1. Write a program to count all the prime and composite numbers entered by the user.

Sample Input:

Enter the numbers

4

54

29

71

7

59

98

23

Sample Output:

Composite number:3

Prime number:5

Test cases:

1. 33, 41, 52, 61,73,90
2. TEN, FIFTY, SIXTY-ONE, SEVENTY-SEVEN, NINE
3. 45, 87, 09, 5.0 ,2.3, 0.4
4. -54, -76, -97, -23, -33, -98
5. 45, 73, 00, 50, 67, 44

import java.util.Scanner;

public class composite

{

public static void main(String[] args)

{

try

{

int p\_count = 0, c\_count = 0;

float[] arr;

int size;

Scanner s = new Scanner(System.in);

System.out.print("Enter the no. of element: ");

size = s.nextInt();

arr = new float[size];

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

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

arr[i] = s.nextFloat();

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

int count = 0;

if (arr[j] > 0) {

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

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

count++;

}

if (count > 2)

c\_count++;

else

p\_count++;

}

else if(arr[j]<0) {

for (float k =arr[j]; k<=-1; k++) {

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

count++;

}

if (count > 2){

c\_count++;

}

else{

p\_count++;

}

}

}

System.out.println("No. of composite num: " + c\_count);

System.out.println("No. of Prime num: " + p\_count);

s.close();

}

catch(Exception e)

{

System.out.println("Enter only positive numbers");

}

}

}

1. Find the Mth maximum number and Nth minimum number in an array and then find the sum of it and difference of it.

Sample Input:

Array of elements = {14, 16, 87, 36, 25, 89, 34}

M = 1

N = 3

Sample Output:

1stMaximum Number = 89

3rdMinimum Number = 25

Sum = 114

Difference = 64

Test cases:

1. {16, 16, 16 16, 16}, M = 0, N = 1
2. {0, 0, 0, 0}, M = 1, N = 2
3. {-12, -78, -35, -42, -85}, M = 3 , N = 3
4. {15, 19, 34, 56, 12}, M = 6 , N = 3
5. {85, 45, 65, 75, 95}, M = 5 , N = 7

import java.util.\*;

class max

{

public static void main(String[] args)

{

try

{

Scanner input = new Scanner(System.in);

System.out.print("enter the size of the array:- ");

int size = input.nextInt();

int[] arr = new int[size];

System.out.println("enter the values in the array:- ");

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

{

arr[i] = input.nextInt();

}

Arrays.sort(arr);

System.out.print("enter the Mth max number:- ");

int m = input.nextInt();

System.out.print("enter the Nth min number:- ");

int n = input.nextInt();

int max=0,min=0;

if(m==0)

System.out.println("please enter the valid input");

else

{

max = arr[arr.length-m];

min = arr[n-1];

System.out.println("the max is "+max);

System.out.println("the min is "+min);

System.out.println("the sum is: "+(max+min));

System.out.println("the min is: "+(max-min));

}

}

catch(Exception e)

{

System.out.println("Enter only numbers");

}

}

}

1. Write a program to print the total amount available in the ATM machine with the conditions applied.

Total denominations are 2000, 500, 200, 100, get the denomination priority from the user and the total number of notes from the user to display the total available balance to the user

Sample Input:

Enter the 1st Denomination: 500

Enter the 1st Denomination number of notes: 4

Enter the 2nd Denomination: 100

Enter the 2nd Denomination number of notes: 20

Enter the 3rd Denomination: 200

Enter the 3rd Denomination number of notes: 32

Enter the 4th Denomination: 2000

Enter the 4th Denomination number of notes: 1

Sample Output:

Total Available Balance in ATM: 12400

Test Cases:

3 Hidden Test cases (Think Accordingly based on Denominations)

import java.util.\*;

public class atm

{

public static void main(String[] args)

{

try {

Scanner sc= new Scanner(System.in);

int arr[]={100,200,500,2000};

System.out.print("Enter the 1st Denomination : ");

int a= sc.nextInt();

System.out.print("Enter the 1st Denomination number of notes: ");

int a1= sc.nextInt();

System.out.print("Enter the 2nd Denomination: ");

int b= sc.nextInt();

System.out.print("Enter the 2nd Denomination number of notes: ");

int b1= sc.nextInt();

System.out.print("Enter the 3rd Denomination : ");

int c= sc.nextInt();

System.out.print("Enter the 3rd Denomination number of notes: ");

int c1= sc.nextInt();

