1.TWO NUMBERS

Ram and Mohan are two brothers. They are not good in maths. So their father decided to give some assignment as a game so that they can enjoy as well as they can learn the concept of number systems. So he gave two numbers to them. One is the sum of two numbers, x and y, and another is the product of same two numbers. Help them to write a code to find x and y.

Note:

The two numbers should be printed in ascending order.

Input format:

The input contains two integers in the same line separated by space, that denotes the sum of x and y and the product of x and y, respectively.

Output format:

The output consists of two numbers separated by space which corresponds to x and y in ascending order.

Sample Input 1:

5 6

Sample Output 1:

2 3

Sample Input 2:

15 50

Sample Output 2:

5 10

ANSWER:

import java.util.Scanner;

public class TwoNum {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n1=input.nextInt();

int n2=input.nextInt();

int x,y,ans=0,i=0;

while(ans!=n2) {

i++;

ans=(n1\*i)-(i\*i);

}

y=i;

x=n1-y;

int f=x>y?y:x;

int s=x<y?y:x;

System.out.println(f+" "+s);

}

}

2.PICNIC EVENT

As part of the picnic events, the families from ABC apartment plan to organise some games for their kids so that they can learn some thing during leisure time also. The task is as follows.

Given a string S, find the string which has only the distinct character in S. The characters should be presented in the descending order of its occurrence. If the occurrence count of two characters is same, then the order for those two characters should be the same as it is in the input string.

Input Format:

The input is a string that denotes the S.

Note: The String is case-sensitive.

Output Format:

The output is a string that denotes the modified string.

Sample Input 1:

HelloWorld

Sample Output 1:

loHeWrd

Sample Input 2:

Entertainment

Sample Output 2:

nteEraim

ANSWER:

import java.util.Iterator;

import java.util.LinkedHashMap;

import java.util.Map;

import java.util.Map.Entry;

import java.util.Scanner;

import java.util.regex.Matcher;

import java.util.regex.Pattern;

public class Main {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

String str=input.nextLine();

String newstr="";

Map<Character,Integer> map=new LinkedHashMap<Character,Integer>();

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

int count=0;

Pattern pattern=Pattern.compile(String.valueOf(str.charAt(i)));

Matcher matcher=pattern.matcher(str);

while(matcher.find()) {

count++;

}

map.put(str.charAt(i), count);

}

int max=0;

for(Entry<Character, Integer> temp:map.entrySet()) {

if(temp.getValue()>max) {

max=temp.getValue();

}

}

for(int i=max;i>0;i--) {

for(Entry<Character, Integer> temp:map.entrySet()) {

if(temp.getValue()==i) {

System.out.print(temp.getKey());

}

}

}

} }

3.SUMMER VACATION

In summer vacation time, the students of ABC school went for a team outing. One day the instructor decided to find one funny game so that student can enjoy. So he told them to take one integer N and arrange the digits of the integer in incremental order from right to left. Help students by writing a code.

Input format:

The first line contains a single integer that denotes the given number, N.

Output format:

The output is an integer in which the digits of N are arranged in incremental order from right to left.

Sample input 1:

231

Sample output 1:

321

ANSWER:

import java.util.Arrays;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=Integer.parseInt(input.nextLine());

String str=String.valueOf(N);

char[] num=str.toCharArray();

Arrays.sort(num);

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

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

}

}

}

4.TEAM ASSESMENTS

The Team Leader of Testing team of XYZ company decided to assess the team members by organising a programming event. So he gave the team a task.

The task is given an array of size N, find the number of distinct elements in the array and print them in ascending order. Those members who answer, will get some reward.

Write a program to help the Team Leader to evaluate the solutions of the team members.

Input format:

The first line contains a single integer that denotes the size of the array, N.

The second line contains N space separated integer values of the array

Output format:

The first line is an integer that denotes the number of the distinct elements inthe array.

The second line is a series of integers seperated by space that denotes the distinct elements.

Sample Input 1:

5

1 2 1 2 1

Sample Output 1:

2

1 2

Sample Input 2:

5

145 25 21 25 36

Sample Output 2:

4

21 25 36 145

ANSWER:

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class Main {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int[] arr=new int[N];

Set<Integer> set=new TreeSet<Integer>();

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

arr[i]=input.nextInt();

set.add(arr[i]);

}

System.out.println(set.size());

for (Integer integer : set) {

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

}

}

}

5.ANUSHKA’S PROBLEM

5.Anuska has created a robot that will take instructions and work accordingly. Due to budget problem, she has set the memory low. To overcome that, she has to encode the instruction. Given the instruction S, count the consecutive characters and each character will be folowed by its frequency.

For example, if the instruction is "aaaabb", the encoded instruction is "a4b2".

Note:If the frequency is one, the count need not be printed.

Input format:

The first input is a string that denotes the instruction, S.

Output format

Output is a string that denotes the encoded string.

Sample input 1:

aaaabb

Sample output 1:

a4b2

Sample input 2:

sggvvvgss

Sample output 2:

sg2v3gs2

ANSWER:

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

String str="";

str=input.nextLine();

StringBuffer sb=new StringBuffer();

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

if(i==str.length()-1) {

if(str.charAt(i)!=str.charAt(i-1))

{

sb.append(str.charAt(i));

break;

}

}

int count=1;

while(str.charAt(i)==str.charAt(i+1)) {

count++;

i++;

if(i==str.length()-1) {

break;

}

}

if(count==1) {

sb.append(String.valueOf(str.charAt(i)));

i++;

}

else {

sb.append(String.valueOf(str.charAt(i))).

append(String.valueOf(count));

i++;

}

}

System.out.println(sb.toString());

}

}

6.SKILL TESTING

Amar's team is assigned with a project that deals with the digital signals. He is aware that the digital signals will be represented as arrays. He wants to test his team with respect to their strength in arrays.As part of the skill testing, the task is as follows.

You are given an array of integers of size N.

Consider all its contiguous subarrays of length k and find the maximum sum.

Do this for all k from 1 to the length of the input array.

Write a program to evaluate the task done by the kids.

Input format:

The first line contains a single integer denotes the size of the array, N.

The second line contains N-space separated integer that corresponds to the values of the array.

Output format:

The output consists of N-space separated integer values of the result array.

Sample Input 1:

5

-1 2 1 3 -2

Sample Output 1:

3 4 6 5 3

Explanation:

For inputArray = [-1, 2, 1, 3, -2], the output should be [3, 2, 1, 0, 0].

The contiguous subarray

of K = 1, each subarray will have 1 element. The sub-array with maximum sum is [3]. So result[0] = 3.

of K = 2, each subarray will have 2 elements. The sub-array with maximum sum is [1,3]. So result[1] = 4.

of K = 3, each subarray will have 3 elements. The sub-array with maximum sum is [2,1,3]. So result[2] = 6.

of K = 4, each subarray will have 4 elements. The sub-array with maximum sum is [-1,2,1,3]. So result[1] = 5.

of K = 5, each subarray will have 5 elements. The sub-array with maximum sum is [-1,2,1,3,-2]. So result[1] = 3.

So the result array is [3, 4, 6, 5, 3].

