JAVA ASSIGNMENT

1. How do you swap two numbers without using a third variable in Java?

package assignmenttest;

public class swapnumbers {

public static void main(String[] args) {

int a=1;

int b=2;

a = a + b;

b = a - b;

a = a - b;

System.out.println("The value of a is: "+a);

System.out.println("The value of b is: "+b);

}

}

## \*/ o/p: The value of a is 20;

The value of b is 10;

2.Write a Java program to print a Fibonacci sequence using recursion.

package assignmenttest;

import java.util.Scanner;

public class fibonaccisequence {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the range upto which you need fibonacci series");

int n = sc.nextInt();

int prev = 0;

int present = 1;

if (n==1) {

System.out.println(0);

}

else {

System.out.print("0 1 ");

for (int i = 2; i < n ; i++) {

int next = prev + present;

prev = present;

present = next;

System.out.print(next+" ");

}

}

}

}

## o/p : 0 1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987

/\*3. How do you check if an array of integers contains only odd numbers in Java?

package assignmenttest;

public class oddnumber {

public static void main(String[] args) {

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

System.out.println("ODD NUMBERS: ");

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

if (a[i]%2!=0) {

System.out.println(a[i]);

}

}

System.out.println("EVEN NUMBERS: ");

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

if (a[i]%2==0) {

System.out.println(a[i]);

}

}

}

}

## o/p : Enter the size of the array: 5 Enter the array elements:

Element 1: 1

## Element 2: 3

Element 3: 5

## Element 4: 7

Element 5: 9

The array contains only odd numbers.

/\*4. How do you remove spaces from a string in Java?

package assignmenttest;

import java.util.Scanner;

public class removespace {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string");

String s1 = sc.nextLine();

String s2 = "";

for (int i = 0; i <s1.length(); i++)

{

if(s1.charAt(i)==' ')

{

s1=s1.replaceAll(" ",s2);

}

}

System.out.println(s1);

}

}

## enter the string-- lion

string without white space: : lion

/\*5. How can you find the factorial of an integer in Java?

package assignmenttest;

import java.util.Scanner;

public class factorial {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number to find the factorial");

int n = sc.nextInt();

int fact = 1 ;

if (n==0) {

System.out.println(0);

}

else if (n==1) {

System.out.println(1);

}

else {

for (int i = n; i >=1; i--) {

fact = fact \* i;

}

System.out.println(fact);

}

}

}

## \*/ o/p : enter the number 5

output:120

/\*6. Write a java program to find out and print the longest word present in the sentence.

package assignmenttest;

import java.util.Scanner;

public class longestnumber {

public static void main(String args[])

{

Scanner scanner=new Scanner(System.in);

System.out.println("Enter the string");

String s1=scanner.nextLine();

String small="",large="",word="";

String words[]=new String[100];

int length=0;

s1=s1+"";

for(int i=0;i<s1.length();i++)

{

if(s1.charAt(i)==' ')

{

words[length]=word;

length++;

word="";

}

else

{

word=word+s1.charAt(i);

}

}

small=large=words[0];

for(int k=0;k<length;k++)

{

if(small.length()>words[k].length())

{

small=words[k];

}

if(large.length()<words[k].length())

{

large=words[k];

}

}

System.out.println("Smallest word "+small);

System.out.println("Largest word"+large);

}

}

## \*/ o/p : enter the string i love my country

Longest word in the sentence: country

/\*7. Write the code in the findLongestWord() method which accepts a string and returns the longest word

package assignmenttest;

import java.util.Scanner;

public class longestword {

public static void main(String args[])

{

Scanner scanner=new Scanner(System.in);

System.out.println("Enter the string");

String s1=scanner.nextLine();

String small="",large="",word="";

String words[]=new String[100];

int length=0;

s1=s1+"";

for(int i=0;i<s1.length();i++)

{

if(s1.charAt(i)==' ')

{

words[length]=word;

length++;

word="";

}

else

{

word=word+s1.charAt(i);

}

}

small=large=words[0];

for(int k=0;k<length;k++)

{

if(small.length()>words[k].length())

{

small=words[k];

}

if(large.length()<words[k].length())

{

large=words[k];

}

}

System.out.println("Smallest word "+small);

System.out.println("Largest word"+large);

}

}

## \*/ enter the string I love java

Longest word: love

/\*8. Write a java program to take a string as user input and returns the count of uppercase

letters and lowercase letters.

package assignmenttest;

import java.util.Scanner;

public class uppercase and lowercase{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the sentence of your wish");

String s1 = sc.nextLine();

int Upper = 0;

int Lower = 0;

for (int i = 0; i < s1.length(); i++) {

char ch = s1.charAt(i);

if (ch>='A'&&ch<='Z') {

Upper++;

}

else if (ch>='a'&&ch<='z') {

Lower++;

}

}

System.out.println("The num of Upper case letters are: "+Upper);

System.out.println("The num of Lower case letters are: "+Lower);

}}

/\*9. Given an array of integers and a number k, write a function that returns true if given array

can be divided into pairs such that sum of every pair is divisible by k.

Example :

Input : arr = [9, 5, 7, 3], k = 6

Output: True

Explanation: {(9, 3), (5, 7)} is a

possible solution. 9 + 3 = 12 is divisible

by 6 and 7 + 5 = 12 is also divisible by 6.

package assignmenttest;

public class Launch {

public static void main(String[] args) {

}

}

\*/Output: true

/\*10. Given an unsorted array Arr of size N of positive integers. One number 'A' from set {1,

2,....,N} is missing and one number 'B' occurs twice in array. Find these two numbers.

Example 1 :

Input:N = 2

Arr[] = {2, 2}

Output: 2 1

Explanation: Repeating number is 2 and

smallest positive missing number is 1.

Example 2:

Input:N = 3

Arr[] = {1, 3, 3}

Output: 3 2

Explanation: Repeating number is 3 and

smallest positive missing number is 2.

package assignmenttest;

public class Launch {

public static void main(String[] args) {

}

}

\*/ o/p : Repeating number is: 2

Missing number is: 1

/\*11. Given a string S. The task is to print all unique permutations of the given string in

lexicographically sorted order.

Example1:

Input: ABC

Output:

ABC ACB BAC BCA CAB CBA

Explanation:

Given string ABC has permutations in 6

forms as ABC, ACB, BAC, BCA, CAB and CBA .

Example 2:

Input: ABSG

Output:

ABGS ABSG AGBS AGSB ASBG ASGB BAGS

BASG BGAS BGSA BSAG BSGA GABS GASB

GBAS GBSA GSAB GSBA SABG SAGB SBAG

SBGA SGAB SGBA

Explanation:

Given string ABSG has 24 permutations.

## package com.assignment;

## import java.util.\*;

## public class Launch {

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

## String input = "ABSG";

## generateUniquePermutations(input);

## }

## public static void generateUniquePermutations(String input) {

## char[] chars = input.toCharArray();

## Arrays.sort(chars); // Sort the input string for lexicographic order

## boolean[] used = new boolean[chars.length];

## char[] output = new char[chars.length];

## generatePermutations(chars, output, used, 0);

## }

## public static void generatePermutations(char[] chars, char[] output, boolean[] used, int level) {

## if (level == chars.length) {

## System.out.println(new String(output)); return;

## }

## for (int i = 0; i < chars.length; i++) {

## if (!used[i])

## 

## {

## output[level] = chars[i]; used[i] = true;

## generatePermutations(chars, output, used, level + 1);

## used[i] = false;

## // Skip duplicates while (i < chars.length - 1 && chars[i] == chars[i + 1]) {

## i++;

## }

## }

## }

12. Write a Java Program to Check Leap Year?

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int year;

System.out.println("Enter the Year :");

Scanner sc = new Scanner(System.in);

year = sc.nextInt();

if (((year % 4 == 0) && (year % 100!= 0)) || (year%400 == 0))

System.out.println("Specified year is a leap year");

else

System.out.println("Specified year is not a leap year");

}

}

\*/ o/p : enter the year 2024

leap year

/\*13. Write Java Program to Display Armstrong Number Between Two Intervals?

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int num1, num2;

Scanner sc = new Scanner(System.in);

System.out.println("Enter the first number :");

num1 = sc.nextInt();

System.out.println("Enter the second number :");

num2 = sc.nextInt();

for (int i = num1; i<num2; i++){

int check, rem, sum = 0;

check = i;

while(check != 0) {

rem = check % 10;

sum = sum + (rem \* rem \* rem);

check = check / 10;

}

if(sum == i){

System.out.println(i+" is an Armstrong number.");

}

}

}

}

\*/ o/p: Enter the first number :

1

Enter the second number :

1000

1 is an Armstrong number.

1. is an Armstrong number.

370 is an Armstrong number.

371 is an Armstrong number.

407 is an Armstrong number.

/\*14. Write a program to find the first non-repeating character in a given String?

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a string");

String s1 = sc.next();

for (char i: s1.toCharArray()) {

if (s1.indexOf(i)==s1.lastIndexOf(i)) {

System.out.println("The first non repeating character is: "+i);

break;

}

else {

System.out.println("There is no non repeating character in this string");

break;

}

}

}

}

\*/ Enter a string - cook

The first non repeating character is: c

/\*15. Write a program to remove the character in a string

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string");

String s1 = sc.nextLine();

System.out.println(s1);

System.out.println("Enter the character u want to remove");

String s2 = sc.next();

s1 = s1.replace(s2, "");

System.out.println("The string after removing the character");

System.out.println(s1);

}

}

/\*16. Determine the array is subset of another array

## package assignmenttest;

public class subset {

public static boolean isSubset(int[] arr1, int[] arr2) {

if (arr1.length > arr2.length) {

return false; // arr1 can't be a subset of arr2 if it's longer

}

for (int i = 0; i < arr1.length; i++) {

boolean found = false;

for (int j = 0; j < arr2.length; j++) {

if (arr1[i] == arr2[j]) { found = true;

## break;

}}

if (!found) {

return false; // arr1[i] is not present in arr2

}}

return true; // All elements in arr1 are present in arr2

}

public static void main(String[] args) {

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

int[] arr2 = { 3, 1, 2, 4, 5 };

if (*isSubset*(arr1, arr2)) {

System.*out*.println("arr1 is a subset of arr2");

} else {

System.*out*.println("arr1 is not a subset of arr2");

}}}

## o/p : arr1 is a subset of arr2

/\* 17. Find the Smallest and largest element in an array ?

package assismenttest;

public class launch{

public static void main(String[] args) {

int arr[]= {101,100,12,30,10,1,89,10};

int min=arr[0];

int max=arr[0];

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

if(arr[i]<min) {

min=arr[i];

}

if(arr[i]>max) {

max=arr[i];

}}

System.out.println("smallest num:"+ min); System.out.println("largest num:"+ max);

}}

o/p :smallest num:1 largest num:101

\* 18. Find Second Smallest Element in an Array?

package assignmenttest;

import java.util.Scanner;

public class Ques\_18\_small {

public static void main(String[] args) { Scanner sc = new Scanner(System.in);

System.out.println("enter the array size:");

int n = sc.nextInt();

int a[] =new int[n];

for(int i=0;i<a.length;i++) { System.out.println("enter the number");

a[i]=sc.nextInt();

}

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

for(int j=0; j<a.length; j++ ) { int temp = 0; if(a[i]<a[j]) {

temp=a[i]; a[i]=a[j]; a[j]=temp;

}}}

System.out.println("sorted array");

for(int i=0;i<a.length; i++) { System.out.println(a[i]);

}

System.out.println("second smallest no in the given array:" + a[n-2]);

}}

/\*o/p : enter the array size:

3

enter the number 76

enter the number 77

enter the number 65

sorted array 65

76

77

second smallest no in the given array:76

\* 19. Find Second largest Element in an Array?