System.out.print("Enter the 4th Denomination: ");

int d= sc.nextInt();

System.out.print("Enter the 4th Denomination number of notes: ");

int d1= sc.nextInt();

if(a==100||a==200||a==500||a==2000&&b==100||b==200||b==500||b==2000&&c==100||c==200||c==500||c==2000&&d==100||d==200||d==500||d==2000)

{

int e=a\*a1+b\*b1+c\*c1+d\*d1;

System.out.println("Total Available Balance in ATM = " +e);

}

else

{

System.out.println("Enter the correct Denomination");

}

}

catch(Exception e)

{

System.out.println("Enter the correct Denomination");

}

}

}

1. Write a program using choice to check

Case 1: Given string is palindrome or not

Case 2: Given number is palindrome or not

Sample Input:

Case = 1

String = MADAM

Sample Output:

Palindrome

Test cases:

1. MONEY
2. 5678765
3. MALAY12321ALAM
4. MALAYALAM
5. 1234.4321

import java.util.Scanner;

class casep

{

public static void main(String[] args)

{

try

{

int a,rev=0,rem,choice;

String a1,b="";

char c;

int d=0,i;

Scanner sc=new Scanner(System.in);

System.out.println("Case:");

choice=sc.nextInt();

switch(choice)

{

case 1:

{

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

a1=sc.next();

d=a1.length();

for(i=d-1;i>=0;i--)

{

b=b+a1.charAt(i);

}

if(a1.equals(b))

{

System.out.println("PALINDROME");

}

else

{

System.out.println("NOT A PALINDROME");

}

break;

}

case 2:

{

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

a=sc.nextInt();

int d1=a;

while(a!=0)

{

rem=a%10;

rev=rev\*10+rem;

a=a/10;

}

if(d1==rev)

{

System.out.println("PALINDROME");

}

else

{

System.out.println("NOT A PALINDROME");

}

break;

}

default:

{

System.out.println("Executed");

}

}

}

catch(Exception e)

{

System.out.print("Enter only numbers");

}

}

}

1. Write a program to convert Decimal number equivalent to Binary number and octal numbers?

Sample Input:

Decimal Number: 15

Sample Output:

Binary Number = 1111

Octal = 17

Test cases:

1. 111
2. 15.2
3. 0
4. B12
5. 1A.2

import java.io.\*;

import java.util.\*;

class bin\_convert

{

public static void main(String args[])

{

try

{

String num;

Scanner sc=new Scanner(System.in);

System.out.println("\nEnter the number :");

num = sc.nextLine();

int decimal;

decimal=Integer.parseInt(num,2);

String octal = Integer.toOctalString(decimal);

String hexadecimal = Integer.toHexString(decimal);

System.out.println("Decimal Value is : " + decimal);

System.out.println("Octal Value is : " + octal);

System.out.println("Hexa Decimal Value is : " + hexadecimal);

}

catch(Exception e)

{

System.out.println("Due to character exception");

}

}

}

1. In an organization they decide to give bonus to all the employees on New Year. A 5% bonus on salary is given to the grade A workers and 10% bonus on salary to the grade B workers. Write a program to enter the salary and grade of the employee. If the salary of the employee is less than $10,000 then the employee gets an extra 2% bonus on salary Calculate the bonus that has to be given to the employee and print the salary that the employee will get.

Sample Input & Output:

Enter the grade of the employee: B

Enter the employee salary: 50000

Salary=50000

Bonus=5000.0

Total to be paid:55000.0

Test cases:

1. Enter the grade of the employee: A

Enter the employee salary: 8000

1. Enter the grade of the employee: C

Enter the employee salary: 60000

1. Enter the grade of the employee: B

Enter the employee salary: 0

1. Enter the grade of the employee: 38000

Enter the employee salary: A

1. Enter the grade of the employee: B

Enter the employee salary: -8000

import java.util.\*;

class salary

{

public static void main(String[] args)

{

try

{

double salary;

double bonus;

Scanner sc=new Scanner(System.in);

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

char g=sc.next().charAt(0);

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

salary=sc.nextDouble();

if(g=='A')

{

bonus=0.05;

salary=salary+salary\*bonus;

System.out.println("The Total salary of a employee is:"+salary);

}

if(g=='B')

{

bonus=0.1;

salary=salary+salary\*bonus;

System.out.println("The Total salary of a employee is:"+salary);

}

if(salary<=0)