Sample Input 2:

5

2 3 2 -2 3

Sample Output 2:

3 5 7 6 8

ANSWER:

import java.util.Arrays;

import java.util.Scanner;

public class SkillTesting {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int a[]=new int[n];

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

a[i]=input.nextInt();

}

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

int max=0;

for(int i=0;i<(n-k)+1;i++) {

int t[]=new int[k];

t=Arrays.copyOfRange(a, i,i+k);

int sum=0;

for(int val:t) {

sum+=val;

}

if(sum>max) {

max=sum;

}

}

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

}

}

}

**1.BANGLE PROBLEM**

Vel bought 2 bangles for his elder sister Priya, as a birthday gift. Before presenting the gift, Vel challenges his sister to answer his question blindfolded. He places both the bangles on the table and asks Priya to answer two questions:  
1) Are the two bangles intersecting or touching or not touching?  
2) What is the distance between the centers of the bangles?  
To answer these questions, Vel gives the centers C1(x1, y1) and C2(x2, y2) and radius R1 and R2 of the bangles.  
Now write a program to help Priya to answer her brother's questions.  
  
**Input format:**  
First line of the input consists of two integers separated by a space which corresponds to x1 and y1 values, respectively.  
Second line of the input consists of two integers separated by a space which corresponds to x2 and y2 values, respectively.  
Third line of the input consists of two integers separated by a space which corresponds to R1 and R2 values, respectively.  
  
**Output Format:**  
First line of the Output is a string ,"Intersects" or "Touches" or "Do not Touch".  
Second line of the output is a float value which corresponds to the distance between the two centers of the bangles, Rounded-off to 2 decimal points.  
  
**Sample Input 1:**  
3 4  
14 18  
5 8  
**Sample Output 1:**  
Do not Touch  
17.80  
 **Sample Input 2:**  
3 4  
4 6  
4 5  
**Sample Output 2:**  
Intersects  
2.24  
  
**Sample Input 3:**  
1 1  
3 1  
1 1  
**Sample Output 3:**  
Touches  
2.00  
  
**Hints:**  
**Distance between two points formula:**  
D = √((x2-x1)2+(y2-y1)2)

ANSWER:

import java.text.DecimalFormat;

import java.util.Scanner;

public class BanglesGift {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int x1=input.nextInt();

int y1=input.nextInt();

int x2=input.nextInt();

int y2=input.nextInt();

int r1=input.nextInt();

int r2=input.nextInt();

double distance=Math.sqrt((Math.pow((x2-x1),2))+

(Math.pow((y2-y1),2)));

if(distance==(r1+r2)) {

System.out.println("Touches");

}

else if(distance>(r1+r2)) {

System.out.println("Do not Touch");

}

else {

System.out.println("Intersects");

}

DecimalFormat df=new DecimalFormat("#.00");

System.out.println(df.format(distance));

}

}

**2.Sliding Number**

Priya's son Sai is doing his 2nd grade. He started learning about number. To test his intelligence, Priya asks him to solve a problem. The problem is to find the number of sliding numbers within a given range. A number is called sliding number if all adjacent digits have an absolute difference of 1.   
For e.g.,  '432' is a Sliding Number while 542 is not.  
Write a program to help Sai to find the number of sliding numbers in range [M, N]. (both M and N inclusive)  
  
**Input format:**  
First input is an integer that denotes the M value.  
Second input is an integer that denotes the N value.  
**Assume**: N>M>=10.  
  
**Output format:**  
Output is an integer that denotes the number of sliding numbers.  
  
**Sample Input 1:**  
10  
25  
**Sample Output 1:**  
4  
  
**Explanation:**  
Sliding numbers are 10 12 21 23.  
  
**Sample Input 2:**  
10  
15  
**Sample Output 2:**  
2

ANSWER:

import java.util.Scanner;

public class SlidingNumber {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int M=input.nextInt();

int N=input.nextInt();

int count=0;

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

boolean isValid=true;

String str=String.valueOf(i);

for(int j=0;j<str.length()-1;j++) {

int a=Integer.parseInt(String.valueOf(str.charAt(j)));

int b=Integer.parseInt(String.valueOf(str.charAt(j+1)));

if(Math.abs(a-b)!=1) {

isValid=false;

break;

}

}

if(isValid) {

count++;

}

}

System.out.println(count);

}

}

System.out.format(“format String”,arg obj);

System.out.format(“%.2f”,dist);

**3.CHOCOLATE BOX**

Priya returned to India from the onsite trip to Singapore. She bought lots of chocolates for his son Sai and younger brother Vel. The chocolate box is square shaped and contains identical compartments. Each compartment contains several number of chocolates. Now comes the fight between Sai and Vel for the chocolates. To stop the fight, Priya gives a condition to both of them that, the compartments in the upper triangular part of the box are for Sai and that of the lower triangular part is for Vel, and the compartments in the main diagonal will be shared by Priya and her husband.  
Now help Sai and Vel to find how many chocolates did each get, by witing a program.  
  
**Note**: In this problem, Upper Triangle consists of elements above the main diagonal, and the Lower triangle consists of elements below the main diagonal.  
  
**Input Format:**  
First input is an integer that denotes the N value, size of the box. The range of N is [1 - 15], both inclusive.  
Next N lines of the input consists of N integers separated by space in each line, that denotes the number of chocolates in each compartment.  
  
**Output Format:**  
Output is two integers separated by a space that denotes the number of chocolates for Sai and number of chocolates for Vel, respectively.  
**Note:**If N = 1, there is no upper triangle and lower triangle. The single element present corresponds to the main diagonal.  
  
**Sample Input 1:**  
4  
1 2 3 4  
8 5 9 6  
1 2 3 15  
 5 7 1 6  
**Sample Output 1:**  
39 24

ANSWER:

import java.util.Scanner;

public class ChocolateBox {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int a[][]=new int[n][n];

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

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

a[i][j]=input.nextInt();

}

}

int sum1=0,sum2=0;

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

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

if(i>j&&i!=j) {

sum1+=a[i][j];

}

}

}

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

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

if(i<j&&i!=j) {

sum2+=a[i][j];

}

}

}

System.out.println(sum2+" "+sum1);

}

}

**4.SAI'S PROGRAMMING SKILLS**

Sai is doing his 5th grade in VIBGYOR International School. He has already started to do programming. Priya, being a programming geeks, tests her son's knowledge by asking him to code for her requirement. The requirement is as follows:  
Given two arrays A and B each containing N numbers, choose exactly one number from A and exactly one number from B such that the index of the two chosen numbers is not same and the sum of the 2 chosen values is minimum. Formally, if you choose ith element from A whose value is x and jth element from B whose value is y, you need to minimize the value of (x+y) such that i is not equal to j.  
Help Sai in writing the program.  
  
**Input format:**  
First line of the input is an integer that denotes the N value,size of the array.  
Second line of the input is a series of n integers separated by a space that denotes the array values of A.  
Third line of the input is a series of n integers separated by a space that denotes the array values of B.  
**Assume**: All the elements in each array are distinct.  
  
**Output format:**  
Output is an integer that denotes the minimum value as per the requirement.  
  
**Sample Input 1:**  
5  
1 2 3 4 5  
5 4 3 2 1  
**Sample Output 1:**  
2  
**Explanation:**  
Here x is 1 (the first element of A) and y is also 1 (the last element of B). Their sum is **2**.  
  
**Sample Input 2:**  
6   
2 5 4 8 9 6  
4 5 6 9 8 10  
**Sample Output 2:**  
7  
**Explanation:**  
In this sample, 2 and 4 forms the minimum sum. Since both are having the same indices, the next solution is found.  
Here x is 2 (the first element of A) and y is 5 (the second element of B). Their sum is **7**.

