**case** 2:

**if** (size > 0) {

System.***out***.print("Enter the name of the student to delete: ");

String studentToDelete = scanner.next();

**boolean** found = **false**;

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

**if** (students[i].equals(studentToDelete)) {

**for** (**int** j = i; j < size - 1; j++) {

students[j] = students[j + 1];

}

size--;

found = **true**;

System.***out***.println(studentToDelete + " deleted successfully.");

**break**;

}

}

**if** (!found) {

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

}

} **else** {

System.***out***.println("Array is empty. Cannot delete students.");

}

**break**;

**case** 3:

System.***out***.println("Exiting program. Goodbye!");

System.*exit*(0);

**break**;

**default**:

System.***out***.println("Invalid choice. Please enter a valid option.");

}

}

}

}

Output:-

Menu:

1. Add Student

2. Delete Student

3. Exit

Enter your choice: 1

Enter the name of the student to add: Anusha

Anusha added successfully.

Menu:

1. Add Student

2. Delete Student

3. Exit

Enter your choice: 2

Enter the name of the student to delete: Anusha

Anusha deleted successfully.

Menu:

1. Add Student

2. Delete Student

3. Exit

Enter your choice: 3

Exiting program. Goodbye!