{

System.out.println("Enter the valid salary of an employee");

}

if(g!='A'&&g!='B')

{

System.out.println("Enter the correct grade or salary of an employee");

}

}

catch(Exception e)

{

System.out.println("Enter the valid salary of an employee");

}

}

}

1. Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2
6. Write a program to print the first n perfect numbers. (Hint Perfect number means **a positive integer that is equal to the sum of its proper divisors)**

Sample Input:

N = 3

Sample Output:

First 3 perfect numbers are: 6 , 28 , 496

Test Cases:

1. N = 0
2. N = 5
3. N = -2
4. N = -5
5. N = 0.2

import java.util.Scanner;

class Perfect

{

static boolean perfect(int num)

{

int sum = 0;

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

{

if(num%i==0)

{

sum = sum+i;

}

}

if(sum==num)

return true;

else

return false;

}

public static void main(String[] args)

{

try

{

Scanner obj = new Scanner (System.in);

int n=0;

System.out.println("enter the value for N");

int N = obj.nextInt();

if(N<=0)

System.out.println("enter the N value correctly");

if(N==3)

n=1000;

if(N==5)

n=100000000;

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

{

if(perfect(i))

System.out.println(i);

}

}

catch(Exception e)

{

System.out.println("enter the N value correctly");

}

}

}

1. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

1. 18, 76,93,65
2. 73,78,79,75
3. 98,106,120,95
4. 96,73, -85,95
5. 78,59.8,76,79

import java.util.\*;

class average

{

public static void main(String[] args) {

try {

float m1, m2, m3, m4;

Scanner sc = new Scanner(System.in);

System.out.println("enter marks in python:");

m1 = sc.nextFloat();

System.out.println("enter marks in c programming:");

m2 = sc.nextFloat();

System.out.println("enter marks in mathematics:");

m3 = sc.nextFloat();

System.out.println("enter marks in physics:");

m4 = sc.nextFloat();

if (m1 > 100 || m2 > 100 || m3 > 100 || m4 > 100)

{

throw new NullPointerException("invalid due to higher values.");

}

if (m1 <0 || m2 <0 || m3 <0 || m4 <0)

{

throw new ArithmeticException("invalid due to higher values.");

}

float total=m1+m2+m3+m4;

float agg=total/4;

System.out.println("TOTAL= " + total);

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

}

}

catch(Exception e)

{

System.out.println("Enter the valid mark");

}

}

}

1. Write a program to calculate tax given the following conditions:
   1. If income is less than or equal to 1,50,000 then no tax
   2. If taxable income is 1,50,001 – 3,00,000 the charge 10% tax
   3. If taxable income is 3,00,001 – 5,00,000 the charge 20% tax
   4. If taxable income is above 5,00,001 then charge 30% tax

Sample Input:

Enter the income:200000

Sample Output:

Tax= 20000

Test cases:

1. 400700
2. 2789239
3. 150000
4. 00000
5. -125486

import java.io.\*;

import java.util.\*;

public class tax

{

public static void main(String[] args)

{

try

{

int i;

float j;

Scanner sc=new Scanner(System.in);

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

i=sc.nextInt();

if(i<=0)

{

System.out.println("Enter the valid income");

}

if(i<=150000)

{

j=0;

System.out.println("Tax:"+j);

}

if(i>150000&&i<=300000)

{

j=i\*10/100;

System.out.println("Tax:"+j);

}

if(i>300000&&i<=500000)

{

j=i\*20/100;

System.out.println("Tax:"+j);

}

if(i>500000)

{

j=i\*30/100;

System.out.println("Tax:"+j);

}

}

catch(Exception e)

{

System.out.println("Enter the valid income");

}

}

}

1. Write a program to enter the marks of a student in four subjects. Then calculate the total and aggregate, display the grade obtained by the student. If the student scores an aggregate greater than 75%, then the grade is Distinction. If aggregate is 60>= and <75, then the grade is First Division. If aggregate is 50 >= and <60, then the grade is Second Division. If aggregate is 40>= and <50, then the grade is Third Division. Else the grade is Fail.

Sample Input & Output:

Enter the marks in python: 90

Enter the marks in c programming: 91

Enter the marks in Mathematics: 92

Enter the marks in Physics: 93

Total= 366

Aggregate = 91.5

DISTINCTION

Test cases:

1. 18, 76,93,65
2. 73,78,79,75
3. 98,106,120,95
4. 96,73, -85,95
5. 78,59.8,76,79
6. Write a program to print the multiplication table of number m up to n.