ANSWER:

import java.util.Arrays;

import java.util.Scanner;

public class SaisProgrammingSkills {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int[] A=new int[N];

int[] B=new int[N];

int[] A1=new int[N];

int[] B1=new int[N];

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

A[i]=input.nextInt();

A1[i]=A[i];

}

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

B[i]=input.nextInt();

B1[i]=B[i];

}

Arrays.sort(A1);

Arrays.sort(B1);

int min=A1[A1.length-1]+B1[B1.length-1];

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

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

int sum=(A[i]+B[j]);

if(min>sum&&i!=j) {

min=sum;

}

}

}

System.out.println(min);

}

}

**5.REORDER THE STUDENTS**

Sai is doing his 5th grade in VIBGYOR International School. He is the most intelligent student in his class, and his class teacher has made him the representative for his class. Now its play time. There are n students in his class and each student is given a card that has a number written in it. Sai, being the representative, has to reorder those kids as per the order given to him.  
Help Sai in reordering the kids, by writing a program.  
  
**Input format:**  
First input is an integer that denotes the n value, the number of students.  
Second input is a series of n integers separated by a space that denotes the numbers given to each student.  
Third input is a series of n integers separated by a space that denotes the array which contains the order in which the students have to be reordered.  
**Assume:** The elements of the 2nd array are unique and always greater than or equal to 0 and less than n.  
  
**Output format:**  
Output is a series of n integers separated by a space that denotes the reordered array.  
  
**Sample Input 1:**  
3  
14 15 16  
1 2 0  
**Sample Output 1:**  
16 14 15  
**Explanation**:  
The first array : 16 14 15  
The order array : 1 2 0  
Here the first element 16 should be moved to index 1.  
The second element 14 should be moved to index 2.  
The Third element 15 should be moved to index 0.  
So, after moving, the final order is **16 14 15**.  
  
**Sample Input 2:**  
5  
1 2 3 4 5  
1 2 0 3 4  
**Sample Output 2:**  
3 1 2 4 5

ANSWER:

import java.util.Scanner;

public class ReorderTheStudents {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int arr[]=new int[n];

int ind[]=new int[n];

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

arr[i]=input.nextInt();

}

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

ind[i]=input.nextInt();

}

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

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

if(i==ind[j]) {

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

}

}

}

}

}

**6.PRACTICAL EXAM PREPARATION**

Vel is a first year Computer Science student. The next day he is having his practicals lab exam. His sister(Priya) is working in a top IT firm. To test Vel, she gives him a problem to solve. The problem is to find the N-th Special Number, given the value of N. She also defines a number to be special, if it’s units place digit only consist of 0, 1, 2, 3, 4 or 5.  
Write a program to help Vel.  
  
**Input format:**  
First input is an integer that denotes the N value.  
  
**Output Format:**  
Output is an integer that denotes the N-th special number.  
  
**Sample Input 1:**  
5  
**Sample Output 1:**  
4  
**Explanation:**  
Here N = 5.  
The first 5 special numbers are 0, 1, 2, 3 and 4. So the output is **4**.  
  
**Sample Input 2:**  
60  
**Sample Output 2:**  
95  
**Explanation:**  
Here N = 60.  
The first 60 special numbers are as follows:  
0, 1, 2, 3, 4, 5,  
10, 11, 12, 13, 14, 15,  
20, 21, 22, 23, 24, 25,  
30, 31, 32, 33, 34, 35,  
40, 41, 42, 43, 44, 45,  
50, 51, 52, 53, 54, 55,  
60, 61, 62, 63, 64, 65,  
70, 71, 72, 73, 74, 75,  
80, 81, 82, 83, 84, 85,  
90, 91, 92, 93, 94 and 95.  
So the output is **95**.

ANSWER:

**import** java.util.Scanner;

**public** **class** PracticalExamPreparation {

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

Scanner input=**new** Scanner(System.***in***);

**int** n=input.nextInt();

**int** i=0,count=0;

String temp="012345";

**while**(count!=n) {

String s=String.*valueOf*(i);

**if**(temp.indexOf(s.charAt(s.length()-1))>=0) {

count++;

}

i++;

}

System.***out***.println(i-1);

}

}

**1.BOAT TRAVEL**

You're travelling in a boat, the water currents sometimes favours you and sometimes they slow you down. You need to calculate the time to reach the destination knowing the boat speed and the water current speed.   
A boat moves at a constant speed of x kmph without water flow. The water flow also affects the speed boat.  
Given the ideal boat speed and the water flow speed, write a program to find the time required to reach a given distance.  
  
**Input format:**  
First Line of the input contains a single integer value that denotes the ideal boat speed in kmph.  
Second Line of the input contains a single integer value that denotes the water flow speed in kmph.(+ve when it favours the boat and -ve when it disfavours the boat)  
Third Line of the input contains a single integer value that denotes the distance to be covered in km.  
  
**Output format:**  
A single Integer value that denotes the time taken to reach the given distance.  
  
**Assume**: The time taken will always be an integer.  
  
**Sample input 1:**  
20  
5  
100  
**Sample Output** 1:  
4  
  
**Sample input 2:**  
25  
-5  
100  
**Sample output 2:**  
5

**ANSWER:**

import java.util.Scanner;

public class BoatTravel {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int idealspeed=input.nextInt();

int waterflowspeed=input.nextInt();

int distance=input.nextInt();

int timetaken=(distance/(idealspeed+waterflowspeed));

System.out.println(timetaken);

}

}

**2.BIGGER NUMBER**

Suriya loves playing with numbers. He comes up with  a new game idea of creating a greater number than the given number by changing exactly one digit. Now he has to find the number of ways in which this is possible. Help Suriya to develop a program to find the number of ways to obtain a bigger number.  
  
**Input format:**  
Input contains a single integer that denotes the given number.  
  
**Output format:**  
The output contains a single integer that denotes the number of ways.  
  
**Sample Input 1:**  
10  
**Sample Output 1:**  
17  
  
**Sample Input 2:**  
32310  
**Sample output 2:**  
36   
  
**Explanation:**  
First digit 3 can be changed into 4,5,6,7,8,9 so **6** possibilities  
Second digit 2 can be changed into 3,4,5,6,7,8,9 so **7** possibilities  
Third digit 3 can be changed into 4,5,6,7,8,9 so **6** possibilities  
Fourth digit 1 can be changed into 2,3,4,5,6,7,8,9 so **8** possibilities  
Fifth digit 0 can be changed into 1,2,3,4,5,6,7,8,9 so **9** possibilities  
**Total**  
6+7+6+8+9 = 36

**ANSWER:**

import java.util.Scanner;

public class BiggerBumberSuriya {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

String str=String.valueOf(N);

int num=0,sum=0;

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

num=Character.

getNumericValue

(str.charAt(i));

int count=0;

for(int j=num+1;j<=9;j++) {

count++;

}

sum+=count;

}

System.out.println(sum);

}

}

**3.GREATEST COMMON PRIME DIVISOR**

Sapna has learnt about prime numbers and Greatest common divisor. Now she experiments by finding the greatest common prime divisor.  
Help her develop a program by combining her knowledge in both the concepts to find the greatest common prime divisor(gcpd).  
Given two numbers, write a program to find the greatest common prime divisor.  
  
**Note :**  
    If there is no such numbers, print -1.  
  
**Input Format:**  
First Line of the input contains a single integer that denotes the first number.  
Second Line of the input contains a single integer that denotes the second number.  
  
