MEDIUM

1.COUNT PRIME AND COMPOSITE NUMBERS

CODE:

import java.util.Scanner;

import java.lang.Math;

public class InvertedPyramidPattern

{

public static void main(String args[])

{

int arr[]={4,54,29,71,7,59,98,23};

int com=0,pri=0;

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

{

int c=0;

for(int j=1;j<arr[i];j++)

{

if(arr[i]%j==0)

c++;

}

if(c>1)

com++;

else

pri++;

}

System.out.print("Composite Number: "+com);

System.out.println("\nPrime number: "+pri);

}

}

OUTPUT:

Composite Number: 3

Prime number: 5

2.FIND MTH MAXIMUM NUMBER AND NTH MINIMUM NUMBER AND THEN FIND SUM AND DIFFERENCE OF IT

CODE:

import java.util.Scanner;

import java.lang.Math;

public class InvertedPyramidPattern

{

public static void main(String args[])

{

int arr[]={14, 16, 87, 36, 25, 89, 34};

int len=arr.length;

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

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

if (arr[i] > arr[j]) {

int temp = arr[i];

arr[i] = arr[j];

arr[j] = temp;

}

}

}

int m=1,n=3;

int max=arr[len-m];

int min=arr[n-1];

System.out.print(m+" maximum number = "+max);

System.out.print("\n"+n+" minimum number = "+min);

int sum=max+min;

int Diff=max-min;

System.out.print("\nSum = "+sum);

System.out.print("\nDifference = "+Diff);

}

}

OUTPUT:

1 maximum number = 89

3 minimum number = 25

Sum = 114

Difference = 64

3.CHECKING BALANCE IN AN ATM

CODE:

import java.util.\*;

public class BalanceAtm

{

public static void main(String[]args)

{

int n1=500,d1=4,n2=100,d2=20,n3=200,d3=32,n4=2000,d4=1;

int Total=(n1\*d1)+(n2\*d2)+(n3\*d3)+(n4\*d4);

System.out.print("Total Available Balance in ATM: "+Total);

}

}

OUTPUT:

Total Available Balance in ATM: 12400

4.check whether a string or number or string is palindrome or not

CODE:

import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.println("Enter 1 to check if a string is a palindrome");

System.out.println("Enter 2 to check if a number is a palindrome");

int choice = scanner.nextInt();

switch (choice) {

case 1:

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

String inputString = scanner.next();

String reversedString = new StringBuilder(inputString).reverse().toString();

System.out.println(inputString.equals(reversedString) ? "Palindrome" : "Not a Palindrome");

break;

case 2:

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

int num = scanner.nextInt();

int originalNum = num;

int reversedNum = 0;

while (num > 0) {

reversedNum = reversedNum \* 10 + num % 10;

num /= 10;

}

System.out.println(originalNum == reversedNum ? "Palindrome" : "Not a Palindrome");

break;

default:

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

}

}

}

OUTPUT:

Enter 1 to check if a string is a palindrome

Enter 2 to check if a number is a palindrome

1

Enter the string: madam

Palindrome

5.CONVERSION OF DECIMAL TO BINARY AND OCTAL NUMBER

CODE:

import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

int dec=15;

String bin=Integer.toBinaryString(dec);

String oct=Integer.toOctalString(dec);

System.out.println("Binary number = "+bin);

System.out.print("octal number = "+oct);

}

}

OUTPUT:

Binary number = 1111

octal number = 17

6.CALCULATE BONUS AND SALARY OF THE EMPLOYEE TO BE PAID

CODE:

import java.util.Scanner;

public class PalindromeChecker {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int a,b;

double bonus=0;

System.out.print("Enter the grade of the employee :");

char a1=input.next().charAt(0);

System.out.print("Enter the salary of employee :");

int b1=input.nextInt();

if(a1=='A')

{

bonus=b1\*(0.05);

if(b1<10000)

{

bonus=bonus+b1\*(0.02);

}

System.out.println("salary = "+b1);

System.out.println("bonus = "+bonus);

System.out.println("total to be paid ="+(b1+bonus));

}

else if(a1=='B')