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = sc.nextInt();

int [] a = new int [n];

//storing the values

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

System.out.println("Enter the number"+(i+1));

a[i] = sc.nextInt();

}

Arrays.sort(a);

//soritng

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

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

}

System.out.println();

System.out.println("The second largest element is: "+a[n-2]);

System.out.println("The second smallest element is: "+a[1]);

System.out.println("The smallest element is: "+a[0]);

System.out.println("The largest element is: "+a[n-1]);

}

}

\*/ o/p :enter the array size: 3

enter the number 34

enter the number 65

enter the number 78

sorted array 34

65

78

second largest no in the given array:65

/\*20. You are working as a software developer at a weather forecasting company. The

company is developing a new feature for their app that allows users to convert

temperatures from Fahrenheit to Celsius. Your task is to create a method that takes a

temperature in Fahrenheit and converts it to Celsius.

Hint: The formula to convert temperature from Fahrenheit to Celsius is

C = (F - 32) \* 5/9

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Surrounding temperature in Fahrenheit");

double Fahrenheit = sc.nextDouble();

double Celcius = (Fahrenheit-32)\*5/9;

System.out.println("The Surrounding temperature in Degree Celcius is "+Celcius);

}

}

\*//\* Enter the Surrounding temperature in Fahrenheit 98

The Surrounding temperature in Degree Celcius is 36.666666666666664

/\*21. You are creating a student portal for your school. The portal needs to display the average

marks of a student for 8 semesters. Your task is to write a Java function that takes the

marks of each semester individually and returns the average.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number of subjects");

int n = sc.nextInt();

int sum = 0;

int [] sub = new int [n];

//for storing marks

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

System.out.println("Enter the marks of the subject number"+(i+1));

sub[i]=sc.nextInt();

}

//for taking the marks for average

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

sum = sum+sub[i];

}

System.out.println("The average of the all sem is "+(sum/8));

}

}

\*//\* Enter the number of subjects 8

Enter the marks of the subjects 1 90

Enter the marks of the subjects 2 90

Enter the marks of the subjects 3 90

Enter the marks of the subjects 4 90

Enter the marks of the subjects 5 90

Enter the marks of the subjects 6 90

Enter the marks of the subjects 7 90

Enter the marks of the subjects 8 90

The average of the all sem is 90

/\*22. Write a Java program that categorizes people based on their age: Child (0-

12), Teen (13-19), Adult (20-59), Senior (60+)

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter your age");

int age = sc.nextInt();

if (age>=0&&age<=12) {

System.out.println("You are a child");

}

else if (age>=13&&age<=19) {

System.out.println("You are a teen");

}

else if (age>=20&&age<=59) {

System.out.println("You are a adult");

}

else if (age>=60) {

System.out.println("You are a senior");

}

else {

System.out.println("Invalid age");

}

}

}

\*/ Enter your age 67

You are a adult

/\*23. Write a java program to display all even prime numbers?

/\*24. Write a java program to display all odd prime numbers

package assignmenttest;

public class odd\_prime {

public static void main(String[] args) {

int limit = 100; // You can change this limit to any desired range

System.out.println("Odd Prime Numbers up to " + limit + ":");

for (int number = 3; number <= limit; number += 2) {

if (isPrime(number)) { System.out.print(number + " ");

}}}

// Function to check if a number is prime

public static boolean isPrime(int num) {

if (num <= 1) {

return false;

}

if (num == 2) {

return true;

}

if (num % 2 == 0) {

return false;

}

for (int i = 3; i <= Math.sqrt(num); i += 2) {

if (num % i == 0) {

return false;

}

}

return true;

}

}

/\* o/p : Odd Prime Numbers up to 100:

3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97

/\*25. Write a program to find the cube root of a number.

package assignmenttest;

import java.util.\*;

public class launch{

public static void main(String[] args) {

// Create a Scanner object to read input from the user Scanner scanner = new Scanner(System.in);

// Prompt the user to enter a number System.out.print("Enter a number: ");

// Read the number from the user

double number = scanner.nextDouble();

// Calculate the cube root using Math.pow() and display the result double cubeRoot = Math.pow(number, 1.0/3.0); System.out.println("Cube root of " + number + " is: " + cubeRoot);

}

}

Enter a number: 125

Cube root of 125.0 is: 4.999999999999999

/\*26. Write a java program to take a string as user input and returns the count of

uppercase letters and lowercase letters.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the sentence of your wish");

String s1 = sc.nextLine();

int Upper = 0;

int Lower = 0;

for (int i = 0; i < s1.length(); i++) {

char ch = s1.charAt(i);

if (ch>='A'&&ch<='Z') {

Upper++;

}

else if (ch>='a'&&ch<='z') {

Lower++;

}

}

System.out.println("The num of Upper case letters are: "+Upper);

System.out.println("The num of Lower case letters are: "+Lower);

}

}

\*/

/\*27. WAP TO PRINT

\*

\* \*

\* \* \*

\* \* \* \*

\* \* \* \* \*

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc= new Scanner(System.in);

System.out.println("Enter the size: ");

int n=sc.nextInt();

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

//for spces

for (int j = n-1; j >=i; j--) {

System.out.print(" ");

}

//for upper triangle

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

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

}

System.out.println();

}

}

}

/\*28. Write a Program to Print the Hollow Diamond Star Pattern.

\*

\* \*

\* \*

\* \*

\* \*

\* \*

\*

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the pattern you want");

int n = sc.nextInt();

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

for (int j = 0; j < n; j++) {

if (i+j==n/2 || j-i==n/2 || i-j==n/2 || i+j==(n-1)+ (n/2)) {

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

} else {

System.out.print(" ");

}

}

System.out.println();

}

}

}

/\*29. Write a program to find the frequency of each digit in a number.

package assignmenttest;

import java.util.Scanner;

public class frequency {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

long number = scanner.nextLong();

int[] digitFrequency = new int[10]; // Array to store the frequency of each digit (0-9)

// Count the frequency of each digit

while (number > 0) {

int digit = (int) (number % 10); // Extract the last digit digitFrequency[digit]++; // Increment the corresponding frequency counter number /= 10; // Remove the last digit

}

// Display the frequency of each digit System.out.println("Digit Frequency:"); for (int i = 0; i < 10; i++) {

if (digitFrequency[i] > 0) {

System.out.println(i + ": " + digitFrequency[i]);

}

} }

}

o/p : Enter a number: 15 Digit Frequency:

1: 1

5: 1

/\*30. Java Program to Multiply two Matrices by Passing Matrix to a Function

package assignmenttest;

import java.util.Scanner;

public class matrix\_mul {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of rows in the first matrix: ");

int rows1 = scanner.nextInt();

System.out.print("Enter the number of columns in the first matrix: ");

int cols1 = scanner.nextInt();

System.out.print("Enter the number of rows in the second matrix: ");

int rows2 = scanner.nextInt();

System.out.print("Enter the number of columns in the second matrix: ");

int cols2 = scanner.nextInt();

if (cols1 != rows2) {

System.out.println("Matrix multiplication is not possible. Columns of the first matrix must be equal to the rows of the second matrix.");

} else {

int[][] matrix1 = new int[rows1][cols1]; int[][] matrix2 = new int[rows2][cols2];

System.out.println("Enter elements of the first matrix:");

enterMatrixElements(scanner, matrix1);

System.out.println("Enter elements of the second matrix:");

enterMatrixElements(scanner, matrix2);

int[][] resultMatrix = multiplyMatrices(matrix1, matrix2);

System.out.println("Resultant matrix after multiplication:");

printMatrix(resultMatrix);

}

scanner.close();

}

public static void enterMatrixElements(Scanner scanner, int[][] matrix) {

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

for (int j = 0; j < matrix[0].length; j++) { matrix[i][j] = scanner.nextInt();

}}}

public static int[][] multiplyMatrices(int[][] matrix1, int[][] matrix2) {

int rows1 = matrix1.length; int cols1 = matrix1[0].length; int cols2 = matrix2[0].length;

int[][] resultMatrix = new int[rows1][cols2]; for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols2; j++) {

for (int k = 0; k < cols1; k++) {

resultMatrix[i][j] += matrix1[i][k] \* matrix2[k][j];

}}}

return resultMatrix;

}

public static void printMatrix(int[][] matrix) {

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

for (int j = 0; j < matrix[0].length; j++) { System.out.print(matrix[i][j] + " ");

}

System.out.println();

}}}

/\*o/p : Enter the number of rows in the first matrix: 2 Enter the number of columns in the first matrix: 2 Enter the number of rows in the second matrix: 2 Enter the number of columns in the second matrix: 2 Enter elements of the first matrix:

2

2

2

2

Enter elements of the second matrix:

2

2

2

2

Resultant matrix after multiplication:

8 8

8 8 \*/

/\*31. write a program to find duplicate characters in a string

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a string");

String s1 = sc.nextLine();

char [] a = s1.toCharArray();

System.out.println("The duplicate characters in the string are:");

for (int i = 0; i < s1.length(); i++) {

for (int j = i + 1; j < s1.length(); j++) {

if (a[i] == a[j]) {

System.out.print(a[j] + " ");

break;

}

}

}

}

}

\*/ Enter a string i love my country

The duplicate characters in the string are:

o y

/\*32. Given a sorted array of distinct integers and a target value, return the index if

the target is found. If not, return the index where it would be if it were inserted in

order.[Example Input: nums = [1,3,5,6], target = 5

Output: 2]

package assignmenttest;

import java.util.Scanner;

public class index\_val {

public static void main(String[] args) { Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] nums = new int[n];

System.out.println("Enter the sorted array elements:");

for (int i = 0; i < n; i++) { nums[i] = scanner.nextInt();

}

System.out.print("Enter the target value: ");

int target = scanner.nextInt();

int result = searchInsert(nums, target); System.out.println("Output: " + result);

}

public static int searchInsert(int[] nums, int target) {

int left = 0;

int right = nums.length - 1;

while (left <= right) {

int mid = left + (right - left) / 2;

if (nums[mid] == target) {

return mid;

} else if (nums[mid] < target) { left = mid + 1;

} else {

right = mid - 1;

}

}

// If the target is not found, return the index where it should be inserted

return left;

}

}

/\* o/p: Enter the number of elements in the array: 4

Enter the sorted array elements:

2 3 4 6

Enter the target value: 3 Output: 1

/\*33. program to split a string where an upper case letter occured in a string in java?

package assignmenttest;

import java.util.regex.\*;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string ");

String s1 = sc.nextLine();

// Define a regular expression pattern to match uppercase letters

Pattern p = Pattern.compile("(?=[A-Z])");

// Use the pattern to split the string

String[] parts = p.split(s1);

// Print the split parts

for (String part : parts) {

System.out.println(part);

}

}

}

\*/ o/p: enter the string I lOve My COuntry

I l Ove My C O untry\*/

/\*34. Given a string, s, and two indices, start and end, print a substring consisting of

all characters in the

inclusive range from start to end - 1.