**Output Format:**   
The output contains a single integer that denotes the gcpd of the two inputs  
  
**Sample Input 1:**  
12  
18  
**Sample Output 1:**  
3  
  
**Sample Input 2:**  
12  
13  
**Sample Output 2:**  
-1

**ANSWER:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class GreatestCommonPrimeDivisor {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int m=input.nextInt();

int gcd=0;

ArrayList<Integer> gcds=new ArrayList<Integer>();

ArrayList<Integer> primes=new ArrayList<Integer>();

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

if(n%i==0&&m%i==0) {

gcd=i;

gcds.add(gcd);

}

}

for(Integer temp:gcds) {

boolean isPrime=true;

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

if(temp%i==0) {

isPrime=false;

}

}

if(isPrime) {

primes.add(temp);

}

}

Collections.sort(primes);

if(primes.isEmpty()) {

System.out.println(-1);

}

else {

System.out.println

(primes.get(primes.size()-1));

}

}

}

**4.POWER PAIRS**

Your friend challenges you to play a game, in which you have to find the number of Power pairs in a series of numbers. A Power pair is formed when a number-b is divisible by another number-a that occurs before the number-b in the given sequence. You can easily win the challenge by developing a program to find it.  
  
**Input Format**  
First Line of the input contains a single integer that denotes the size of the array - n.  
Second line consists of n space separated integers that denote the array values.  
  
**Output Format**  
The output consists a single integer that denotes the number of Power pairs in the given sequence.  
  
**Sample Input 1:**  
3  
1 3 2  
**Sample Output 1:**  
2  
  
**Explanation :**  
For sequence = [1, 3, 2]  
The sorted pairs are: (1, 3), (1, 2).      
So output should be 2.  
  
**Sample Input 2:**  
3  
2 4 8  
**Sample Output 2:**  
3

**ANSWER:**

import java.util.ArrayList;

import java.util.Collections;

import java.util.List;

import java.util.Scanner;

import java.util.Set;

import java.util.TreeSet;

public class PowerPairs {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=input.nextInt();

int a[]=new int[n];

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

a[i]=input.nextInt();

}

int count=0;

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

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

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

count++;

}

}

System.out.println(count);

}

}

**5.DIAGONAL DIFFERENCE**

Roshan just found out that matrix can be implemented using 2d arrays. He is good in programming. So he tries to develop  a program to do his math homework. Help Roshan to write a program to do his math homework.  
In his math home work, a square matrix of size N is given. He needs to calculate the absolute difference between the sums of its diagonals.  
Help Roshan by writing a program.  
  
**Input Format:**  
The first line contains a single integer denotes the size of the matrix, N.  
The next N  lines denote the matrix's rows, with each line containing N space-separated integers describing the columns.  
  
**Output Format:**  
Print the absolute difference between the two sums of the matrix's diagonals as a single integer.  
  
**Sample Input 1:**  
3  
11 2 4  
4 5 6  
10 8 -12  
**Sample Output 1:**  
15  
  
**Explanation:**  
The primary diagonal elements are:  
11 5 -12  
Sum across the primary diagonal: 11 + 5 - 12 = 4  
  
The secondary diagonal elements are:  
4 5 10  
Sum across the secondary diagonal: 4 + 5 + 10 = 19  
Absolute Difference = |4 - 19| = **15**.

**ANSWER:**

import java.util.ArrayList;

import java.util.Scanner;

public class DiagnalDifference {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int[][] a=new int[N][N];

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

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

a[i][j]=input.nextInt();

}

}

int i=0;

int firstsum=0,secondsum=0;

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

firstsum+=a[i][j];

i++;

}

int n=0;

for(int j=N-1;j>=0;j--) {

secondsum+=a[n][j];

n++;

}

System.out.println

(Math.abs(firstsum-secondsum));

}

}

**6.EXCHANGE POSITIONS IN ARRAY**

Nikitha is very interested in programming with arrays, she tries to exhange the positions of the elements by swapping the element and the number at the place which it occupies.  
For Example,  
Consider the array elements.  
1 3 4 2  
The element 1 in given array , is at position 1, so 1 is placed at position 1 in output array  
The element 3 in given array , is at position 2, so 2 is placed at position 3 in output array  
The element 4 in given array , is at position 3, so 3 is placed at position 4 in output array   
The element 2 in given array , is at position 4, so 4 is placed at position 2 in output array  
The output array becomes 1,4,2,3  
  
Develop a program to help Nikitha.  
  
**Input Format:**  
First Line of the input contains a single integer that denotes the size of the array n  
Second  line consists of n space separated integer array values.  The input array values will always be a permutation of numbers from 1 to n.  
  
**Output Format:**  
The output consists of n space separated integers that denote the output array values.  
  
**Sample Input 1:**  
4  
1 3 4 2  
**Sample Output 1:**  
1 4 2 3  
  
**Sample Input 2:**  
3  
1 2 3  
**Sample Output 2:**  
1 2 3

**ANSWER:**

import java.util.Scanner;

public class ExchangePositionsInArray {

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int N=input.nextInt();

int a[]=new int[N];

int b[]=new int[N];

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

a[i]=input.nextInt();

}

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

if((i+1)==a[i]) {

b[i]=i+1;

}

else {

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

}

}

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

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

}

}

}

**1)CompletetheSequence**

import java.util.HashSet;

import java.util.Scanner;

import java.util.Set;

public class CompletetheSequence

{

public static void main(String[] args)

{

Scanner input=new Scanner(System.in);

String str=input.nextLine();

String arr[]=str.split(",");

int d1=Math.abs(Integer.parseInt(arr[0])-Integer.parseInt(arr[1]));

Set set=new HashSet();

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

set.add(Math.abs(Integer.parseInt(arr[i])-Integer.parseInt(arr[i+1])));

}

if(set.size()==1){

System.out.println(Integer.parseInt(arr[arr.length-1])+d1);

}

else{

System.out.println(Integer.parseInt(arr[arr.length-1])\*2);

}

}

}

2) HappyCoding

import java.util.\*;

public class HappyCoding {

public static void main(String[] args)

{

//System.out.println("enter n,m1,m2 values ");

Scanner s=new Scanner(System.in);

int n=s.nextInt();

int m1=s.nextInt();

int m2=s.nextInt();

int counthappy=0;

int countcoding=0;

int counthc=0;

StringBuffer sb=new StringBuffer();

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

{

if((i%m1==0) && (i%m2==0))

{

sb.append("Happycoding");

counthc++;

}

else if(i%m2==0)

{

sb.append("coding");

countcoding++;

}

else if(i%m1==0)

{

sb.append("Happy");

counthappy++;

}

else

{

sb.append(" "+i);

}

}

System.out.println(sb+ " ");

System.out.println("Happy -"+counthappy+"\ncoding - "+countcoding+"\nHappycoding - "+counthc);

}

}

3) Modifythestring

import java.util.Scanner;

public class Modifythestring

{

public static void main(String[] args) {

Scanner input=new Scanner(System.in);

int n=Integer.parseInt(input.nextLine());

int arr[]=new int[n];

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

arr[i]=input.nextInt();

}

input.nextLine();

String s=input.nextLine();

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

char c=s.charAt(i);

for(int j=0;j<arr[i];j++){

c=++c;

if(c==123){

c='a';

}

System.out.println(c);

}

}

}

}

4) MysteriousNumber

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

/\*\*

\*

\*/

/\*\*

\* @author sirin

\*

\*/

public class MysteriousNumber {

/\*\*

\* @param args

\* @throws IOException

\* @throws NumberFormatException

\*/

public static void main(String[] args) throws NumberFormatException, IOException

{

//System.out.println("enter the number");

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int k=Integer.parseInt(br.readLine());

int count=0;

int i=3;

while(true)

{

//check for positive number

if(i<=0)

{

i++;

continue;

}

//check for prime number

int is\_prime=1;

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

{

int temp=i%j;

if(temp==0)

{

is\_prime=0;

break;

}

else

{

is\_prime=1;

}

}

if(is\_prime==1)

{

i++;

continue;

}

//check for fibonnacci number

int x=5\*i\*i+4;

int y=5\*i\*i-4;

int s=(int) Math.sqrt(x);

boolean r=s\*s==x;

int p=(int) Math.sqrt(y);

boolean r1=p\*p==y;

if(!(r|r1))

