**11.Count of 3 Multiples**

Write a program to find the count of 3 multiples in a given input integer array.

Include a function named **divisibleBy3** that accepts 2 arguments and returns an int. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The function returns an int that corresponds to the count of 3 multiples.

If the size of the array is negative or if any element in the array is negative, print “Invalid Input” and terminate the program.

**Input and Output Format:**

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Output consists of an integer that corresponds to the count of 3 multiples

Assume that the maximum number of elements in the array is 20.

**Sample Input 1:**

8

1

6

3

5

61

80

102

9

**Sample Output 1:**

4

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

**Implementation:**

import java.util.Scanner;

public class Countofthreemultiples {

public static void main(String[] args) {

int n, i,count=0,flag=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0) {

System.out.print("Invalid array size");

System.exit(0);

}

else {

int a[]=new int[n];

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

{

a[i] = in.nextInt();

if(a[i] < 0) {

flag=1;

System.out.print("Invalid input");

System.exit(0);

}

}

if(flag!=1) {

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

if(a [i]%3==0 && a[i]!=0)

count++;

}

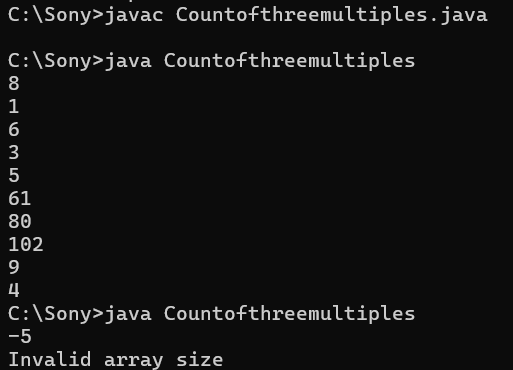
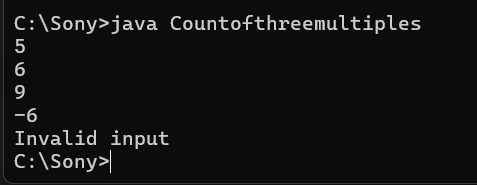
System.out.print(count);

}

}

}

}**Output:**

**12.Odd Even Average**

The Owner of a block visited the Layout and found that he has some plot numbers of his own and some are odd numbers and some are even numbers. He is maintaining the details in a file in the system. For the password protection our owner has followed one formula. He calculated the sum of his even numbers plot and sum of odd numbers plot and found the average of those two and he used that average as his password for the details file. Find the password that our owner has arrived.

Include a function named **avgOddEvenSum** that accepts 2 arguments and returns a float. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The function returns a float that corresponds to the average of the array.

If the size of the array is negative or if any element in the array is negative , print “Invalid Input” and terminate the program.

**Input and Output Format:**

Input consists of n+1 integers. The first integer corresponds to n, the number of elements in the array. The next 'n' integers correspond to the elements in the array.

Output consists of a floating point number that corresponds to the average. It is displayed correct to 2 decimal places.

Assume that the maximum size of the array is 20.

**Sample Input 1:**

5

1

2

3

4

5

**Sample Output 1:**

7.50

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Implementation:**

import java.util.Scanner;

public class OddEvenAverage {

public static void main(String[] args) {

int n, i,sumo=0,sume=0;

double sum,avg;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0) {

System.out.print("Invalid array size");

System.exit(0);

}

else {

int a[]=new int[n];

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

a[i] = in.nextInt();

if(a[i] < 0) {

System.out.print("Invalid input");

System.exit(0);

}

}

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

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

sume=sume+a[i];

else

sumo=sumo+a[i];

}

sum=sume+sumo;

avg = sum/2;

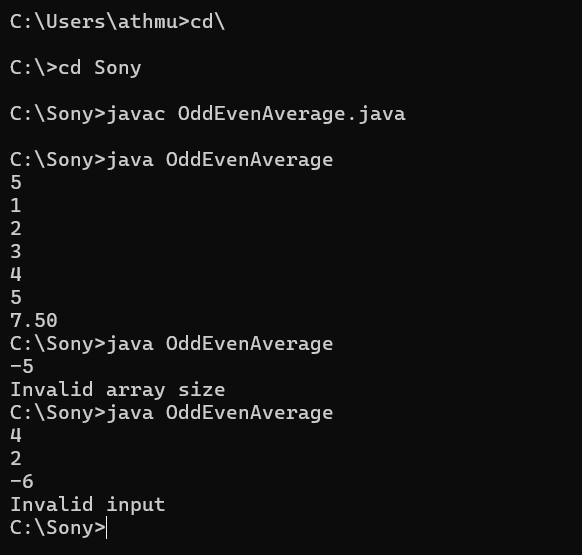
System.out.printf("%.2f",avg);

}

}

}

**ScreenShot:**

****

**14.Arithmetic Operation**

Write a program to perform a specific arithmetic operation

Include a function named **performArithmeticOperation** that accepts 3 integer arguments and returns an integer that corresponds to the result. The first and second arguments correspond to the input numbers and the third argument corresponds to the choice of arithmetic operation.

If argument 3 =1, calculate the sum of input1 and input2

If argument 3 =2, calculate the difference of input1 and input2

If argument 3 =3, calculate the product of input1 and input2

If argument 3 =4, calculate the quotient of input1 divided by input 2

If the first two argument's values is negative or greater than 32767 , the function returns -1.