Sample Input:Helloworld

3

7

Output:lowo

package assignmenttest;

import java.util.Scanner;

public class start\_end {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user for input System.out.print("Enter a string: "); String s = scanner.nextLine();

System.out.print("Enter the start index: ");

int start = scanner.nextInt(); System.out.print("Enter the end index: "); int end = scanner.nextInt();

// Check if the indices are valid

if (start >= 0 && end > start && end <= s.length()) {

// Get the substring from start to end - 1 String substring = s.substring(start, end);

bounds.");

// Print the substring System.out.println("Substring: " + substring);

} else {

System.out.println("Invalid input. Please make sure the indices are within

}

}

}

/\* o/P : Enter a string: helloworld Enter the start index: 3

Enter the end index: 7 Substring: lowo \*/

/\*35. Write a java program to take a string as user input and returns the count of

uppercase letters

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the sentence of your wish");

String s1 = sc.nextLine();

int Upper = 0;

int Lower = 0;

for (int i = 0; i < s1.length(); i++) {

char ch = s1.charAt(i);

if (ch>='A'&&ch<='Z') {

Upper++;

}

else if (ch>='a'&&ch<='z') {

Lower++;

}

}

System.out.println("The num of Upper case letters are: "+Upper);

System.out.println("The num of Lower case letters are: "+Lower);

}

}

/\*36. Write a java program to find out and print the longest word present in the

sentence.Write the code in the findLongestWord() method which accepts a string and

returns the longest word

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String args[])

{

Scanner scanner=new Scanner(System.in);

System.out.println("Enter the string");

String s1=scanner.nextLine();

String small="",large="",word="";

String words[]=new String[100];

int length=0;

s1=s1+"";

for(int i=0;i<s1.length();i++)

{

if(s1.charAt(i)==' ')

{

words[length]=word;

length++;

word="";

}

else

{

word=word+s1.charAt(i);

}

}

small=large=words[0];

for(int k=0;k<length;k++)

{

if(small.length()>words[k].length())

{

small=words[k];

}

if(large.length()<words[k].length())

{

large=words[k];

}

}

System.out.println("Smallest word "+small);

System.out.println("Largest word"+large);

}

}

/\*37. Write a program that takes your full name as input and displays the abbreviations

of the first and middle names except the last name which is displayed as it is. For

example, if your name is Robert Brett Roser, then the output should be R.B.Roser.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String args[])

{

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your full name: ");

String fullName = scanner.nextLine();

scanner.close();

String[] nameParts = fullName.split(" ");

int numNameParts = nameParts.length;

if (numNameParts >= 2) {

StringBuilder abbreviation = new StringBuilder();

// Abbreviate the first name

abbreviation.append(nameParts[0].charAt(0)).append(".");

// Abbreviate the middle names (if any)

for (int i = 1; i < numNameParts - 1; i++) {

abbreviation.append(nameParts[i].charAt(0)).append(".");

}

// Append the last name

abbreviation.append(nameParts[numNameParts - 1]);

System.out.println("Abbreviated name: " + abbreviation.toString());

} else {

System.out.println("Invalid input. Please enter your full name with at least two parts.");

}

}

}

\*/ o/p: Enter your full name: paul wesly john

Abbreviation: p.w.john

/\*38. Write a program to delete all consonents from the string "Hello, have a good day".

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

String s;

int j=0;

System.out.println("Enter a string");

Scanner so=new Scanner(System.in);

s= so.nextLine();

char ch[]=new char[20];

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

if(s.charAt(i)=='a'|| s.charAt(i)=='A'||s.charAt(i)=='e'|| s.charAt(i)=='E'||

s.charAt(i)=='i'|| s.charAt(i)=='I'||s.charAt(i)=='o'|| s.charAt(i)=='O'||

s.charAt(i)=='U'|| s.charAt(i)=='u')

{

continue;

}

else{

ch[j++]=s.charAt(i);

}

}

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

{

System.out.print(ch[i]);

}

System.out.println();

}

}

\*/ o/p :enter the string hii have a good day

Original string: hii have a good day

String with consonants removed: ii ae a oo a

/\*39. write a java program to capitalize each word in a string . input: "hello this is

java program" then the output should be : : "Hello This Is Java Program".

Package assignmenttest;

import java.util.Scanner;

public class capitalize {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

String input = scanner.nextLine();

String capitalizedString = capitalizeWords(input);

System.out.println("Capitalized string: " + capitalizedString);

}

public static String capitalizeWords(String input) { String[] words = input.split(" "); StringBuilder result = new StringBuilder();

for (String word : words) {

if (!word.isEmpty()) {

// Capitalize the first letter of each word

char firstChar = Character.toUpperCase(word.charAt(0)); String restOfWord = word.substring(1).toLowerCase(); result.append(firstChar).append(restOfWord).append(" ");

}}

// Remove the trailing space

if (result.length() > 0) { result.setLength(result.length() - 1);

}

return result.toString();

}}

/\*o/p: Enter a string: hello this is a java program

Capitalized string: Hello This Is A Java Program \*/

/\*40. WAP to reverse each word in a string. input : "i love my country" output: "i evol ym yrtnuoc".

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

String s1 =sc.nextLine();

String s2 = "";

int sp\_count=0;

//counting words

for (int i = 0; i < s1.length(); i++) {

if (s1.charAt(i)==' ') {

sp\_count++;

}

}

int word\_count=sp\_count+1;

//creating the array

String str[]=new String[word\_count];

int count=0;

//traversing string in reverse order

for (int i = s1.length()-1; i>=0; i--) {

if (s1.charAt(i)==' ') {

str[count]=s2;

s2="";

count++;

}

else {

s2=s2+s1.charAt(i);

}

}

str[count]=s2;

//printing the array

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

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

}

}

}

/\*41. Take 10 integer inputs from user and store them in an array. prompt user to give a

number.Check whether that number is present in array or not and if present print index

of the number.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int [] a = new int[10];

//storing all the 10 integer values

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

System.out.println("Enter the number"+(i+1));

a[i]=sc.nextInt();

}

//For searching the number in the array

System.out.println("Enter the number to search");

int search = sc.nextInt();

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

if (a[i]==search) {

System.out.println("The number present in the index of "+i);

}

}

}

}

\*/ Enter the number1 12

Enter the number2 16

Enter the number3

27

Enter the number4 19

Enter the number5 33

Enter the number to search 27

The number present in the index of 2

/\*42. Take 20 integer inputs from user and print the following:

number of positive numbers

number of negative numbers

number of odd numbers

number of even numbers

number of 0s.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int [] a = new int[20];

int positive = 0;

int negative = 0;

int odd = 0;

int even = 0;

int zero = 0;

//storing all the 10 integer values

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

System.out.println("Enter the number"+(i+1));

a[i]=sc.nextInt();

}

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

if (a[i]>=0) {

positive++;

}

if (a[i]<0) {

negative++;

}

if (a[i]%2==0&&a[i]%2>=0) {

even++;

}

if (a[i]%2!=0&&a[i]%2>=0) {

odd++;

}

if(a[i]==0) {

zero++;

}

}

System.out.println("The number of positive numbers are: "+(positive));

System.out.println("The number of negative numbers are: "+(negative));

System.out.println("The number of even numbers are: "+(even));

System.out.println("The number of odd numbers are: "+(odd));

System.out.println("The number of zeros are: "+(zero));

}

}

\*/ Enter 5 integers, one at a time:

Enter integer 1: 2

Enter integer 2: 0

Enter integer 3: 0

Enter integer 4: 4

Enter integer 5: 9

Number of positive numbers: 3 Number of negative numbers: 0 Number of odd numbers: 1 Number of even numbers: 4 Number of 0s: 2

/\*43. write a program to check whether the integer array contains only odd numbers?

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = sc.nextInt();

int [] a = new int[n];

//storing the values

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

System.out.println("Enter the num"+(i+1));

a[i]=sc.nextInt();

}

//to check weather the array contains only odd numbers or not

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

if (a[i]%2!=0) {

System.out.println("This array is containing odd numbers");

continue;

}

else if (a[i]%2==0) {

System.out.println("This array is also containing the even numbers");

break;

}

else {

System.out.println("This array is containing odd numbers");

continue;

}

}

}

}

\*//\* o/p : Enter the number of elements in the array: 5 Enter the array elements:

Element 1: 1

Element 2: 3

Element 3: 5

Element 4: 7

Element 5: 9

The array contains only odd numbers.