{

count++;

if(count==k)

{

System.out.println(i);

break;

}

}

i++;

}

}

}

5) PossiblNumbers

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class PossiblNumbers

{

public static void main(String[] args) throws IOException

{

String arrstr[]=new String[1000];

int arr[]=new int[1000];

BufferedReader br=new BufferedReader(new InputStreamReader(System.in) ); // TODO Auto-generated method stub

int n=Integer.parseInt(br.readLine().trim());

int count=Integer.parseInt(br.readLine().trim());

arrstr=br.readLine().split("\\s");

int result=0;

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

arr[i]=Integer.parseInt(arrstr[i].trim());

}

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

int number=i;

int temp[]=new int[1000];

int j=0;

while(number>0)

{

temp[j]=number%10;

number=number/10;

j++;

}int num\_digit=0;

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

int digit\_present=0;

for(int q=0;q<count;q++){

if(temp[p]==arr[q]){

digit\_present=1;

break;

}

}

if(digit\_present==1)

{

num\_digit++;

}

}

if(num\_digit==1){

result++;

}

}

System.out.println(result);

}

}

6) StrictlyIncreasingSum

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class StrictlyIncreasingSum {

public static void main(String[] args) throws NumberFormatException, IOException

{

String arrstr[]=new String[1000];

int arr[]=new int[1000];

BufferedReader br=new BufferedReader(new InputStreamReader(System.in));

int n=Integer.parseInt(br.readLine());

arrstr=br.readLine().split("\\s");

int max\_sum=0;

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

{

arr[i]=Integer.parseInt(arrstr[i]);

}

//Initialize current sum be arr[0]

int current\_sum=arr[0];

//Traverse array elements after first element

for (int i = 1; i < arr.length; i++)

{

//update current sum for strictly increasing sub array

if(arr[i-1]<arr[i])

{

current\_sum=current\_sum+arr[i];

}

else //strictly increasing sub array break

{

//update max sum and current sum

max\_sum= Math.max(max\_sum, current\_sum);

current\_sum=arr[i];

}

}

int final\_sum=Math.max(max\_sum, current\_sum);

System.out.println(final\_sum);

}

}

7) TravelSpeed

import java.text.DecimalFormat;

import java.util.Scanner;

public class TravelSpeed {

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

Double x=s.nextDouble();

Double y=s.nextDouble();

Double z=s.nextDouble();

Double sp=x\*y;

Double ans=sp/z;

DecimalFormat df=new DecimalFormat("##.00");

System.out.println(df.format(ans));

}

}

7) HighestMultiNumInArray

import java.util.ArrayList;

import java.util.Collections;

import java.util.Scanner;

public class HighestMultiNumInArray {

public static void main(String[] args)

{

int max=0;

ArrayList<Integer> a=new ArrayList<>();

Scanner s=new Scanner(System.in);

int limit=s.nextInt();

int sum=0;

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

{

a.add(s.nextInt());

}

Collections.sort(a);

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

{

for(int k=j+1;k<limit;k++)

{

if(a.contains(a.get(j)\*a.get(k)))

{

max=a.get(j)\*a.get(k);

}

}

}

System.out.println(max);

}

}

8) ArrangeTheArray

import java.util.ArrayList;

import java.util.Scanner;

public class ArrangeTheArray {

public static void main(String[] args)

{

Scanner s=new Scanner(System.in);

ArrayList<Integer> pos=new ArrayList<Integer>();

ArrayList<Integer> neg=new ArrayList<Integer>();

int l=s.nextInt();

Integer a[]=new Integer[l];

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

{

int temp=s.nextInt();

if(temp>=0)

{

pos.add(temp);

}

else

{

neg.add(temp);

}

}

boolean rev=true;

if(rev)

{

neg.addAll(pos);

a=neg.toArray(a);

}

else

{

pos.addAll(neg);

a=pos.toArray(a);

}

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

{

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

}

}

}

8) TheDifference

import java.util.Scanner;

public class TheDifference {

public static void main(String[] args) {

Scanner s=new Scanner(System.in);

int r=s.nextInt();

int c=s.nextInt();

int[][] a=new int[r][c];

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

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

a[i][j]=s.nextInt();

}

}

int msr=0;

int msc=0;

int minsr=0;

int minsc=0;

int iminr=0;

int iminc=0;

int imr=0;

int imc=0;

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

int sum=0;

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

sum=sum+a[i][j];

}

if(sum>msr){

msr=sum;

imr=i;

}

}

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

int sum=0;

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

sum=sum+a[j][i];

}

if(sum>msc){

msc=sum;

imc=i;

}

}

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

int sum=0;

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

sum=sum+a[i][j];

}

if(sum<minsr){

minsr=sum;

iminr=i;

}

}

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

int sum=0;

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

sum=sum+a[i][j];

}

if(sum<minsc){

minsc=sum;

iminc=i;

}

}// TODO Auto-generated method stub

System.out.println(Math.abs(a[imr][imc]-a[iminr][iminc]));

}

}

**1.calculateSal**

Read the question carefully and follow the input and output format.  
  
Karen got salary for this month and she spends 20% of her salary for food and 30% of her salary for travel. If she takes care of other shifts she will get 2% of the salary per day. Given her salary and the number of shifts she handled. Calculate how much she can save in her pocket after spending all these?  
  
**Input and Output Format :**  
First line of input consists of an integer, salary. Next line correspond to the number of shifts. Output consist of an integer, which is saving.  
  
1) Print "Salary too large" when salary is greater than 8000.  
2) Print "Shifts too small" when the shift is less than 0.  
3) Print "Salary too small" when the salary is less than 0.  
  
Include a function named calculateSal(int salary, int shifts) whose return type is an integer, which is the saving.  
  
**Sample Input 1:**  
7000  
5  
**Sample Output 1:**  
4200  
  
**Sample Input 2:**  
8001  
  
**Sample Output 2:**  
Salary too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int salary=0,shifts=0;

                        double savings=0;

                        Scanner in=new Scanner(System.in);

                        salary = in.nextInt();

                        shifts = in.nextInt();

                        if(salary > 8000)

                        System.out.print("Salary too large ");

                        else if(shifts<0)

                        System.out.print ("Shifts too small\n");

                        else if(salary<0)

                        System.out.print ("Salary too small");

                        else

                        {

                                    savings = (salary\*0.5)+(salary\*0.02\*shifts);

                                    System.out.printf ("%.0f",savings);

                        }

            }

}

**2.Repeated Salary Count**

 John is working as a clerk in an organization where N number of people are working. His boss has asked him to get the count of employees who get same salary. Help him to get the count of repeated salary.

 Include a function named **countRepeaters** 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 number of repeaters.

 If the size of the array is negative or if any of the array elements are 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 number of repeaters.

 Assume that utmost one element in the array would repeat.

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

**Sample Input 1:**

5

1000

2000

3500

2000

5000

**Sample Output 1:**

2

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

1000

-2000

**Sample Output 3:**

Invalid Input

import java.util.\*;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,k,count=1;

                        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;)

                                                {

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

                                                            {

                                                                        count++;

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

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

                                                                        n--;

                                                            }

                                                            else

                                                                        j++;

                                                }

                                    }

                                    System.out.print(count);

                        }

            }

}

**3.maximumSum**

Read the question carefully and follow the input and output format.  
  
Given an Integer array, find out sum of Even and odd Numbers individually and find the maximum.  
  
**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 maximum of odd and even sum.  
  
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 maximumSum(int numbers[], int size) whose return type is an integer,.  
  