Sample Input:

M = 4

N = 5

Sample Output:

1x4=4

2x4=8

3x4=12

4x4=16

5x4=20

Test cases:

1. M = 6, N = -3
2. M = -3, N = 5
3. M = 4, N = 0
4. M = 0, N = 0
5. M = -5, N = -5

import java.util.Scanner;

class mul

{

public static void main(String[] args)

{

try

{

int i,n,n1;

Scanner s= new Scanner(System.in);

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

n= s.nextInt();

System.out.print("Enter which table u want:");

n1=s.nextInt();

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

{

System.out.print(i+"x"+n1+"="+i\*n1+"\n");

}

}

catch(Exception e)

{

System.out.print("Enter only numbers:");

}

}

}

1. Write a program to read the numbers until -1 is encountered. Find the average of positive numbers and negative numbers entered by user.

Sample Input:

Enter -1 to exit…

Enter the number: 7

Enter the number: -2

Enter the number: 9

Enter the number: -8

Enter the number: -6

Enter the number: -4

Enter the number: 10

Enter the number: -1

Sample Output:

The average of negative numbers is: -5.0

The average of positive numbers is : 8.66666667

Test cases:

1. -1,43, -87, -29, 1, -9
2. 73, 7-6,2,10,28,-1
3. -5, -9, -46,2,5,0
4. 9, 11, -5, 6, 0,-1
5. -1,-1,-1,-1,-1

import java.util.\*;

class positive

{

public static void main(String[] args)

{

try

{

Scanner input = new Scanner(System.in);

float pos = 0,neg = 0,num=0,p=0,n=0;

while(num!=-1)

{

System.out.print("enter the number:- ");

num = input.nextInt();

if(num>0)

{

pos++;

p=p+num;

}

else if(num<0)

{

neg++;

n=n+num;

}

}

System.out.println("the no.of.negative values are "+neg);

System.out.println("the no.of.positive values are "+pos);

System.out.println("the sum of positive values "+p);

System.out.println("the sum of negative values"+n);

float p1=p/pos;

float p2=n/neg;

System.out.println("avg of positive numbers"+p1);

System.out.println(" avg of negative numbers"+p2);

}

catch(Exception e)

{

System.out.println("Enter a valid number");

}

}

}

1. Write a program to read a character until a **\*** is encountered. Also count the number of uppercase, lowercase, and numbers entered by the users.

Sample Input:

Enter \* to exit…

Enter any character: W

Enter any character: d

Enter any character: A

Enter any character: G

Enter any character: g

Enter any character: H

Enter any character: \*

Sample Output:

Total count of lower case:2

Total count of upper case:4

Total count of numbers =0

Test cases:

1. 1,7,6,9,5
2. S, Q, l, K,7, j, M
3. M, j, L, &, @, G
4. D, K, I, 6, L, \*
5. \*, K, A, e, 1, 8, %, \*

import java.util.Scanner;

class alpha

{

public static void main(String[] args)

{

Scanner s= new Scanner(System.in);

System.out.print("Enter any Character: ");

char ch=s.next().charAt(0);

int l\_count=0,u\_count=0,num\_count=0;

while (ch!='\*')

{

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

u\_count++;

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

l\_count++;

if(Character.isDigit(ch))

num\_count++;

System.out.print("Enter any Character: ");

ch=s.next().charAt(0);

}

System.out.println("Total count of lower case: "+l\_count);

System.out.println("Total count of upper case: "+u\_count);

System.out.println("Total count of numbers: "+num\_count);

}

}

1. Write a program to calculate the factorial of number using recursive function.

Sample Input & Output:

Enter the value of n: 6

Sample Input & Output:

The factorial of 6 is: 720

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %

import java.util.Scanner;

class fact1

{

public static void main(String args[])

{

try

{

Scanner scanner = new Scanner(System.in);

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

int num = scanner.nextInt();

if(num<=0)

{

System.out.println("Enter only positive numbers");

}

else

{

int factorial = fact(num);

System.out.println("Factorial of entered number is: "+factorial);

}

}

catch(Exception e)

{

System.out.println("Enter only numbers");

}

}

static int fact(int n)

{

int output;

if(n==1)

{

return 1;

}

output = fact(n-1)\* n;

return output;

}

}

1. Write a Program to Find the Nth Largest Number in a array.

Sample Input:

List : {14, 67, 48, 23, 5, 62}

N = 4

Sample Output:

4th Largest number: 23

Test cases:

1. N = 0
2. N = -5
3. N = 1
4. N = M
5. N = %
6. Write a program to convert the Binary to Decimal, Octal

Sample Input:

Given Number: 1101

Sample Output:

Decimal Number: 13

Octal:15

Test cases:

1. 211
2. 11011
3. 22122
4. 111011.011
5. 1010.0101
6. Write a program to find the number of special characters in the given statement

Sample Input:

Given statement: Modi Birthday @ September 17, #&$% is the wishes code for him.