/\*44. Take 10 integer inputs from user and store them in an array. and copy all the

elements into an another array in reverse order.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter the size of the array");

int n = sc.nextInt();

int [] a = new int[n];

int [] b = new int[n];

//To store the elements

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

System.out.println("enter the element num"+(i+1));

a[i] = sc.nextInt();

}

//To reverse the array

int temp = 0;

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

b[temp]=a[i];

temp++;

}

//To print the reversed array

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

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

}

}

}

\*/ enter the size of the array 5

enter the element num1 12

enter the element num2 14

enter the element num3 16

enter the element num4 18

enter the element num5 20

20 18 16 14 12

/\*45. Find largest and smallest elements of an array.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = sc.nextInt();

int [] a = new int [n];

//storing the values

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

System.out.println("Enter the number"+(i+1));

a[i] = sc.nextInt();

}

Arrays.sort(a);

//soritng

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

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

}

System.out.println();

System.out.println("The smallest element is: "+a[0]);

System.out.println("The largest element is: "+a[n-1]);

}

}

/\*46. program to remove a particular element from an array.

package assignmenttest;

import java.lang.reflect.Array;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = sc.nextInt();

int [] a = new int[n];

int [] new\_arr = new int[n-1];

//for storing the elements

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

System.out.println("Enter the element num"+(i+1));

a[i]=sc.nextInt();

}

//to print the elements

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

System.out.print(a[i]);

}

System.out.println();

System.out.println("Enter the number you want to remove");

int remove =sc.nextInt();

for(int i=0, k=0;i<a.length;i++){

if(i!=remove){

new\_arr[k]=a[i];

k++;

}

}

System.out.println("Before deletion :" + Arrays.toString(a));

System.out.println("After deletion :" + Arrays.toString(new\_arr));

}

}

\*/ o/p: Enter the element to remove: 4 Updated Array:

1 2 3 5 6

/\*47. program to find the second largest integer in an array.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = sc.nextInt();

int [] a = new int [n];

//storing the values

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

System.out.println("Enter the number"+(i+1));

a[i] = sc.nextInt();

}

Arrays.sort(a);

//soritng

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

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

}

System.out.println();

System.out.println("The largest element is: "+a[n-2]);

}

}

\*/

/\*48. program sort an array of 0 and 1? input:[0,1,1,1,0,1,0] then the output should be

[0,0,0,1,1,1,1] without using

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int [] a = {0,1,1,1,0,1,0};

Arrays.sort(a);

System.out.println(Arrays.toString(a));

}

}

## o/p : [0, 0, 0, 1, 1, 1, 1]

/\*49. Consider an integer array, the number of elements in which is determined by the

user. The elements are also taken as input from the user. Write a program to find

those pair of elements that has the maximum and minimum difference among all element

pairs.

package assignmenttest;

import java.util.Scanner;

public class max\_min {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user for the number of elements in the array System.out.print("Enter the number of elements in the array: "); int n = scanner.nextInt();

// Initialize the array with user-defined size

int[] arr = new int[n];

// Prompt the user for the elements of the array System.out.println("Enter the elements of the array:"); for (int i = 0; i < n; i++) {

System.out.print("Element " + (i + 1) + ": "); arr[i] = scanner.nextInt();

}

// Initialize variables to keep track of max and min differences

int maxDifference = Integer.MIN\_VALUE;

int minDifference = Integer.MAX\_VALUE;

int maxDiffPairElement1 = 0, maxDiffPairElement2 = 0;

int minDiffPairElement1 = 0, minDiffPairElement2 = 0;

// Compare all pairs of elements to find max and min differences

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

for (int j = i + 1; j < n; j++) {

int difference = Math.abs(arr[i] - arr[j]);

// Update max difference

if (difference > maxDifference) { maxDifference = difference; maxDiffPairElement1 = arr[i]; maxDiffPairElement2 = arr[j];

}

// Update min difference

if (difference < minDifference) { minDifference = difference; minDiffPairElement1 = arr[i]; minDiffPairElement2 = arr[j];

}

}

}

// Print the elements with max and min differences

System.out.println("Pair with maximum difference: " + maxDiffPairElement1 + " and " + maxDiffPairElement2 + " with a difference of " + maxDifference);

System.out.println("Pair with minimum difference: " + minDiffPairElement1 + " and " + minDiffPairElement2 + " with a difference of " + minDifference);

}

}

/\* o/p : Enter the number of elements in the array: 4 Enter the elements of the array:

Element 1: 2

Element 2: 8

Element 3: 5

Element 4: 10

Pair with maximum difference: 2 and 10 with a difference of 8 Pair with minimum difference: 8 and 10 with a difference of 2 \*/

/\*50. program to find the missing integer in an array of range 1 to 10.

package assignmenttest;

import java.util.Arrays;

public class Launch {

public static void main(String[] args) {

int [] a = {1,2,3,5,6,7,8,9,10};

//to get the sum from 1 to 10

int sum =0;

for (int i = 1; i <= 10; i++) {

sum += i;

}

System.out.println("The sum of 1 to 10 range is: "+(sum));

//to get the actual sum of the defined array

int act\_sum = 0;

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

act\_sum += a[i];

}

System.out.println("The actual sum is: "+(act\_sum));

//To print the missing element in range of 1 to 10

if (a.length==10) {

System.out.println("There is no missing element in the range of 1 to 10");

} else {

System.out.println("The missing number in the 1 to 10 range is: "+(sum - act\_sum));

}

}

}

\*//\* o/p: The actual sum is: 47

The missing number in the 1 to 10 range is: 8

/\*51. How to Find Common Element Between Two Arrays In Java?

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the size of the array");

int n = scanner.nextInt();

int [] a = new int [n];

int [] b = new int [n];

int [] c = new int [n];

// to store the elements

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

System.out.println("Enter the element"+(i+1));

a[i]=scanner.nextInt();

}

// to store the elements

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

System.out.println("Enter the element"+(i+1));

b[i]=scanner.nextInt();

}

// to store the elements

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

if (a[i]==b[i]) {

c[i]=a[i];

}

}

//to print the same elements of two arrrays

System.out.println(Arrays.toString(c));

}

}

\*/ o/p: Enter the size of the first array: 3 Enter the elements of the first array:

2

4

6

Enter the size of the second array: 3 Enter the elements of the second array:

3

2

9

Common elements between the two arrays are: 2

/\*52. Find the highest palindrome number in a given array which contains both palindrome

and non-palindrome numbers?

package assignmenttest;

import java.util.Scanner;

public class palin\_nonpalin {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Prompt the user for the array size System.out.print("Enter the size of the array: "); int size = scanner.nextInt();

// Initialize the array

int[] arr = new int[size];

// Prompt the user to enter the elements of the array System.out.println("Enter the elements of the array:"); for (int i = 0; i < size; i++) {

arr[i] = scanner.nextInt();

}

int highestPalindrome = findHighestPalindrome(arr);

if (highestPalindrome != -1) {

System.out.println("The highest palindrome number in the array is: " + highestPalindrome);

} else {

System.out.println("No palindrome number found in the array.");

}}

// Function to check if a number is a palindrome

public static boolean isPalindrome(int num) {

int originalNum = num;

int reversedNum = 0;

while (num != 0) {

int digit = num % 10;

reversedNum = reversedNum \* 10 + digit; num /= 10;

}

return originalNum == reversedNum;

}

// Function to find the highest palindrome number in an array

public static int findHighestPalindrome(int[] arr) {

int highestPalindrome = -1;

for (int num : arr) {

if (isPalindrome(num) && num > highestPalindrome) { highestPalindrome = num;

}}

return highestPalindrome;

}}

/\*o/p: Enter the size of the array: 4 Enter the elements of the array:

1

2

2

1

The highest palindrome number in the array is: 2

/\*53. Move all negative numbers to beginning and positive to end.

package assignmenttest;

import java.util.Arrays;

import java.util.Iterator;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the size of an array");

int n = scanner.nextInt();

int [] a = new int[n];

//to store the elements

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

System.out.println("Enter the element"+(i+1));

a[i]=scanner.nextInt();

}

Arrays.sort(a);

System.out.println("The sorted array is "+Arrays.toString(a));

}

}

\*//\* Enter the size of an array 5

Enter the element1

-6

Enter the element2 4

Enter the element3 9

Enter the element4

-9

Enter the element5 6

The sorted array is [-9, -6,4, 6, 9]

/\*54. Given an integer array nums and an integer val, remove all occurrences of val in

nums in-place. The order of the elements may be changed. Then return the number of

elements in nums which are not equal to val.

package assignmenttest;

import java.util.Scanner;

public class int\_val {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the length of the integer array: ");

int n = scanner.nextInt();

int[] nums = new int[n];

System.out.println("Enter the elements of the integer array:");

for (int i = 0; i < n; i++) { nums[i] = scanner.nextInt();

}

System.out.print("Enter the value to remove: ");

int val = scanner.nextInt();

int newLength = removeElement(nums, val); System.out.println("Modified Array:");

for (int i = 0; i < newLength; i++) { System.out.print(nums[i] + " ");

}

System.out.println("\nNumber of elements not equal to " + val + ": " +

newLength);

scanner.close();

}

public static int removeElement(int[] nums, int val) {

int index = 0;

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

if (nums[i] != val) {

nums[index++] = nums[i];

}

}

return index;

}

}

/\* o/p:Enter the length of the integer array: 4 Enter the elements of the integer array:

2

4

6

8

Enter the value to remove: 8 Modified Array:

2 4 6

Number of elements not equal to 8: 3

/\*55. Consider the number of elements in nums which are not equal to val be k, to get

accepted, you need to do the following things:

Change the array nums such that the first k elements of nums contain the elements

which are not equal to val. The remaining elements of nums are not important as well

as the size of nums.

Return k.

Example

Input: nums = [3,2,2,3], val = 3

Output: 2, nums = [2,2,,]

Explanation: Your function should return k = 2, with the first two elements of nums

being 2

package assignmenttest;

import java.util.Scanner;

public class num {

public static int removeElement(int[] nums, int val) {

int k = 0; // Initialize k to 0 the array

// Iterate through the array

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

if (nums[i] != val) {

nums[k] = nums[i]; // Move non-val elements to the front ofk++; // Increment k

}

}

return k;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int n = scanner.nextInt();

int[] nums = new int[n];

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

for (int i = 0; i < n; i++) { nums[i] = scanner.nextInt();

}

System.out.print("Enter the value to remove: ");

int val = scanner.nextInt();

int k = removeElement(nums, val);

System.out.println("Output: " + k); System.out.print("Updated array: ["); for (int i = 0; i < k; i++) {

System.out.print(nums[i]);

if (i < k - 1) {

System.out.print(", ");

}

}

System.out.println("]");

}

}

/\* o/p : Enter the number of elements in the array: 4 Enter the elements of the array:

3 2 2 3

Enter the value to remove: 3 Output: 2

Updated array: [2, 2]

/\*56. find the sum of the digits of a number \*/

package assignmenttest;

import java.util.Scanner;

public class Ques\_56\_digit {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number: ");

int number = scanner.nextInt();

int sum = 0;

int originalNumber = number;

// Calculate the sum of digits

while (number != 0) {

int digit = number % 10; sum += digit;

number /= 10;

}

System.out.println("Sum of digits in " + originalNumber + " is: " + sum);

}

}

/\* o/p : Enter a number: 5000

Sum of digits in 5000 is: 5

/\*57. print below pattern

1

0 1

1 0 1

0 1 0 1

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter the size");

int n = sc.nextInt();

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

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

if ((i+j)%2==0) {

System.out.print(1+" ");

}

else {

System.out.print(0+" ");

}

}System.out.println();