**Sample Input 1:**  
5  
12  
13  
14  
15  
16  
  
**Sample Output 1:**  
42  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid array size

**4.Product of Digits**

 In a car racing video game, the car is an object. You can drive the car, turn the car, or stop the car when needed but you need to drive long. You will get money according to the Km you have travelled. For example if you have travelled 123 km then the product of the km (ie 1\*2\*3 = 6) would be the amount you win. Write a program to find the product of the digits in the given input number.

 Include a function named **productDigits** that accepts an integer argument and returns an integer that corresponds to the product of digits in the integer.

The function returns -1 if the input number is negative or greater than 32767.

 If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of an integer.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

32

**Sample Output 1:**

6

**Sample Input 2:**

-67

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rem,prod=1;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0 || n>32767)

                        {

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

                                    System.exit(0);

                        }

                        else

                        {

                                    while(n!=0)

                                    {

                                                rem=n%10;

                                                prod=prod\*rem;

                                                n=n/10;

                                    }

                                    System.out.println(prod);

                        }

            }

}

**5.findCricketerId**

Read the question carefully and follow the input and output format.  
  
Given an input array first Index indicates the cricketer’s id and second index indicates the score and so on……Find out the cricketer's id who scored more than given score  
  
**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 of the input consists of an integer that corresponds to the given score. Output consist of an integer array, which contains cricketer's id who have scored more than the given score.  
  
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 "Invalid score" when the score is negative.  
  
Include a function named findCricketerId(int array[], int size, int score) whose return type is void.  
The output array is stored in a global variable named cricketer.  
  
**Sample Input 1:**  
6  
1  
1000  
5  
2000  
3  
4000  
1000  
  
**Sample Output 1:**  
5  
3  
  
**Sample Input 2:**  
6  
1  
1000  
5  
3000  
3  
4000  
-1000  
  
**Sample Output 2:**  
Invalid score

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j=0,score;

                        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);

                                                }

                                    }

                                    score = in.nextInt();

                                    if(score<0)

                                    {

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

                                                System.exit(0);

                                    }

                                    int cricketer[]=new int[100];

                                    for(i=1;i<n;i=i+2)

                                    {

                                                if(a [i]>score)

                                                {

                                                            cricketer[j]=a [i-1];

                                                            j++;

                                                }

                                    }

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

                                    {

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

                                    }

                        }

            }

}

**6.Fahrenhiet to Centigrade**

 Write a program to convert given temperature from Fahrenheit to Centigrade.

**Formula:**

C/5 = (F-32)/9

C stands for Centigrade.

F stands for Fahrenheit.

 Include a function named **convertToCentigrade** that accepts an integer argument and returns a float that corresponds to the centigrade equivalent.

 If the input is a negative number, print Invalid Input and terminate the program.

**Input and Output Format:**

Input consists of a single integer.

Output consists of a floating point number that corresponds to the centigrade equivalent. The output is displayed correct to 2 decimal places.

**Sample Input 1:**

77

**Sample Output 1:**

25.00

**Sample Input 2:**

**-**2345

**Sample Output 2:**

Invalid Input

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        double f,c;

                        Scanner in=new Scanner(System.in);

                        f = in.nextDouble();

                        if(f < 0)

                        {

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

                                    System.exit(0);

                        }

                        else

                        {

                                    c= (f-32)/9\*5;

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

                        }

            }

}

**7.highestFeedBack**

Read the question carefully and follow the input and output format.  
  
In a company there are some managers working on two different projects (MetLife and Hardfort). When the feedback was taken their feedback was present in both MetLife Feedback as well as Hardfort Feedback. Write a method to create a consolidated feedback for the managers for MetLife and HardForts. For those working on both the projects the highest feedback is taken. In the 2 given arrays, the First Index represents the Employee id and second one Represents The Feed Back Score and so on....  
  
**Input and Output Format:**  
First line corresponds to n, the size of the array. The next n lines correspond to the elements of the first array. The next n lines correspond to the elements in the second array. Output corresponds to the consolidated feedback score.  
  
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 highestFeedBack(int metlife[],int hardfort[],int size) whose return type is void.  
The output array is stored in a global variable named fedback.  
  
**Sample Input 1:**  
8  
1  
90  
2  
75  
3  
92  
5  
85  
1  
80  
2  
85  
3  
80  
4  
85  
**Sample Output 1:**  
1  
90  
2  
85  
3  
92  
5  
85  
4  
85  
  
**Sample Input 2:**  
5  
5  
8  
9  
1  
-6  
**Sample Output 2:**  
Invalid number  
  
**Sample Input 3:**  
-4  
**Sample Output 3:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,k=0,count,count1;

                        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 c[]=new int[100];

                                    for(i=0;i<n;i=i+2)

{

                                                count=0;

                                                for(j=0;j<n;j=j+2)

{

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

{

                                                                        count=1;

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

{

                                                                                    c[k]=a[i];

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

                                                                                    k++;

                                                                        }

                                                                        else

{

                                                                                    c[k]=a[i];

                                                                                    c[++k]=b[j+1];

                                                                                    k++;

                                                                        }

                                                            }

                                                }

                                                if(count==0)

{

                                                            c[k]=a[i];

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

                                                            k++;

                                                }

                                    }

                                    for(i=0;i<n;i=i+2)

{

                                                count1=0;

                                                for(j=0;j<n;j=j+2)

{

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

{

                                                                        count1=1;

                                                            }

                                                }

                                                if(count1!=1)

{

                                                            c[k]=b[i];

                                                            c[++k]=b[i+1];

                                                            k++;

                                                }

                                    }

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

{

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

                                    }

                        }

            }

}

**8.primeIndexSum**

Read the question carefully and follow the input and output format.  
  
Given an Integer array. Find the average of the numbers located on the Prime Indexes of the Array. Consider 0 index as 1 and 1 index is 2 and so on……  
  
Hint :Consider 1 is not a prime number  
  
**Input and Output Format :**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements . Output consists of an Integer, the prime index sum.  
  
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 primeIndexSum(int array[], int size) whose return type is an integer,which is the sum.  
  
**Sample Input 1:**  
7  
2  
45  
1  
9  
3  
8  
  
**Sample Output 1:**  
6  
  
**Sample Input 2:**  
-7  
  
**Sample Output 2:**  
Invalid array size

import java.util.\*;

public class Main

{

            public static void main(String[] args)

            {

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

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

                                    {

                                                a[i] = in.nextInt();

                                                 if(a[i] < 0)

                                                {

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

                                                            System.exit(0);

                                                }

                                    }

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

{

                                                count=0;

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

{

                                                            if(i%j==0)

                                                            count++;

                                                }

                                                if(count==2)

{

                                                            sum = sum+a[i];

                                                            temp++;

                                                }

                                    }

                                    avg=sum/temp;

                                    System.out.print(avg);

                        }

            }

}

**9.Element Count**

Write a program to find the number of times a particular number occurs in a given input array.

Include a function named findElementCount that accepts 3 arguments and returns an int. The first argument is the input array, the second argument is an int that corresponds to the size of the array and the third argument is the element to be searched for. The function returns an int that corresponds to the number of times the search element occurs in 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+2 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. The last integer corresponds to the element whose count needs to be found.

Output consists of an integer that corresponds to the number of times the search element occurs in the array.