{

bonus=b1\*(0.1);

if(b1<10000)

{

bonus=bonus+b1\*(0.02);

}

System.out.println("salary = "+b1);

System.out.println("bonus = "+bonus);

System.out.println("total to be paid ="+(b1+bonus));

}

else {

System.out.print("Enter valid grade");

}

}

}

OUTPUT:

Enter the grade of the employee :A

Enter the salary of employee :100000

salary = 100000

bonus = 5000.0

total to be paid =105000.0

7.FIRST N PERFECT NUMBERS

CODE:

import java.util.Scanner;

public class PefectNumber{

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int sum=0,temp=0;

for(int j=2;j<=1000;j++)

{

if(n>temp)

sum=1;

for(int i=2;i<j;i++)

{

if(j%i==0)

sum=sum+i;

}

if(sum==j)

{

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

temp=temp+1;

}

}

}

}

OUTPUT:

3

6 28 496

8.TOTAL AND AGGREAGRATE OF A STUDENTS BY USING THE MARKS

CODE:

import java.util.Scanner;

public class Aggregate{

public static void main(String[] args) {

int a1=90;

int a2=91;

int a3=92;

int a4=93;

int total=(a1+a2+a3+a4) ;

float agg=total/4f;

System.out.println(total);

System.out.println(agg);

if(agg>75)

System.out.println("DISTINCTION");

else if(agg>=60 && agg<75)

System.out.println("First Division");

else if(agg>=50 && agg<60)

System.out.println("Second Division");

else if(agg>=40 && agg<50)

System.out.println("Third Division");

else System.out.println("Fail");

}

}

OUTPUT:

366

91.5

DISTINCTION

9.CALCULATE TAX

CODE:

import java.util.Scanner;

public class Aggregate{

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int income=input.nextInt();

float tax;

if(income<=150000)

System.out.println("No tax");

else if(income>=150001 && income<=300000)

System.out.println("Tax= "+income/10);

else if(income>=300001 && income<=500000)

System.out.println("Tax= "+income/20);

else

System.out.println("Tax= "+income/30);

}

}

OUTPUT:

2000000

Tax= 66666

10.MULTIPLICATION TABLE

CODE:

import java.util.Scanner;

public class Aggregate{

public static void main(String[] args) {

int M=2;

int N=10;

for(int i=1;i<=N;i++)

{

System.out.println(i+"x"+M+"="+(i\*M));

}

}

}

OUTPUT:

1x2=2

2x2=4

3x2=6

4x2=8

5x2=10

6x2=12

7x2=14

8x2=16

9x2=18

10x2=20

11.AVERAGE OF POSITIVE AND NEGATIVE NUMBERS

CODE:

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

double sumPositive = 0;

double sumNegative = 0;

int countPositive = 0;

int countNegative = 0;

while (true) {

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

int number = scanner.nextInt();

if (number == -1) {

break;

}

if (number > 0) {

sumPositive += number;

countPositive++;

} else if (number < 0) {

sumNegative += number;

countNegative++;

}

}

if (countPositive > 0) {

double averagePositive = sumPositive / countPositive;

System.out.printf("The average of positive numbers is: %.8f%n", averagePositive);

} else {

System.out.println("No positive numbers were entered.");

}

if (countNegative > 0) {

double averageNegative = sumNegative / countNegative;

System.out.printf("The average of negative numbers is: %.1f%n", averageNegative);

} else {

System.out.println("No negative numbers were entered.");

}

}

}

OUTPUT:

Enter the number: 1

Enter the number: 2

Enter the number: 3

Enter the number: -8

Enter the number: -9

Enter the number: 6

Enter the number: 9

Enter the number: 3

Enter the number: 7

Enter the number: 2

Enter the number: 4

Enter the number: -1

The average of positive numbers is: 4.11111111

The average of negative numbers is: -8.5

12.COUNT NUMBER OF LOWERCASE, UPPERCASE AND NUMBERS

CODE:

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

System.out.println("Enter \* to exit....");

char c='0';

int lower=0,upper=0,digit=0;

while(c!='\*')

{

c=input.next().charAt(0);

if(c>=65 && c<=90)

upper=upper+1;

else if(c>=97 && c<=122)

lower=lower+1;

else if(c>=48 && c<=57)

digit=digit+1;

}

System.out.println("Lower: "+lower);

System.out.println("Upper: "+upper);

System.out.println("Digit: "+digit);

}

}

OUTPUT:

Enter \* to exit....