If the third argument's value is not in the range 1 to 4, the function returns -1.

If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of 3 integers.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

4

12

3

**Sample Output 1:**

48

**Sample Input 2:**

-67

2

1

**Sample Output 2:**

Invalid Input

**Implementation:**

import java.util.Scanner;

public class ArithemeticOperations {

public static void main(String[] args) {

int a,b,choice,result=0;

Scanner in=new Scanner(System.in);

a = in.nextInt();

b = in.nextInt();

choice = in.nextInt();

if(a<0 || b<0 || a>32767 || b>32767)

System.out.print("Invalid input");

else if((choice<1)||(choice>4))

System.out.print("Invalid input");

else {

switch(choice) {

case 1:

result = a+b;

break;

case 2:

result = a-b;

break;

case 3:

result = a\*b;

break;

case 4:

result = a/b;

break;

}

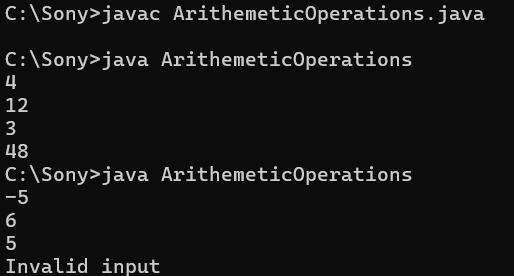
System.out.print(result);

}

}

}

**Output:**



**16.searchKeys**

Read the question carefully and follow the input and output format.  
  
Given an integer array, first index represents the key & second index represents the value. Find keys for the given value.  
  
**Input and Output Format:**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements. The next line consistts of an integer that represents the value to be searched.   
Output consist of an integer array.  
  
1) Print "Invalid array size" when size of the array is negative and terminate the program.  
2) Print "Invalid input" when there is any negative numbers available in the input array and terminate the program.  
3) Print "Key not found" when there is no keys found.  
  
Include a function named searchKeys(int array[], int size) whose return type is void.  
The output array is stored in a global variable named found.  
  
**Sample Input 1:**  
8  
1  
4  
2  
4  
3  
4  
5  
6  
4  
  
**Sample Output 1:**  
1  
2  
3  
  
**Sample Input 2:**  
5  
5  
6  
7  
8  
9  
-5  
  
**Sample Output 2:**  
Key not found

**Implementation:**

import java.util.ArrayList;

import java.util.Scanner;

public class KeyValueSearch {

private static ArrayList<Integer> found = new ArrayList<>();

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

if (n < 0) {

System.out.println("Invalid array size");

return;

}

int[] array = new int[n];

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

array[i] = scanner.nextInt();

if (array[i] < 0) {

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

return;

}

}

int value = scanner.nextInt();

if (value < 0) {

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

return;

}

searchKeys(array, n, value);

if (found.isEmpty()) {

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

} else {

for (int key : found) {

System.out.println(key);

}

}

}

public static void searchKeys(int array[], int size, int value) {

for (int i = 0; i < size; i += 2) {

if (i + 1 < size && array[i + 1] == value) {

found.add(array[i]);

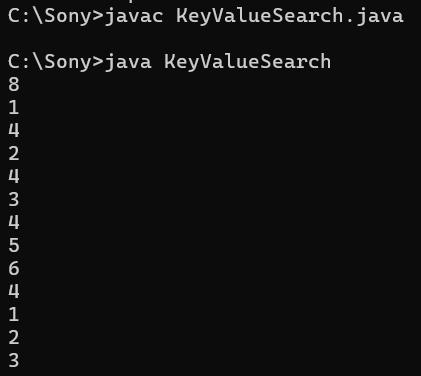
}

}

}

}

**Output:**



**17.passCount**

Read the question carefully and follow the input and output format.  
  
Given a input array, First index Represents RollNo second index represents Mark and so on.  Write a program to find the number of students who had cleared the exam.  
  
Note : If marks >=70 then He /she Cleared the exam. Array size is always even.  
  
**Input and Output Format :**  
  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements. Output consist of an integer,  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program.  
2) Print "Invalid input" when there is any negative number available in the input array and terminate the program.  
  
Include a function named passCount(int array[], int size) whose return type is an integer , the count.  
  
**Sample Input 1:**  
8  
1  
70  
2  
55  
3  
75  
4  
80  
  
**Sample Output 1:**  
3  
  
**Sample Input 2:**  
5  
60  
22  
88  
-29  
**Sample Output 2:**  
Invalid input

**Sample Input 2:**  
-5  
**Sample Output 2:**  
Invalid input

**Implementation:**

import java.util.Scanner;

public class Passcount {

public static int passCount(int[] array, int size) {

int count = 0;

for (int i = 1; i < size; i += 2) {

if (array[i] >= 70) {

count++;

}

}

return count;

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

int n = scanner.nextInt();

if (n < 0) {

System.out.println("Invalid array size");

return;

}

int[] array = new int[n];

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

array[i] = scanner.nextInt();

if (array[i] < 0) {

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

return;

}

}

int result = passCount(array, n);

System.out.println(result);

}

}

**Output Screenshot:**