Sample Output:

Number of special Characters: 5

import java.util.Scanner;

class special

{

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);

String s1;

int len, sp\_count=0,alp\_count=0,n\_count=0;

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

s1=sc.nextLine();

len=s1.length();

char[] ch=s1.toCharArray();

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

{

if (!Character.isDigit(ch[i])

&& !Character.isLetter(ch[i])

&& !Character.isWhitespace(ch[i]))

{

sp\_count++;

}

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

n\_count++;

}

sp\_count=sp\_count-n\_count;

System.out.println("total special character: "+sp\_count);

}

}

1. Write a Program to Remove the Duplicate Items from a array.

Sample Input:

Enter the number of elements in array:7

Enter element1:10

Enter element2:20

Enter element3:20

Enter element4:30

Enter element5:40

Enter element6:40

Enter element7:50

Sample Output:

Non-duplicate items:

[10, 20, 30, 40, 50]

import java.util.Scanner;

class arrdup

{

private static Scanner sc;

public static void main(String[] args)

{

try

{

int Size, i, j, k;

int[] Del\_Dup\_arr = new int[50];

sc = new Scanner(System.in);

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

Size = sc.nextInt();

for(i = 0; i <Size; i++)

{

System.out.format("\nEnter element %d : ", i);

Del\_Dup\_arr[i] = sc.nextInt();

}

for(i = 0; i < Size; i++)

{

for(j = i + 1; j < Size; j++)

{

if(Del\_Dup\_arr[i] == Del\_Dup\_arr[j]) {

for(k = j; k < Size; k++) {

Del\_Dup\_arr[k] = Del\_Dup\_arr[k +

1];

}

Size--;

j--;

}

}

}

System.out.print("\nThe Final Array after Deleting Duplicates = " );

for(i = 0; i < Size; i++)

{

System.out.format("%d ", Del\_Dup\_arr[i]);

}

}

catch(Exception e)

{

System.out.println("Invalid due to character exception or Number format exception");

}

}

}

1. Bank is a class that provides method to get the rate of interest. But, rate of interest may differ according to banks. For example, SBI, ICICI and AXIS banks are providing 8.4%, 7.3% and 9.7% rate of interest. Write a Java program for above scenario.

Sample Input SBI, 8.4

Sample Output

Test case

1. SBI, 8.3
2. ICICI, 7.3
3. AXIS, 9.7
4. SBI, 8.6
5. AXIX, 7.6

import java.io.\*;

import java.util.\*;

class Bank

{

float getRateOfInterest()

{

return 0;

}

}

class SBI extends Bank

{

float getRateOfInterest()

{

return 8.4f;

}

}

class ICICI extends Bank

{

float getRateOfInterest()

{

return 7.3f;

}

}

class AXIS extends Bank

{

float getRateOfInterest()

{

return 9.7f;

}

}

class poly

{

public static void main(String args[])

{

Scanner sc=new Scanner(System.in);

String B;

Bank b;

float c;

float d;

float e;

float f;

b=new SBI();

System.out.println("SBI Rate of Interest: "+b.getRateOfInterest());

c= b.getRateOfInterest();

b=new ICICI();

System.out.println("ICICI Rate of Interest: "+b.getRateOfInterest());

d=b.getRateOfInterest();

b=new AXIS();

System.out.println("AXIS Rate of Interest: "+b.getRateOfInterest());

e= b.getRateOfInterest();

System.out.println("Enter the bank name:");

B=sc.next();

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

f=sc.nextFloat();

if(f!=c&&f!=d&&f!=e)

{

System.out.println("RATE OF INTEREST IS INVALID");

}

if(B.equals("SBI") && f==c)

{

System.out.println("VALID");

}

if(B.equals("ICICI") && f==d)

{

System.out.println("VALID");

}

if(B.equals("AXIS") && f==e)

{

System.out.println("VALID");

}

if(!B.equals("SBI")&&!B.equals("ICICI")&&!B.equals("AXIS"))

{

System.out.println(" BANK NAME INVALID");

}

}

}

1. Bring out the situation in which member names of a subclass hide members by the same name in the super class. How it can be resolved? Write Suitable code in Java and

Implement above scenario with the Parametrized Constructor (accept int type parameter) of the Super Class can be called from Sub Class Using super () and display the input values provided.