}

}

}

Output

1

0 1

1 0 1

0 1 0 1

/\*58. Write a program to remove characters from the first string which are present in

second string

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the string");

String s1 = scanner.nextLine();

System.out.println("Enter the character you want to replace");

String s2 = scanner.next();

s1=s1.replace(s2, "");

System.out.println(s1);

}

}

\*/ o/p :Enter the first string: king

Enter the second string: queen

Result after removing common characters: kig

/\*59. write a program to find the given string is sorted with a specified string or

not .Example(Launch="JavaScript" ,specified string is"java" the output is "true")

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the string");

String s1 = scanner.nextLine();

System.out.println("Enter the string you want to search");

String s2 = scanner.next();

if (s1.contains(s2)) {

System.out.println("The string contains the word you entered");

} else {

System.out.println("The string doesn't contains the word u entered");

}

}

}

\*/ Enter the string corejava

Enter the string you want to search java

The string contains the word you entered

/\*60. print below pattern

A B C

A B

A

A

A B

A B C

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size");

int n = sc.nextInt();

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

for (int j = 0; j < n; j++) {

if (j==0) {

System.out.print("A ");

}

else if (j==1&&i<=1 || j==1&&i>=4) {

System.out.print("B ");

}

else if (j==2&&i==0 || j==2&&i==n-1) {

System.out.print("C ");

}

}

System.out.println();

}

}

}

\*/ A B C

A B

A

A

A B

A B C

BASED ON OOPS

61. Write a java code to solve the Dining Philosophers problem?

package oops\_questions;

import java.util.concurrent.Semaphore;

class Philosopher extends Thread {

private Semaphore leftFork;

private Semaphore rightFork;

private int id;

public Philosopher(int id, Semaphore leftFork, Semaphore rightFork) {

this.id = id;

this.leftFork = leftFork;

this.rightFork = rightFork;

}

private void think() {

System.out.println("Philosopher " + id + " is thinking.");

try {

Thread.sleep((long) (Math.random() \* 10));

} catch (InterruptedException e) {

e.printStackTrace();

}

}

private void eat() {

System.out.println("Philosopher " + id + " is eating.");

try {

Thread.sleep((long) (Math.random() \* 10));

}

catch (InterruptedException e) {

e.printStackTrace();

}

}

public void run() {

while (true) {

think();

try {

leftFork.acquire();

System.out.println("Philosopher " + id + " picked up the left fork.");

rightFork.acquire();

System.out.println("Philosopher " + id + " picked up the right fork and is now eating.");

eat();

leftFork.release();

System.out.println("Philosopher " + id + " released the left fork.");

rightFork.release();

System.out.println("Philosopher " + id + " released the right fork and is now thinking.");

} catch (InterruptedException e) {

e.printStackTrace();

} } }}

public class program61 {

public static void main(String[] args) {

int numPhilosophers = 5;

Semaphore[] forks = new Semaphore[numPhilosophers];

for (int i = 0; i < numPhilosophers; i++) {

forks[i] = new Semaphore(1);

}

Philosopher[] philosophers = new Philosopher[numPhilosophers];

for (int i = 0; i < numPhilosophers; i++) {

philosophers[i] = new Philosopher(i, forks[i], forks[(i + 1) % numPhilosophers]);

philosophers[i].start()}

}

}

Output

Philosopher 1 is thinking.

Philosopher 0 is thinking.

Philosopher 2 is thinking.

Philosopher 3 is thinking.

Philosopher 1 picked up the left fork.

Philosopher 1 picked up the right fork and is now eating.

Philosopher 1 is eating.

Philosopher 4 is thinking.

Philosopher 3 picked up the left fork.

Philosopher 3 picked up the right fork and is now eating.

Philosopher 3 is eating.

62. Solve the 0/1 Knapsack Problem using dynamic programming in Java.

package assignmenttest;

public class Launch {

public static int knapsack(int[] weights, int[] values, int capacity) {

int n = weights.length;

int[][] dp = new int[n + 1][capacity + 1];

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

for (int w = 0; w <= capacity; w++) {

if (i == 0 || w == 0) {

dp[i][w] = 0;

} else if (weights[i - 1] <= w) {

dp[i][w] = Math.max(values[i - 1] + dp[i - 1][w - weights[i - 1]], dp[i -

1][w]);

} else {

dp[i][w] = dp[i - 1][w];

}

}

}

return dp[n][capacity];

}

public static void main(String[] args) {

int[] weights = {2, 3, 4, 5};

int[] values = {3, 4, 5, 6};

int capacity = 5;

int maxValue = knapsack(weights, values, capacity);

System.out.println("Maximum Value: " + maxValue);

}

}

63.Write a program to solve the classic "N-Queens" problem, where you need to place N queens on an N×N chessboard such that no two queens threaten each other?(N-Queens Problem)

package oops\_questions;

import java.util.\*;

public class program43 {

// Store all the possible answers

static List<List<String> > answer = new ArrayList<>();

// Print the board

static void print\_board()

{

for (String str : answer.get(1)) {

for (Character letter : str.toCharArray())

System.out.print(letter + " ");

System.out.println();

}

return;

}

static boolean safe(int row, int col,

List<String> board)

{

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

if (board.get(i).charAt(col) == 'Q')

return false;

}

int i = row, j = col;

while (i >= 0 && j >= 0)

if (board.get(i--).charAt(j--) == 'Q')

return false;

i = row;

j = col;

while (i >= 0 && j < board.size())

if (board.get(i--).charAt(j++) == 'Q')

return false;

return true;

}

static void rec(List<String> board, int row)

{

if (row == board.size()) {

answer.add(board);

return;

}

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

if (safe(row, i, board)) {

List<String> temp = new ArrayList<>(board);

temp.set(

row,

temp.get(row).substring(0, i) + "Q"

+ temp.get(row).substring(i + 1));

rec(temp, row + 1);

}

}

return;

}

// Function to solve n queens

static List<List<String> > solveNQueens(int n)

{

String s

= new String(new char[n]).replace("\0", ".");

List<String> board = new ArrayList<>();

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

board.add(s);

rec(board, 0);

return answer;

}

public static void main(String[] args)

{

// Size 4x4 is taken and we can pass some other

// dimension for chess board as well

System.out.println(solveNQueens(4).size());

System.out.println("Out of " + answer.size()

+ " solutions one is following");

print\_board();

}

}

Output

2

Out of 2 solutions one is following

. . Q .

Q . . .

. . . Q

. Q . .

64.Define a base class "Shape" with a method "calculateArea." Create two subclasses, "Circle" and "Triangle," that inherit from "Shape" and provide their own implementations of "calculateArea." Demonstrate polymorphism by calculating the area of various shapes using an array of "Shape" objects.

import java.util.Scanner;

// Base class

class Shape {

public double calculateArea() {

return 0.0; // Default implementation

}

}

// Subclass: Circle

class Circle extends Shape {

private double radius;

public Circle(double radius) {

this.radius = radius;

}

@Override

public double calculateArea() {

return Math.PI \* radius \* radius;

}

}

// Subclass: Triangle

class Triangle extends Shape {

private double base;

private double height;

public Triangle(double base, double height) {

this.base = base;

this.height = height;

}

@Override

public double calculateArea() {

return 0.5 \* base \* height;

}

}

public class Ques\_shapes {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter the number of shapes:");

int numShapes = scanner.nextInt();

Shape[] shapes = new Shape[numShapes];

for (int i = 0; i < numShapes; i++) {

System.out.println("Enter shape type (1 for Circle, 2 for Triangle):");

int shapeType = scanner.nextInt();

if (shapeType == 1) {

System.out.println("Enter circle radius:");

double radius = scanner.nextDouble();

shapes[i] = new Circle(radius);

} else if (shapeType == 2) {

System.out.println("Enter triangle base:");

double base = scanner.nextDouble();

System.out.println("Enter triangle height:");

double height = scanner.nextDouble();

shapes[i] = new Triangle(base, height);

}

}

System.out.println("Calculating areas:");

for (Shape shape : shapes) {

double area = shape.calculateArea();

System.out.println("Area: " + area);

}

}

}

Output

Enter the number of shapes: 1

Enter shape type (1 for Circle, 2 for Triangle): 1

Enter circle radius: 2.6

Calculating areas: Area: 21.237166338267002

65.Implement a class hierarchy for a zoo simulation. Create a base class "Animal" with a method "makeSound." Then, create subclasses for different animals like "Lion," "Elephant," and "Monkey" that override the "makeSound" method to produce their specific sounds. Use polymorphism to make animals in the zoo make their sounds.

package oops\_questions;

import java.util.Scanner;

class Animal {

public void makeSound() {

System.out.println("The animal makes a generic sound.");

}

}

// Subclass Lion

class Lion extends Animal {

@Override

public void makeSound() {

System.out.println("The lion roars.");

}

}

// Subclass Elephant

class Elephant extends Animal {

@Override

public void makeSound() {

System.out.println("The elephant trumpets.");

}

}

// Subclass Monkey

class Monkey extends Animal {

@Override

public void makeSound() {

System.out.println("The monkey chatters.");

}

}

class zoo{

void permit(Animal ref) {

ref.makeSound();

}

}

public class program65 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("\*\*\*\*\*\*Welcome to the Zoo Simulation!\*\*\*\*\*\*");

Lion l = new Lion();

Elephant e = new Elephant();

Monkey m = new Monkey();

zoo z= new zoo();

z.permit(l);

z.permit(e);

z.permit(m);

}

}

Output

\*\*\*\*\*\*Welcome to the Zoo Simulation\*\*\*\*\*

The lion roars.

The elephant trumpets.

The monkey chatters.

66.Develop a banking system with a base class "Account" and subclasses "SavingsAccount" and "CheckingAccount." Each account type should have a method "calculateInterest" that calculates interest differently.