A

B

s

h

7

k

S

\*

Lower: 3

Upper: 3

Digit: 1

13.FACTORIAL OF A NUMBER

CODE:

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int fact=1;

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

{

fact=fact\*i;

}

System.out.println("The factorial of "+n+" is: "+fact);

}

}

OUTPUT:

6

The factorial of 6 is: 720

14.Nth LARGEST NUMBER IN AN ARRAY

CODE:

import java.util.Arrays;

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner input = new Scanner(System.in);

int a[] = {14, 67, 48, 23, 5, 62};

int len = a.length;

Arrays.sort(a);

int N = 4;

if (N > 0 && N <= len) {

System.out.println(N + "th Largest number: " + a[len - N]);

} else {

System.out.println("Invalid value of N. It must be between 1 and " + len);

}

}

}

OUTPUT:

4th Largest number: 23

15.DUPLICATES

CODE:

import java.util.Arrays;

import java.util.Scanner;

public class AverageCalculator {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

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

int n=input.nextInt();

int a[]=new int[n];

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

{

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

a[i]=input.nextInt();

}

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

{

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

{

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

{

for(int k=j;k<n-1;k++)

{

a[k]=a[k+1];

}

j--;

n--;

}

}

}

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

{

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

}

}

}

OUTPUT:

Enter the number of elements: 5

Enter element: 10

Enter element: 20

Enter element: 20

Enter element: 10

Enter element: 30

10 20 30

16.RATE OF INTERSEST FOR BANK

CODE:

import java.util.HashMap;

import java.util.Map;

import java.util.Scanner;

abstract class Bank {

abstract double getRateOfInterest();

}

class SBI extends Bank {

@Override

double getRateOfInterest() {

return 8.4;

}

}

class ICICI extends Bank {

@Override

double getRateOfInterest() {

return 7.3;

}

}

class AXIS extends Bank {

@Override

double getRateOfInterest() {

return 9.7;

}

}

public class BankInterestRate {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Bank sbi = new SBI();

Bank icici = new ICICI();

Bank axis = new AXIS();

Map<String, Bank> bankMap = new HashMap<>();

bankMap.put("SBI", sbi);

bankMap.put("ICICI", icici);

bankMap.put("AXIS", axis);

System.out.println("Enter bank name and rate of interest (e.g., SBI, 8.4):");

String input = scanner.nextLine();

String[] inputParts = input.split(",\\s\*");

if (inputParts.length == 2) {

String bankName = inputParts[0];

double rate = Double.parseDouble(inputParts[1]);

Bank bank = bankMap.get(bankName);

if (bank != null) {

System.out.println(bankName + ", " + bank.getRateOfInterest());

} else {

System.out.println("Invalid bank name");

}

} else {

System.out.println("Invalid input format");

}

}

}

OUTPUT:

Enter bank name and rate of interest (e.g., SBI, 8.4):

SBI,8.7

SBI, 8.4

17.MULTIPLICATION TABLE USING DEADLOCK

CODE:

import java.util.Scanner;

class A extends Thread

{

public void run()

{

int n=5;

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

{

System.out.println(n+" X "+i+" = "+(n\*i));

}

}

}

class B extends Thread

{

public void run()

{

int n=10;

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

{

System.out.println(n+" X "+i+" = "+(n\*i));

}

}

}

public class ak

{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

A threadA=new A();

B threadB=new B();

threadA.start();

threadB.start();

}

}

OUTPUT:

10 X 1 = 10

10 X 2 = 20

10 X 3 = 30

5 X 1 = 5

10 X 4 = 40

10 X 5 = 50

5 X 2 = 10

10 X 6 = 60

10 X 7 = 70

10 X 8 = 80

10 X 9 = 90

5 X 3 = 15

5 X 4 = 20

5 X 5 = 25

10 X 10 = 100

18.FIBONACCI SERIES USING THREADS

CODE:

import java.util.Scanner;

class FibonacciRunnable implements Runnable {

private int count;

public FibonacciRunnable(int count) {

this.count = count;

}

@Override

public void run() {

if (count <= 0) {

System.out.println("Number of terms must be positive.");

return;

}

System.out.print("Fibonacci series: ");

int a = 0, b = 1;

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

if (count > 1) {

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

}

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

int next = a + b;

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

a = b;

b = next;

}

System.out.println();

}

}

public class FibonacciSeries {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter the number of terms for Fibonacci series: ");

String input = scanner.nextLine();

try {

int numberOfTerms = Integer.parseInt(input);

if (numberOfTerms < 0) {

System.out.println("Negative numbers are not allowed.");

} else {

FibonacciRunnable fibonacciRunnable = new FibonacciRunnable(numberOfTerms);

Thread fibonacciThread = new Thread(fibonacciRunnable);

fibonacciThread.start();

try {

fibonacciThread.join();

} catch (InterruptedException e) {

System.out.println("Thread was interrupted.");

}

}

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter a positive integer.");

} finally {

scanner.close();

}

}

}

OUTPUT:

Enter the number of terms for Fibonacci series: 4

Fibonacci series: 0 1 1 2

19.SUM OF N NUMBERS IN AN ARRAY USING THREADS

CODE:

import java.util.Scanner;

public class SumOfNumbers {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

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

int N;

try {

N = Integer.parseInt(scanner.nextLine());

if (N <= 0) {

throw new IllegalArgumentException("Number of elements must be positive.");

}

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter a positive integer.");

scanner.close();

return;

}

int[] numbers = new int[N];

System.out.println("Enter " + N + " numbers:");

try {

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

numbers[i] = Integer.parseInt(scanner.nextLine());

}

// Calculate the sum and simulate ArrayIndexOutOfBoundsException

int sum = 0;

for (int i = 0; i <= N; i++) { // Loop until N, causing an out-of-bounds access on the last iteration

if (i >= numbers.length) {

throw new ArrayIndexOutOfBoundsException("Index " + i + " is out of bounds for array of size " + numbers.length);

}

sum += numbers[i];

}

System.out.println("Sum of numbers: " + sum);

} catch (ArrayIndexOutOfBoundsException e) {

System.out.println("Exception: " + e.getMessage());

} catch (NumberFormatException e) {

System.out.println("Invalid number input. Please enter integer values.");

} finally {

scanner.close();

}

}

}

20.PRIME OR NOT USING THREADS

CODE:

import java.util.Scanner;

public class PrimeCheckerThread implements Runnable {

private final int number;

public PrimeCheckerThread(int number) {

this.number = number;

}

@Override

public void run() {

if (number < 2) {

System.out.println(number + " is not Prime");

return;

}

boolean isPrime = true;

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

if (number % i == 0) {

isPrime = false;

break;

}

}

if (isPrime) {

System.out.println(number + " is Prime");

} else {

System.out.println(number + " is not Prime");

}

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

System.out.print("Enter a number to check if it's prime: ");

String input = scanner.nextLine();

try {

int number = Integer.parseInt(input);

Thread thread = new Thread(new PrimeCheckerThread(number));

thread.start();

thread.join();

} catch (NumberFormatException e) {

System.out.println("Invalid input. Please enter an integer.");

} catch (InterruptedException e) {

System.out.println("Thread interrupted.");

} finally {

scanner.close();

}

}

}

OUTPUT:

Enter a number to check if it's prime: 67

67 is Prime

21.COUNT NO OF CHARACTERS IN THE STRING

CODE:

import java.util.Arrays;

import java.util.Locale;

import java.util.Scanner;

public class ak {

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

String name=input.next();

name.split("NULL");

name=name.trim();

int len=0;

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

{

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

break;

else

len++;

}

System.out.println(len);

}

}

OUTPUT:

HELLO

5