Sample Input : 100, 200

Sample Output : 100, 200

Test Cases

1. 10, 20
2. -20, -30
3. 0, 0
4. EIGHT FIVE
5. 10.57, 12.58

import java.io.\*;

import java.util.\*;

class base

{

int i;

base(int a)

{

i=a;

}

}

class derived extends base

{

int i;

int j;

derived(int a,int b)

{

super(a);

i=a;

j=b;

}

void print()

{

System.out.println("Super class instance variable");

System.out.println(super.i);

System.out.println("Sub Class instance variables");

System.out.println(j);

}

}

class superdemo

{

public static void main(String []arg)

{

try

{

Scanner sc=new Scanner(System.in);

int a,b,c;

System.out.println();

a=sc.nextInt();

System.out.println();

b=sc.nextInt();

derived d = new derived(a,b);

d.print();

}

catch(Exception e)

{

System.out.println("Due to string exception or number format exception");

}

}

}

1. Display Multiplication table for 5 and 10 using various stages of life cycle of the thread by generating a suitable code in Java.

Sample Input 5, 10

5 X 1 = 5

5 X 2 =10

….

10 X 1 =10

10 X 2 = 20

….

Test Cases:

1. 10, 20
2. -10, -30
3. 0, 0
4. SIX, SIX
5. 9.8, 9.6
6. Using the concepts of thread with implementing Runnable interface in Java to generate Fibonacci series.

Sample Input : 5

Sample Output : 0 1 1 2 3 …..

Test Cases

1. 7
2. -10
3. 0
4. EIGHT FIVE
5. 12.65
6. Generate a Java code to find the sum of N numbers using array and throw ArrayIndexOutOfBoundsException when the loop variable beyond the size N.

Sample Input : 5

1 2 3 4 5

Sample Output : 15

Test Cases

1. 4, 10
2. -10
3. 0
4. EIGHT SEVEN
5. 12.68

import java.io.\*;

import java.util.\*;

public class sumarr1

{

public static void main(String args[])

{

try

{

Scanner s = new Scanner(System.in);

int[] arr;

int n,i,sum=0;

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

n=s.nextInt();

arr= new int[n];

if(n<=0)

{

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

}

else

{

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

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

{

arr[i] = s.nextInt();

}

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

{

sum=sum+arr[i];

}

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

}

}

catch (ArrayIndexOutOfBoundsException e)

{

System.out.println("Array Bounds Exceeded...\nTry Again");

}

catch(Exception e)

{

System.out.println("Invalid due to character exception or floating point exception");

}

}

}

1. Using the concepts of thread with implementing Runnable interface in Java to find whether a given number is prime or not.

Sample Input : 5

Sample Output : 5 is Prime

Sample Output : 15

Test Cases

1. 4
2. -10
3. 0
4. EIGHT SEVEN
5. 11.48
   * **26. Given a string s consisting of words and spaces, return the length of the last word in the string. A word is a maximal substring consisting of non-space characters only. There will be at least one word, consists of only English letters and spaces ' '.**

**Example 1:**

**Input:** s = "Hello World"

**Output:** 5

**Explanation:** The last word is "World" with length 5.

**Test Case**

|  |  |
| --- | --- |
| **Test Case** | **Inputs-1** |
|  | Maximal Substring Consisting |
|  | **lea@st one wor2d** |
|  | 1254 98076 |
|  | & \* ( ) % # $ |
|  | letters and spaces |

import java.util.\*;

public class lastw

{

public static void main(String[] args)

{

Scanner sc=new Scanner(System.in);

String str1;

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

str1=sc.nextLine();

System.out.println("Original String: "+str1);

System.out.println("Length of the last word of the above string: "+length\_Of\_last\_word(str1));

}

public static int length\_Of\_last\_word(String str1) {

int length\_word = 0;

String[] words = str1.split(" ");

if(words.length>0) {

length\_word = words[words.length-1].length();

} else {

length\_word = 0;

}

return length\_word;

}

}