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

**Sample Input 1:**

8

2

1

3

8

6

8

10

8

8

**Sample Output 1:**

3

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

#include<stdio.h>

#include<stdlib.h>

int findElementCount(int,int[],int);

int main(){

            int n=0,flag=0,i,input[20],search=0;

            int count=0;

            scanf("%d",&n);

            if(n<0){

                        printf("Invalid Input");

                        getchar();

                        getchar();

                        exit(0);

            }

            for(i=0;i<n;i++){

                        scanf("%d",&input[i]);

                        if(input[i]<0)

                                    flag=1;

            }

            if(flag==1){

                        printf("Invalid Input");

                        getchar();

                        getchar();

                        exit(0);

            }

            scanf("%d",&search);

            count = findElementCount(n,input,search);

            printf("%d",count);

            getchar();

            getchar();

            return 0;

}

int findElementCount(int n,int array[],int find){

            int count=0,i;

            for(i=0;i<n;i++){

                        if(array[i]==find)

                                    count++;

            }

            return count;

}

**10. powerOfTwo**

Read the question carefully and follow the input and output format.  
  
Check whether given number is a power of 2 or not .If yes Print 'Yes' else 'No'  
  
**Input and Output Format :**  
Input consists of an integer number. And output is a single line that displays 'Yes' or 'No'  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
  
Include a function named powerOfTwo(int n) that returns an integer.  
  
  
**Sample Input 1:**  
3  
**Sample Output 1:**  
No  
  
**Sample Input 2**:  
34569  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rem,prod=1;

                        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);

                        }

                        else

                        {

                                    if(n==0)

                                    {

                                                System.out.println(yes);

System.exit(0);

                                    }

while(n!=1)

                                    {

                                                if(n%2!=0)

                                                {

                                                            System.out.println ("no ");

                                                            System.exit(0);

                                                }

                                                n=n/2;

                                    }

                                    System.out.println("yes");

                        }

            }

}

**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

 import java.util.Scanner;

public class Main

{

            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);

                                    }

                        }

            }

}

**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

**Sample Output 3:**

Invalid Input

 import java.util.Scanner;

public class Main

{

            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);

                        }

            }

}

**13.Decimal Conversion**

Write a program to convert a given input binary number to decimal.

Include a function named **convertToDecimal** that accepts an integer argument and returns an integer that corresponds to the decimal representation of the input number. If the input value is not a binary value or if the input is negative or if the input is greater than 11111, the function returns -1.

If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer that corresponds to the binary representation of a number.

Output consists of a single integer that corresponds to the decimal equivalent of the given number.

Refer sample output for formatting specifications.

**Sample Input 1:**

1100

**Sample Output 1:**

12

**Sample Input 2:**

101010

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

1201

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int num=0,binary,rem,power;

                        double a,decimal=0;

                        Scanner sc=new Scanner(System.in);

                        num=sc.nextInt();

                        if(num<0 || num>11111)

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

                        else

                        {

                                    binary = num;

                                    power = 0;

                                    while (binary != 0)

                                    {

                                                rem = binary % 10;

                                                a= Math.pow(2, power);

                                                decimal = decimal+ rem \* a;

                                                power++;

                                                binary = binary / 10;

                                    }

                                    System.out.printf("%.0f",decimal);

                        }

            }

}

**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

import java.util.Scanner;

public class Main

{

            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);

                        }

            }

}

**15.digitFactorial**

Read the question carefully and follow the input and output format.  
  
In a given input number , find out the factorial of each individual digit and assign it to output array.  
  
**Input and Output Format:**  
Input consists of a single integer. Output consists of an Integer array, the individual factorials.  
  
Print "Number too large" when the given input numbers is greater than 32767 .  
Print "Number too small" when the given input is a negative number.  
  
Include a function named digitFactorial(int number) whose return type is void.  
The output array is stored in a global variable named factorial.  
  
**Sample Input 1:**  
123  
  
**Sample Output 1:**  
1  
2  
6

**Sample Input 2:**  
-2526  
  
**Sample Output 2:**  
Number too small

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,fact,k=0,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);

                        }

                        int factorial[]=new int[100];

while(n!=0)

{

                                  rem=n%10;

                                 fact=1;

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

                                  {

                                          fact = fact \* i;

                                  }

                                  factorial[k]=fact;

                                   k++;

                                   n=n/10;

                        }

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

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

           }

}

**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

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,k=0,se,flag=0,f=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);

                                                }

                                    }

                                    se=in.nextInt();

                                    if(se<0)

{

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

System.exit(0);

                                   }

                                    int found[]=new int[100];

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

                                    {

                                      if(a [i]==se)

{

                                                    f=1;

                                                            found[k]=a [i-1];

                                                            k++;

                                                  }

                                     }

                                      if(f==0)

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

                                       else

                                      {

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

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

                                       }

                        }

           }

}

**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  
6  
2  
8  
-2  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,count=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n < 0 || n%2!=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=1;i<n;i=i+2)

{

                                                if(a[i]>=70)

                                                count++;

                                   }

                                     System.out.print(count);

}

          }

}

**18.5 Multiples --- Average**

Write a program to find the average of multiples of 5 upto 'n'. n is given as input.

Include a function named **findAverageBy5s** that accepts an integer argument and returns a float that corresponds to the average of multiples of 5.

If the input value is negative or greater than 32767, print Invalid Input and terminate the program.

**Input and Output Format:**

Input consists of a single integer.

Output consists of a floating point number. Output is displayed correct to 2 decimal places.

Refer sample output for formatting specifications.

**Sample Input 1:**

10

**Sample Output 1:**

7.50

**Sample Input 2:**

-67

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,count=0;

                        double sum=0,avg;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0 || n>32767)

                        {

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

                                    System.exit(0);

                        }

                       for(i=5;i<=n;i++)

{

                                    if(i%5==0)

{

                                                sum=sum+i;

                                                count++;

                                    }

                        }

                        avg = sum/count;

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

            }

}

**19.Sum of Odd Digits**

Write a program to find the sum of the odd digits in a number.

Include a function named **sumOddDigits**that accepts an integer argument and returns an integer that corresponds to the sum of the odd digits. The function returns -1 if the input is less than zero or if it is greater than 32767.

If the function returns -1, print “Invalid Input”.

**Input and Output Format:**

The input consists of an integer.

The output consists of an integer that corresponds to the sum of the odd digits in the number.

**Sample Input 1:**

3487

**Sample Ouput 1:**

10

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,rem,sum=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0 || n>32767)

                        {

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

                                    System.exit(0);

                        }

                      else

{

                                    while(n!=0)

{

                                                rem=n%10;

                                                if(rem%2!=0)

                                                sum=sum+rem;

                                                n=n/10;

                                    }

                                    System.out.println (sum);

                        }

            }

}

**20.Largest Array**

Write a program which takes two arrays of the same size as a input and compares the first element of first array with the first element of second array and stores the largest of these into the first element of the output array. Repeat the process till the last element of the first array is checked with the last element of the second array.

Include a function named **largestArray** that accepts 3 arguments and its return type is void. The first argument is input array 1, the second argument is  input array 2 and the third argument is an int that corresponds to the size of the array. The output array is stored in a global variable named output1.

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 the largest array.

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

**Sample Input 1:**

4

2

1

3

4

1

9

2

8

**Sample Output 1:**

2

9

3

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.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,j,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)

                                                {

                                                            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 output1[]=new int[100];

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

                                     {

                                              if(a[i]>b[i])

{

                                                            output1[i]=a [i];

                                                    }

                                                    else

{

                                                               output1[i]=b[i];

                                                     }

                                      }

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

{

                                                   System.out.println(output1[i]);

                                      }

                        }

            }

}

**21.Sum of Prime Numbers**

Write a program to find the sum of the prime numbers present in the given input array.

Include a function named**sumPrime**that accepts 2 arguments and returns an int. The first argument is a pointer to the input array and the second argument is an int that corresponds to the size of the array. The function returns the sum of the prime numbers in the input 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.

