**102.countEvenRepeat**

Read the question carefully and follow the input and output format.  
  
Write a program to count the number of repeated even elements in a given input array.  
  
**Input and Output Format :**  
First line of input consists of n, the number of elements. And the remaining n lines is the elements of the array.  
Output is a single integer that displays the count.  
  
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 numbers available in the input array and terminate the program.  
  
Include a function named countEvenRepeat(int array[], int size) whose return type is an integer, the count  
  
**Sample Input 1:**  
9  
8  
4  
5  
8  
4  
2  
1  
4  
5  
  
**Sample Output 1:**  
2  
  
**Sample Input 2:**  
10  
4  
5  
6  
-8  
  
**Sample Output 2:**  
Invalid input

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int n, i,j,k,count=0,sum=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[100];

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++)

{

count=1;

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

{

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

{

count++;

for(k=j;k<n;k++)

{

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

}

n--;

}

else

j++;

}

if(count!=1 && a [i]%2==0)

sum=sum+1;

}

System.out.print(sum);

}

}

}

**111.Repeated Element**

Write a program to find the maximum repeated element in a given input array.

Include a function named **maxRepeatedElement** 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 maximum repeated element.

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 maximum repeated element.

Assume that the maximum number of elements in the array is 20 and that there will always be a unique maximum repeated element.

**Sample Input 1:**

8

2

1

3

4

6

8

10

8

**Sample Output 1:**

8

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int n, i,j,k=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[100];

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++)

{

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

{

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

{

k=a[i];

break;

}

}

System.out.print(k);

}

}

}

}

**112.subTwoArrays**

Read the question carefully and follow the input and output format.  
  
Given two input arrays,  write a program to find out numbers which is present in the first array and not in the second array.  
  
**Input and Output Format :**  
First line of input consists of n, the number of elements. Next n lines correspond to the first array elements and the next n lines correspond to the second array elements. Output consist of an integer array.  
  
1) Print "Invalid array size" when size of the array is a negative number.  
2) Print "Invalid input" when there is any negative numbers available in the input array.  
  
  
Include a function named subTwoArrays(int elements1[], int elements2[], int size) whose return type is void.  
The output array is stored in a global variable named no\_common.  
  
**Sample Input 1:**  
5  
1  
2  
3  
4  
5  
3  
5  
7  
9  
10  
  
**Sample Output 1:**  
1  
2  
4  
  
**Sample Input 2:**  
4  
1  
2  
-3  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j ,k=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)

                                                {

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

                                                            System.exit(0);

                                                }

                                    }

                                    int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

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

                                                            System.exit(0);

                                                }

                                    }

int no\_common[]=new int[100];

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

{

int flag=0;

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

{

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

flag=1;

}

if(flag==0)

{

no\_common[k]=a[i];

k++;

}

}

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

System.out.println(no\_common[i]);

}

}

}

**113.primeFactorialSum**

Read the question carefully and follow the input and output format.  
  
In a given input number , find out the sum of factorial of digits that are prime.  
  
**Input and Output Format :**  
Input consists of an integer. Output consists of the factorial sum.  
1) Print "Number too large" when the given input number is greater than 32767  
2) Print "Number too small" when the given input number is a negative number.  
  
Include a function named primeFactorialSum(int number) whose return type is an integer.  
  
**Sample Input 1:**  
123  
**Sample Output 1:**  
8  
  
Hint : 2! + 3! = (8)  
  
**Sample Input 2:**  
32768  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, sum=0,count=0,i,fact=1,rem;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

System.out.println ("Number too small");

System.exit(0);

}

if(n>32767)

{

System.out.println ("Number too large");

System.exit(0);

}

while(n!=0)

{

rem=n%10;

count=0;

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

{

if(rem%i==0)

count++;

}

if(count==2)

{

fact=1;

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

fact = fact \*i;

sum = sum + fact;

}

n=n/ 10;

}

System.out.println (sum);

}

}

**60.adjecentDifference**

Read the question carefully and follow the input and output format.  
  
Given an input Integer array, find the difference in the adjacent elements and print the largest difference  
  
**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 largest adjacent difference.  
  
Print "Invalid array size" when size of the array is a negative number and terminate the program  
Print "Invalid input" when there is any negative number available in the input array and terminate the program.  
  
Include a function named adjecentDifference(int numbers[], int size) whose return type is an integer  
  
**Sample Input 1:**  
7  
2  
4  
5  
1  
9  
3  
8  
**Sample Output 1:**  
8  
  
Hint: The AdjecentElement Diff are :2,1,4,8,6,5 and Maximum diff is 8 which is obtained by 2-4 =2, 4-5=1,5- 1 =4, 1-9=8,9-3=6,3-8 =5  
  
**Sample Input 2:**  
5  
1  
7  
3  
-8  
**Sample Output 2:**  
Invalid input

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int n, i,diff,max=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[100];

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-1;i++)

{

if(a[i]>a[i+1])

diff=a[i]-a[i+1];

else

diff=a [i+1]-a[i];

if(diff>max)

max=diff;

}

System.out.print(max);

}

}

}

**82.clearedStage1**

Read the question carefully and follow the input and output format.  
  
Given an integer array. The first index represents the Student id, Second index represents C-programming marks and the third index Represents SQL marks. Write a program to find the Ids of students who have cleared both C-programming and SQL.  
  
Note :(1) The Pass Marks is >=70  
  
**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 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 number available in the input array and terminate the program.  
  
Include a function named clearedStage1(int array[], int size) whose return type is void.  
The output array is stored in a global variable named cleared.  
  
**Sample Input 1:**  
9  
1  
25  
75  
3  
75  
80  
2  
75  
75  
  
**Sample Output 1:**  
3  
2  
  
**Sample Input 2:**  
6  
4  
25  
-78  
  
**Sample Output 2:**  
Invalid input

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int n, i,k=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[100];

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

{

a[i] = in.nextInt();

if(a[i] < 0)

{

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

System.exit(0);

}

}

int b[]=new int[100];

for(i=0;i<n;i=i+3)

{

if(a[i+1]>=70 && a[i+2]>=70)

{

b[k]=a[i];

k++;

}

}

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

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

}

}

}