Demonstrate polymorphism by calling the "calculateInterest" method on both account types.

package oops\_questions;

import java.util.Scanner;

//Base class

class Account {

protected double balance;

public Account(double balance) {

this.balance = balance;

}

public void deposit(double amount) {

balance += amount;

}

public void withdraw(double amount) {

if (amount <= balance) {

balance -= amount;

} else {

System.out.println("Insufficient funds");

}

}

public void displayBalance() {

System.out.println("Current balance: $" + balance);

}

public void calculateInterest() {

System.out.println("Base Account - Interest calculation method");

}

}

//Subclass for Savings Account

class SavingsAccount extends Account {

public SavingsAccount(double balance) {

super(balance);

}

@Override

public void calculateInterest() {

double interest = balance \* 0.03; // Example: 3% annual interest

balance += interest;

System.out.println("Savings Account - Interest calculated: $" + interest);

}

}

//Subclass for Checking Account

class CheckingAccount extends Account {

public CheckingAccount(double balance) {

super(balance);

}

@Override

public void calculateInterest() {

System.out.println("Checking Account - No interest is calculated");

}

}

public class program66 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Welcome to the Simple Banking System!");

System.out.print("Enter initial balance for Savings Account: ");

double savingsBalance = scanner.nextDouble();

System.out.print("Enter initial balance for Checking Account: ");

double checkingBalance = scanner.nextDouble();

SavingsAccount savingsAccount = new SavingsAccount(savingsBalance);

CheckingAccount checkingAccount = new CheckingAccount(checkingBalance);

System.out.print("Enter the amount to deposit in Savings Account: ");

double depositAmount = scanner.nextDouble();

savingsAccount.deposit(depositAmount);

System.out.print("Enter the amount to withdraw from Checking Account: ");

double withdrawAmount = scanner.nextDouble();

checkingAccount.withdraw(withdrawAmount);

// Demonstrate polymorphism

Account[] accounts = {savingsAccount, checkingAccount};

for (Account account : accounts) {

account.displayBalance();

account.calculateInterest();

}

}

}

Output

Welcome to the Simple Banking System!

Enter initial balance for Savings Account:

2000

Enter initial balance for Checking Account: 500

Enter the amount to deposit in Savings Account: 500

Enter the amount to withdraw from Checking Account: 500

Current balance: $2500.0

Savings Account - Interest calculated: $75.0

Current balance: $0.0

Checking Account - No interest is calculated

67. Write a java code to verify performance of StringBuffer and StringBuilder classes.

public class StringBufferVsStringBuilderPerformance {

public static void main(String[] args) {

int iterations = 100000;

String textToAppend = "Hello, World!";

long startTime, endTime;

// Using StringBuffer

StringBuffer stringBuffer = new StringBuffer();

startTime = System.nanoTime();

for (int i = 0; i < iterations; i++) {

stringBuffer.append(textToAppend);

}

endTime = System.nanoTime();

long stringBufferTime = endTime - startTime;

// Using StringBuilder

StringBuilder stringBuilder = new StringBuilder();

startTime = System.nanoTime();

for (int i = 0; i < iterations; i++) {

stringBuilder.append(textToAppend);

}

endTime = System.nanoTime();

long stringBuilderTime = endTime - startTime;

System.out.println("StringBuffer Time: " + stringBufferTime + " nanoseconds");

System.out.println("StringBuilder Time: " + stringBuilderTime + " nanoseconds");

}

}

Output

StringBuffer Time: 15076400 nanoseconds

StringBuilder Time: 6554100 nanoseconds

68.Build a simple media player application with a base class "MediaPlayer" and subclasses "AudioPlayer" and "VideoPlayer." Each subclass should have a method "play" to display messages like "Playing audio" or "Playing video." Utilize polymorphism to play different media types.

package oops\_questions;

import java.util.Scanner;

//Base class MediaPlayer

class MediaPlayer {

public void play() {

// Default play method for all media types

System.out.println("Playing media...");

}

}

//Subclass AudioPlayer

class AudioPlayer extends MediaPlayer {

@Override

public void play() {

System.out.println("Playing audio...");

}

}

//Subclass VideoPlayer

class VideoPlayer extends MediaPlayer {

@Override

public void play() {

System.out.println("Playing video...");

}

}

class player{

void permit(MediaPlayer ref) {

ref.play();

}

}

public class program68 {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println(" media type");

AudioPlayer ap = new AudioPlayer();

VideoPlayer vp = new VideoPlayer();

player p = new player();

p.permit(ap);

p.permit(vp);

}

}

Output

media type

Playing audio...

Playing video...

69.Create a Java class hierarchy for geometric shapes, including a base class "Shape" with methods for calculating area and perimeter. Implement two subclasses, "Circle" and "Rectangle," that inherit from the "Shape" class and provide their own implementations of area and perimeter calculations.(Inheritance)

package oops\_questions;

//Base class Shape5

abstract class Shape5 {

abstract double calculateArea();

abstract double calculatePerimeter();

}

//Subclass Circle5

class Circle5 extends Shape5 {

private double radius;

public Circle5(double radius) {

this.radius = radius;

}

@Override

double calculateArea() {

return Math.PI \* radius \* radius;

}

@Override

double calculatePerimeter() {

return 2 \* Math.PI \* radius;

}

}

//Subclass Rectangle

class Rectangle extends Shape5 {

private double width;

private double height;

public Rectangle(double width, double height) {

this.width = width;

this.height = height;

}

@Override

double calculateArea() {

return width \* height;

}

@Override

double calculatePerimeter() {

return 2 \* (width + height);

}

}

public class program69 {

public static void main(String[] args) {

Circle5 Circle5 = new Circle5(5.0);

Rectangle rectangle = new Rectangle(4.0, 6.0);

System.out.println("Circle5 Area: " + Circle5.calculateArea());

System.out.println("Circle5 Perimeter: " + Circle5.calculatePerimeter());

System.out.println("Rectangle Area: " + rectangle.calculateArea());

System.out.println("Rectangle Perimeter: " + rectangle.calculatePerimeter());

}

}

Output

Circle5 Area: 78.53981633974483

Circle5 Perimeter: 31.41592653589793

Rectangle Area: 24.0

Rectangle Perimeter: 20.0

70.Define a class "Animal" with properties like name, age, and sound. Create two subclasses, "Dog" and "Cat," that inherit from the "Animal" class. Add methods to both subclasses to make them produce their respective sounds. Demonstrate polymorphism by creating instances of each subclass and calling the sound methods. (Inheritance)

package oops\_questions;

class Animall {

private String name;

private int age;

private String sound;

public Animall(String name, int age, String sound) {

this.name = name;

this.age = age;

this.sound = sound;

}

public String getName() {

return name;

}

public int getAge() {

return age;

}

public void makeSound() {

System.out.println(name + " makes a sound: " + sound);

}

}

// The Dog subclass that inherits from Animall

class Dog extends Animall {

public Dog(String name, int age) {

super(name, age, "Woof!");

}

}

// The Cat subclass that inherits from Animall

class Cat extends Animall {

public Cat(String name, int age) {

super(name, age, "Meow!");

}

}

public class program70 {

public static void main(String[] args) {

// Create instances of Dog and Cat

Dog myDog = new Dog("shadow", 3);

Cat myCat = new Cat("monty", 5);

// Demonstrate polymorphism by calling the makeSound method

Animall[] Animalls = {myDog, myCat};

for (Animall Animall : Animalls) {

System.out.println(Animall.getName() + " is " + Animall.getAge() + " years old.");

Animall.makeSound();

System.out.println();

}

}

}

Output

shadowis 3 years old.

shadowmakes a sound: Woof!

Monty is 5 years old.

monty makes a sound: Meow!

71.Design a class hierarchy for a banking system. Create a base class

"Account" with fields for account number, account holder name, and

balance. Implement two subclasses, "SavingsAccount" and

"CheckingAccount," that inherit from the "Account" class. Add methods for deposit, withdrawal, and account-specific behaviors. (Inheritance)

package assignmenttest;

//Base class Account

class Account {

private int accountNumber;

private String accountHolderName;

private double balance;

public Account(int accountNumber, String accountHolderName, double balance) {

this.accountNumber = accountNumber;

this.accountHolderName = accountHolderName;

this.setBalance(balance);

}

public int getAccountNumber() {

return accountNumber;

}

public String getAccountHolderName() {

return accountHolderName;

}

public double getBalance() {

return balance;

}

public void deposit(double amount) {

if (amount > 0) {

setBalance(getBalance() + amount);

System.out.println("Deposited $" + amount);

} else {

System.out.println("Invalid deposit amount.");

}

}

public void withdraw(double amount) {

if (amount > 0 && amount <= getBalance()) {

setBalance(getBalance() - amount);

System.out.println("Withdrawn $" + amount);

} else {

System.out.println("Invalid withdrawal amount or insufficient balance.");

}

}

public void displayAccountInfo() {

System.out.println("Account Number: " + accountNumber);

System.out.println("Account Holder: " + accountHolderName);

System.out.println("Balance: $" + getBalance());

}

public void setBalance(double balance) {

this.balance = balance;

}

}

//Subclass SavingsAccount

class SavingsAccount extends Account {

private double interestRate;

public SavingsAccount(int accountNumber, String accountHolderName, double balance, double

interestRate) {

super(accountNumber, accountHolderName, balance);

this.interestRate = interestRate;

}

public void applyInterest() {

double interest = getBalance() \* interestRate;

deposit(interest);

System.out.println("Interest applied: $" + interest);

}

}

//Subclass CheckingAccount

class CheckingAccount extends Account {

private double overdraftLimit;

public CheckingAccount(int accountNumber, String accountHolderName, double balance, double

overdraftLimit) {

super(accountNumber, accountHolderName, balance);

this.overdraftLimit = overdraftLimit;

}

@Override

public void withdraw(double amount) {

if (amount > 0 && amount <= (getBalance() + overdraftLimit)) {

setBalance(getBalance() - amount);

System.out.println("Withdrawn $" + amount);

} else {

System.out.println("Invalid withdrawal amount or exceeded overdraft limit.");

}

}

}

public class Launch {

public static void main(String[] args) {

// Create a savings account

SavingsAccount savingsAccount = new SavingsAccount(12345, "John Doe", 1000.0, 0.05);

savingsAccount.displayAccountInfo();

savingsAccount.deposit(500.0);

savingsAccount.applyInterest();

savingsAccount.withdraw(200.0);

savingsAccount.displayAccountInfo();

// Create a checking account

CheckingAccount checkingAccount = new CheckingAccount(67890, "Jane Smith", 500.0,

100.0);

checkingAccount.displayAccountInfo();

checkingAccount.deposit(200.0);

checkingAccount.withdraw(800.0); // Attempt to exceed overdraft limit

checkingAccount.displayAccountInfo();

}

}

72. Create a "Person" class with properties like name and address. Implement a subclass "Student" that inherits from "Person" and includes additional properties like student ID and GPA. Then, create another subclass "Teacher" that inherits from "Person" and has properties like employee ID and subject taught.

package assignmenttest;

//Person class

class Person {

private String name;

private String address;

public Person(String name, String address) {

this.name = name;

this.address = address;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getAddress() {

return address;

}

public void setAddress(String address) {

this.address = address;

}

}

//Student class (subclass of Person)

class Student extends Person {

private int studentId;

private double gpa;

public Student(String name, String address, int studentId, double gpa) {

super(name, address);

this.studentId = studentId;

this.gpa = gpa;

}

public int getStudentId() {

return studentId;

}

public void setStudentId(int studentId) {

this.studentId = studentId;

}

public double getGpa() {

return gpa;

}

public void setGpa(double gpa) {

this.gpa = gpa;

}

}

//Teacher class (subclass of Person)

class Teacher extends Person {

private int employeeId;

private String subjectTaught;

public Teacher(String name, String address, int employeeId, String subjectTaught) {

super(name, address);

this.employeeId = employeeId;

this.subjectTaught = subjectTaught;

}

public int getEmployeeId() {

return employeeId;

}

public void setEmployeeId(int employeeId) {

this.employeeId = employeeId;

}

public String getSubjectTaught() {

return subjectTaught;

}

public void setSubjectTaught(String subjectTaught) {

this.subjectTaught = subjectTaught;

}

}

//Sample program to demonstrate usage

public class Launch {

public static void main(String[] args) {

Student student = new Student("John Doe", "123 Main St", 101, 3.8);

Teacher teacher = new Teacher("Jane Smith", "456 Elm St", 201, "Mathematics");

// Display student information

System.out.println("Student Information:");

System.out.println("Name: " + student.getName());

System.out.println("Address: " + student.getAddress());

System.out.println("Student ID: " + student.getStudentId());

System.out.println("GPA: " + student.getGpa());

// Display teacher information

System.out.println("\nTeacher Information:");

System.out.println("Name: " + teacher.getName());

System.out.println("Address: " + teacher.getAddress());

System.out.println("Employee ID: " + teacher.getEmployeeId());

System.out.println("Subject Taught: " + teacher.getSubjectTaught());

}

}

73. Create a Java class to represent a "Book" with private fields for title,

author, and price. Provide encapsulated methods to get and set these

fields. Write a sample program to demonstrate its usage.

package assignmenttest;

public class Launch {

private String title;

private String author;

private double price;

// Constructor

public Launch(String title, String author, double price) {

this.title = title;

this.author = author;

this.price = price;

}

// Getter for title

public String getTitle() {

return title;

}

// Setter for title

public void setTitle(String title) {

this.title = title;

}

// Getter for author

public String getAuthor() {

return author;

}

// Setter for author

public void setAuthor(String author) {

this.author = author;

}

// Getter for price

public double getPrice() {

return price;

}

// Setter for price

public void setPrice(double price) {

this.price = price;

}

// Sample program to demonstrate usage

public static void main(String[] args) {

// Create a Book object

Launch myBook = new Launch("The Great Gatsby", "F. Scott Fitzgerald", 12.99);

// Access and print the book's information

System.out.println("Title: " + myBook.getTitle());

System.out.println("Author: " + myBook.getAuthor());

System.out.println("Price: $" + myBook.getPrice());

// Update the book's price

myBook.setPrice(14.99);

// Display the updated price

System.out.println("Updated Price: $" + myBook.getPrice());

}

}

74. Implement a class called "Employee" with private fields for name, salary, and employee ID. Ensure encapsulation and provide a method to give a yearly bonus to the employee. Write a program to test this class.

package assignmenttest;

public class Launch {

private String name;

private double salary;

private int employeeID;

// Constructor to initialize name, salary, and employeeID

public Launch(String name, double salary, int employeeID) {

this.name = name;

this.salary = salary;

this.employeeID = employeeID;

}

// Getter method to get the name

public String getName() {

return name;

}

// Setter method to set the name

public void setName(String name) {

this.name = name;

}

// Getter method to get the salary

public double getSalary() {

return salary;

}

// Setter method to set the salary

public void setSalary(double salary) {

this.salary = salary;

}

// Getter method to get the employee ID

public int getEmployeeID() {

return employeeID;

}

// Method to give a yearly bonus to the employee

public void giveYearlyBonus(double bonus) {

salary += bonus;

}

public static void main(String[] args) {

// Create an Employee object

Launch employee = new Launch("John Doe", 50000.0, 12345);

// Display employee information

System.out.println("Employee Name: " + employee.getName());

System.out.println("Employee Salary: " + employee.getSalary());

System.out.println("Employee ID: " + employee.getEmployeeID());

// Give a yearly bonus of $2000

employee.giveYearlyBonus(2000.0);

// Display updated salary after the bonus

System.out.println("Updated Salary after Bonus: " + employee.getSalary());

}

}

75.Create a "Circle" class with a private field for radius. Implement

encapsulated methods to set and get the radius and calculate the area of

the circle. Write a program to calculate the area of a circle using this class.

package assignmenttest;

public class Launch {

private double radius;

// Constructor to initialize the radius

public Launch(double radius) {

this.radius = radius;

}

// Getter method to get the radius

public double getRadius() {

return radius;

}

// Setter method to set the radius

public void setRadius(double radius) {

this.radius = radius;

}

// Method to calculate the area of the circle

public double calculateArea() {

return Math.PI \* radius \* radius;

}

public static void main(String[] args) {

// Create a Circle object with a radius of 5.0

Launch circle = new Launch(5.0);

// Get the radius

double radius = circle.getRadius();

// Calculate and display the area

double area = circle.calculateArea();

System.out.println("Radius: " + radius);

System.out.println("Area: " + area);

}

}

76.Design a class called "Person" with private fields for name, age, and

gender. Ensure encapsulation and provide a method to check if a person is

eligible to vote (age >= 18). Write a program to test this class.

package assignmenttest;

public class Launch {

private String name;

private int age;

private String gender;

// Constructor

public Launch(String name, int age, String gender) {

this.name = name;

this.age = age;

this.gender = gender;

}

// Getter methods

public String getName() {

return name;

}

public int getAge() {

return age;

}

public String getGender() {

return gender;

}

// Method to check if the person is eligible to vote

public boolean isEligibleToVote() {

return age >= 18;

}

public static void main(String[] args) {

// Creating a Person object

Launch person1 = new Launch("John", 25, "Male");

// Testing eligibility to vote

if (person1.isEligibleToVote()) {

System.out.println(person1.getName() + " is eligible to vote.");

} else {

System.out.println(person1.getName() + " is not eligible to vote.");

}

}

}

77.Implement a "Bank" class that manages customer accounts. Each account should have a private balance field. Provide methods for deposit,

withdrawal, and checking the balance, ensuring that the balance cannot go

below zero. Write a program to simulate banking operations.

package assignmenttest;

import java.util.Scanner;

class BankDetails {

private String accno;

private String name;

private String acc\_type;

private long balance;

Scanner sc = new Scanner(System.in);

//method to open new account

public void openAccount() {

System.out.print("Enter Account No: ");

accno = sc.next();

System.out.print("Enter Account type: ");

acc\_type = sc.next();

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

name = sc.next();

System.out.print("Enter Balance: ");

balance = sc.nextLong();

}

//method to display account details

public void showAccount() {

System.out.println("Name of account holder: " + name);

System.out.println("Account no.: " + accno);

System.out.println("Account type: " + acc\_type);

System.out.println("Balance: " + balance);

}

//method to deposit money

public void deposit() {

long amt;

System.out.println("Enter the amount you want to deposit: ");

amt = sc.nextLong();

balance = balance + amt;

}

//method to withdraw money

public void withdrawal() {

long amt;

System.out.println("Enter the amount you want to withdraw: ");

amt = sc.nextLong();

if (balance >= amt) {

balance = balance - amt;

System.out.println("Balance after withdrawal: " + balance);

} else {

System.out.println("Your balance is less than " + amt + "\tTransaction failed...!!" );

}

}

//method to search an account number

public boolean search(String ac\_no) {

if (accno.equals(ac\_no)) {

showAccount();

return (true);

}

return (false);

}

}

public class Launch {

public static void main(String arg[]) {

Scanner sc = new Scanner(System.in);

//create initial accounts

System.out.print("How many number of customers do you want to input? ");

int n = sc.nextInt();

BankDetails C[] = new BankDetails[n];

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

C[i] = new BankDetails();

C[i].openAccount();

}

// loop runs until number 5 is not pressed to exit

int ch;

do {

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

System.out.println("1. Display all account details \n 2. Search by Account number\n

3. Deposit the amount \n 4. Withdraw the amount \n 5.Exit ");

System.out.println("Enter your choice: ");

ch = sc.nextInt();

switch (ch) {

case 1:

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

C[i].showAccount();

}

break;

case 2:

System.out.print("Enter account no. you want to search: ");

String ac\_no = sc.next();

boolean found = false;

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

found = C[i].search(ac\_no);

if (found) {

break;

}

}

if (!found) {

System.out.println("Search failed! Account doesn't exist..!!");

}

break;

case 3:

System.out.print("Enter Account no. : ");

ac\_no = sc.next();

found = false;

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

found = C[i].search(ac\_no);

if (found) {

C[i].deposit();

break;

}

}

if (!found) {

System.out.println("Search failed! Account doesn't exist..!!");

}

break;

case 4:

System.out.print("Enter Account No : ");

ac\_no = sc.next();

found = false;

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

found = C[i].search(ac\_no);

if (found) {

C[i].withdrawal();

break;

}

}

if (!found) {

System.out.println("Search failed! Account doesn't exist..!!");

}

break;

case 5:

System.out.println("See you soon...");

break;

}

}

while (ch != 5);

}

}

78. A chocolate factory is packing chocolates into the packets. The

chocolate packets here represent an array of N number of integer

values. The task is to find the empty packets(0) of chocolate and push

it to the end of the conveyor belt(array

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("enter the size");

// Read the value of N

int N = sc.nextInt();

sc.nextLine(); // Consume the newline

// Read the array elements

int[] arr = new int[N];

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

arr[i] = sc.nextInt();

}

// Process the array to push empty packets to the end

int emptyPacketCount = 0;

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

if (arr[i] != 0) {

arr[emptyPacketCount++] = arr[i];

}

}

// Fill the remaining positions with 0s

while (emptyPacketCount < N) {

arr[emptyPacketCount++] = 0;

}

// Print the modified array

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

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

}

sc.close();

}

}

79. Selection of MPCS exams include a fitness test which is conducted on

ground. There will be a batch of 3 trainees, appearing for running test

in track for 3 rounds. You need to record their oxygen level after every

round. After trainee are finished with all rounds, calculate for each

trainee his average oxygen level over the 3 rounds and select one with

highest oxygen level as the most fit trainee. If more than one trainee

attains the same highest average level, they all need to be selected.

Display the most fit trainee (or trainees) and the highest average

oxygen level

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int[][] trainee = new int[3][3];

int[] average = new int[3];

int max = 0;

Scanner sc = new Scanner(System.in);

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

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

if(trainee[i][j] < 1 || trainee[i][j] > 100) {

trainee[i][j] = 0;

}

}

}

for(int i = 0; i < 3; i++) {

for(int j = 0; j < 3; j++) {

average[i] = average[i] + trainee[j][i];

}

average[i] = average[i] / 3;

}

for(int i = 0; i < 3; i++) {

if(average[i] > max) {

max = average[i];

}

}

for(int i = 0; i < 3; i++) {

if(average[i] == max) {

System.out.println("Trainee Number : " + (i + 1));

}

if(average[i] <70) {

System.out.print("Trainee is Unfit");

}

}

}

}

80. The Caesar cipher is a type of substitution cipher in which each alphabet

in the plaintext or messages is shifted by a number of places down the

alphabet.

import java.util.Scanner;

public class Launch {

public static String CustomCaesarCipher(int key, String message) {

if (key < 0) {

return "INVALID INPUT";

}

StringBuilder cipherText = new StringBuilder();

for (char character : message.toCharArray()) {

if (Character.isLetter(character)) {

char base = Character.isLowerCase(character) ? 'a' : 'A';

cipherText.append((char) ((character - base + key) % 26 + base));

} else if (Character.isDigit(character)) {

int digit = Character.getNumericValue(character);

int shiftedDigit = (digit + key) % 10;

cipherText.append(shiftedDigit);

} else {

cipherText.append(character); // Keep non-alphanumeric characters unchanged

}

}

return cipherText.toString();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter your PlainText: ");

String plainText = scanner.nextLine();

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

int key = scanner.nextInt();

scanner.close();

String cipherText = CustomCaesarCipher(key, plainText);

System.out.println("The encrypted Text is: " + cipherText);

}

81. Write a program in Java to calculate the number of times a digit ‘D’ appears in a number N. You have to take N and D as inputs from the user.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number");

long n = sc.nextLong();

System.out.println("Enter the digit to find");

int d = sc.nextInt();

int count = 0;

while(n>0){

long rem = n%10;

if(rem == d)

count++;

n /= 10;

}

System.out.println("The count of the digit "+d+" is "+ count);

}

}

82. Write a program in Java to Toggle the case of every character of a string.For instance, if the input string is “ApPLe”, the output should be “aPplE”.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

// Input the string

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

String inputString = scanner.nextLine();

// Toggle the case of each character and build the result string

StringBuilder result = new StringBuilder();

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

char currentChar = inputString.charAt(i);

if (Character.isUpperCase(currentChar)) {

result.append(Character.toLowerCase(currentChar));

}

else if (Character.isLowerCase(currentChar)) {

result.append(Character.toUpperCase(currentChar));

}

else {

// If the character is not a letter, keep it unchanged

result.append(currentChar);

}

}

// Output the result

System.out.println("Toggled case string: " + result.toString());

}

}

83.write a java program to Merge the two Arrays.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

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

int [] b = {5,6,7,8,9,10};

int [] res = new int [a.length+b.length];

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

res[i]=a[i];

}

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

res[i+a.length]=b[i];

}

System.out.println(Arrays.toString(res));

}

}

84. print 1 to n numbers using foreach loop

\* In Java, you can't directly use a foreach loop to print numbers from 1 to n,

\* as foreach loops are primarily used for iterating through arrays or collections.

\* However, you can use a regular for loop to achieve this task.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the size");

int n = sc.nextInt();

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

System.out.print(i+" ");

}

}

}

85.Write a Java Program to reverse a string with using String inbuilt function reverse().

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the Name");

String s = sc.nextLine();

StringBuilder s1 = new StringBuilder(s);

s1.reverse();

System.out.println(s1);

}

}

86.find the given element that is present in an array or not, by using Binary Search.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the element to search");

int n = sc.nextInt();

int [] a = {1,2,3,4,5,6};

int res = Arrays.binarySearch(a, n);

if (res<0) {

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

} else {

System.out.println("The element found at the index: "+res);

}

}

}

87. convert the given integer into a binary number format.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the number");

int n = sc.nextInt();

String binary = Integer.toBinaryString(n);

System.out.println("Binary representation of " + n + " is: " + binary);

}

}

88. convert the given integer into a binary number format.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int[] array = { -40, -5, 1, 3, 6, 7, 8, 20 };

int targetSum = 15;

findPairsWithSum(array, targetSum);

}

public static void findPairsWithSum(int[] array, int targetSum) {

for (int i = 0; i < array.length - 1; i++) {

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

if (array[i] + array[j] == targetSum) {

System.out.println("Pair of elements whose sum is equal to " +

targetSum + ": " + array[i] + ", " + array[j]);

}

}

}

}

}

89. convert the given integer into a binary number format.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int[][] matrix = {

{5, 2, 9, 1},

{8, 3, 7, 6},

{4, 5, 1, 2}

};

findSmallestInEachRow(matrix);

}

public static void findSmallestInEachRow(int[][] matrix) {

for (int row = 0; row < matrix.length; row++) {

int smallest = matrix[row][0]; // Initialize the smallest with the first element of the

row

for (int col = 1; col < matrix[row].length; col++) {

if (matrix[row][col] < smallest) {

smallest = matrix[row][col]; // Update smallest if a smaller element is

found

}

}

System.out.println("Smallest element in row " + (row + 1) + ": " + smallest);

}

}

90. perfor XOR operation on two integers.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the num1 value");

int num1 = sc.nextInt(); // First integer

System.out.println("Enter the num2 value");

int num2 = sc.nextInt(); // Second integer

int result = num1 ^ num2; // Perform XOR operation

System.out.println("Result of " + num1 + " XOR " + num2 + " is: " + result);

}

}

91. From the given input separate numbers and characters

Example- input:

j34784ha

output:

jha

3478

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

String s1 = "j34784ha";

String numbers = "";

String characters = "";

for (int i = 0; i < s1.length(); i++) {

char ch = s1.charAt(i);

if (Character.isDigit(ch)) {

numbers += ch;

} else if (Character.isLetter(ch)) {

characters += ch;

}

}

System.out.println("Numbers: " + numbers);

System.out.println("Characters: " + characters);

}

}

92.write a program to find the sum of odd integers in an array and prduct of even integers in array.finally find sum of the both of the results.

example:

input:[3,4,5,2,1,7]

output:

odd sum=3+5+1+7=16

even product=4+2=6

final sum=22

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

int [] a = {3,4,5,2,1,7};

int even = 0;

int odd = 0;

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

if (a[i]%2==0) {

even += a[i];

}

else {

odd += a[i];

}

}

System.out.println("The even sum in the array is: "+even);

System.out.println("The odd sum in the array is: "+odd);

System.out.println("The addition of both odd and even sum is: "+(odd+even));

}

}

93.find the given input number is perfect number or not.The perfect number is the number where the sum of its divisors is equal to the number itself.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

long n, sum=0;

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

n=sc.nextLong();

int i=1;

while(i <= n/2)

{

if(n % i == 0)

{

//calculates the sum of factors

sum = sum + i;

}

i++;

}

//compares sum with the number

if(sum==n)

{

//prints if sum and n are equal

System.out.println(n+" is a perfect number.");

}

else

//prints if sum and n are not equal

System.out.println(n+" is not a perfect number.");

}

}

94.Replace all zero's with one's and one's with zero's.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

String input = "010101001011";

String result = "";

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

if (input.charAt(i) == '0') {

result += '1'; // Replace '0' with '1'

} else if (input.charAt(i) == '1') {

result += '0'; // Replace '1' with '0'

}

}

System.out.println("Original: " + input);

System.out.println("Replaced: " + result);

}

}

95.Sort the first half of the array elements in ascending order second half of the array elements in descending order.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

static void printOrder(int[] arr, int n)

{

// sorting the array

Arrays.sort(arr);

// printing first half in ascending order

for (int i = 0; i < n / 2; i++)

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

// printing second half in descending order

for (int j = n - 1; j >= n / 2; j--)

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

}

// Main

public static void main(String[] args)

{

int[] arr = { 5, 4, 6, 2, 1, 3, 8, 9, 7,10 };

int n = arr.length;

printOrder(arr, n);

}

}

\*/

96.find the most repeated word in a sentence.

package assignmenttest;

import java.util.Arrays;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter the string");

String s1 = sc.nextLine();

char[] str = s1.toCharArray();

int len[] = new int [s1.length()];

for (int i = 0; i < s1.length(); i++) {

len[i]=1;

for (int j = i+1; j < s1.length(); j++) {

if (str[i]==str[j]) {

len[i]++;

str[j]=0;

}

}

}

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

if (str[i]==' '||str[i]==0) {

} else {

System.out.println(str[i]+"---->"+len[i]);

}

}

}

}

97.check number belongs to Fibonacci series or not.

package assignmenttest;

import java.util.Scanner;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

System.out.println("Enter a number");

// read the number to be checked

int numberToCheck = sc.nextInt();

int firstNumber = 0, secondNumber = 1, fibonacciNumber = 0;

// loop till the current fibonacci number is less than the number to

// check

while (fibonacciNumber < numberToCheck) {

// calculate the next fibonacci number

fibonacciNumber = firstNumber + secondNumber;

// move the fibonacci series ahead

firstNumber = secondNumber;

secondNumber = fibonacciNumber;

}

// compare the current fibonacci number with number to check

if (numberToCheck == fibonacciNumber) {

System.out.println("Number belongs to Fibonacci series");

} else {

System.out.println("Number does not belong to Fibonacci series");

}

}

}

98.Remove an element at specific index from an array.

package assignmenttest;

import java.util.\*;

public class Launch {

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

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

System.out.println("Before deletion :" + Arrays.toString(arr));

System.out.println("Enter the index to delete the element");

int remove = sc.nextInt();

int[] arr\_new = new int[arr.length-1];

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

if(i!=remove){

arr\_new[k]=arr[i];

k++;

}

}

System.out.println("After deletion :" + Arrays.toString(arr\_new));

}

}

99.check whether the given string is panagram or not.

package assignmenttest;

import java.util.\*;

public class Launch {

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

System.out.println("Enter Your String:");

String str=sc.nextLine();

str=str.replaceAll("","").toLowerCase();

// empty string

String s="";

// checking characters (a-z or A-Z)

for(char i='a';i<='z';i++){

//indexOf(char i)--> This method returns '-1' substring not found, if the position of

substrings 'i' in 'str'

if(str.indexOf(i)!=-1){

s=s+i;// empty string+character

}

}

if(s.length()==26){

System.out.println("Pangram");

}

else{

System.out.println("Not Pangram");

}

}

}

100. find sum of prime numbers in a given range.

package assignmenttest;

import java.util.\*;

public class Launch {

public static void main(String args[])

{

int number = 1, count, sum = 0;

while(number <= 100)

{

count = 0;

int i = 2;

while(i <= number/2 )

{

if(number % i == 0)

{

count++;

break;

}

i++;

}

if(count == 0 && number != 1 )

{

sum = sum + number;

}

number++;

}

System.out.println("The Sum of Prime Numbers from 1 to 100 is: " + sum);

}

}