Please note that 1 is neither prime nor composite.

**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.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

5

2

4

8

9

11

**Sample Output 1:**

13

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,flag=0, sum=0,count=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);

                                                }

                                    }

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

                                    {

                                                   count=0;

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

                                                   {

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

                                                             count++;

                                                    }

                                                    if(count==2)

                                                            sum=sum+a[i];

                                     }

                                     System.out.print(sum);

                        }

            }

}

**22.Leap Year**

Write a program to find whether the given input year is a Leap Year.

Include a function named **checkLeapYear** that accepts an integer and returns an integer. The function returns

1. 1 if the input is a Leap Year
2. 0 if the input is not a Leap Year
3. -1 if the input is a negative number

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

2000

**Sample Output 1:**

yes

**Sample Input 2:**

1610

**Sample Output 2:**

no

**Sample Input 3:**

**-**2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int year;

                        Scanner in=new Scanner(System.in);

                        year = in.nextInt();

if(year<0)

                        {

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

                                    System.exit(0);

                        }

                        if(year%4==0)

                                    System.out.println("yes");

                        else

                                    System.out.println("no");

            }

}

**23.secondLargest**

Read the question carefully and follow the input and output format.  
  
Write a function to find second largest number in the given input integer array.  
  
Assume that all elements in the input array are unique.  
  
**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, which is the second largest.  
  
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 secondLargest(int array[], int size) whose return type is an integer, the second largest.  
  
**Sample Input 1:**  
5  
3  
342  
53  
2  
12  
  
**Sample Output 1:**  
53  
  
**Sample Input 2:**  
5  
3  
342  
53  
-2  
  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,largest=0,temp;

                        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++)

                                   {

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

                                                {

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

                                                              {

                                                                        temp = a[j] ;

                                                                        a[j] = a [i] ;

                                                                        a [i] = temp ;

                                                               }

                                                 }

                                      }

                                       System.out.println(a[1]);

                        }

            }

}

**24.findFirstLargest**

Read the question carefully and follow the input and output format.  
  
Write a function to find the product of the first and the third largest numbers in the given input integer array.  
  
**Input and Output Format :**First line of input corresponds to n, the size of array and next n lines correspond to the elements of the array  
  
Print "Invalid array size" when size of the array is a negative number and terminate the program  
Print "Invalid number" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named findFirstLargest(int n, int array[]) that returns an integer, product of first and third largest numbers.  
  
**Sample Input 1:**  
5  
11  
241  
83  
8  
2  
  
**Sample Output 1:**  
2651  
  
**Sample Input 2:**  
3  
83  
-8  
**Sample Output 2:**  
Invalid number

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,k,temp;

                        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 number");

                                                            System.exit(0);

                                                }

                                    }

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

                                      {

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

                                                 {

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

                                                            {

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

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

                                                                       n--;

                                                            }

                                                             else

                                                             j++;

                                                  }

                                       }

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

                                       {

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

                                                  {

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

                                                             {

                                                                       temp = a[j] ;

                                                                       a[j] = a[i] ;

                                                                       a[i] = temp ;

                                                              }

                                                   }

                                        }

                                        System.out.print(a[0] \* a[2]);

                        }

            }

}

**25.generateCode**

Read the question carefully and follow the input and output format.  
  
In a game show everybody got one coupon with some code. They need to generate a code with only even numbers in that coupon. Find the answer.  
  
**Input and Output Format :**  
Input consists of  an integer. Output consist of an integer, which is the generated code.  
  
1) Print "Number too small" when the given input number is a negative number.  
2) Print "Number too large" when the given input number is greater than 32767.  
3) Print 0 If the coupon does not contain any even numbers.  
  
Include a function named generateCode(int coupon) whose return type is an integer, which is the generated code.  
  
**Sample Input 1:**  
4352  
  
**Sample Output 1:**  
42  
  
**Sample Input 2:**  
1357  
  
**Sample Output 2:**  
0  
  
**Sample Input 3:**  
-1357  
  
**Sample Output 3:**  
Number too small

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i=1,n,flag=0,rem,res=0;

                        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;

                                     if(rem%2==0)

                                     {

                                                res=res+(rem\*i);

                                                i=i\*10;

                                      }

                                       n=n/10;

                       }

                       System.out.println(res);

            }

}

**26.calculateBonus**

Read the question carefully and follow the input and output format.  
  
Given the basic salary as input, write a program to calculate the bonus and display it.  
  
The bonus will be calculated based on the below category.  
Basic>20000 bonus=17%of basic+1500  
Basic>15000 bonus=15%of basic+1200  
Basic>10000 bonus=12%of basic+1000  
for rest =8%of basic+500  
  
**Input and Output Format :**  
  
First line of input consists of n, the basic salary.  
Output is a single integer that displays the bonus.  
  
Print "Number too large" when the given input numbers is greater than 32767 .  
Print "Number too small" when the given input is a negative number.  
  
Include a function named calculateBonus(int basic) whose return type is an integer, the bonus.  
  
**Sample Input 1:**  
21000  
  
**Sample Output 1:**  
5070  
  
**Sample Input 2:**  
327678  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,res=0;

                        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);

}

                      if(n>20000)

                        res=(n\*17/100)+1500;

                        else if(n>15000)

                        res=(n\*15/100)+1200;

                        else if(n>10000)

                        res=(n\*12/100)+1000;

                         else

                        res=(n\*8/100)+500;

                        System.out.println(res);

            }

}

**27.Minimum of 3**

 Write a program to find the minimum of 3 numbers.

 Include a function named **findSmallest** that accepts 3 integer arguments and returns an integer that corresponds to the minimum value.

**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:**

3

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                                int a,b,c;

                        Scanner in=new Scanner(System.in);

                        a = in.nextInt();

b = in.nextInt();

c = in.nextInt();

                        if(a<b && a<c)

                        System.out.print(a);

                        else if(b<a && b<c)

                        System.out.print(b);

                        else

                        System.out.print(c);

            }

}

**28.Sort and Delete**

Write a program to delete the given number in the input array and then to sort the array.

 Include a function named **sortAndDelete** that accepts 3 arguments and its return type is void. The first argument is  the input array and the second argument is an int that corresponds to the size of the array and the third argument is the array element to be deleted. The number of elements in the modified array is stored in the global variable named output1.

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

 Please note that the elements in the array may not be unique.

**Input and Output Format:**

Input consists of n+2 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. The last integer corresponds to the element to be deleted. Output consists of an integer array.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

8

1

6

3

5

8

10

4

8

8

**Sample Output 1:**

1

3

4

5

6

10

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,k=0,temp,elt;

                        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 number");

                                                            System.exit(0);

                                                }

                                    }

                                    elt=in.nextInt();

                                    int output1[]=new int[20];

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

{

                                                if(a[i]!=elt)

{

                                                            output1[k]=a [i];

                                                            k++;

                                                 }

                                     }

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

{

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

                                                             if(output1[i]>output1[j])

{

                                                                        temp = output1[j];

                                                                        output1[j] = output1[i];

                                                                        output1[i] = temp;

                                                              }

                                                 }

                                   }

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

{

                                              System.out.println(output1[i]);

                                    }

                        }

            }

}

**29.generateNewNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to generate new number from the given input based on following conditions.  
  
(i) Even digit should be replaced by next Even digit.  
(ii) Odd digit should be replaced with next Odd digit  
  
**Input and Output Format :**  
Input consists of an integer. Output is also an integer.  
  
1) Print "Number too small" when any of given input numbers is a negative number.  
2) Print "Number too large" when any of given input numbers is greater than 32767.  
  
  
**Sample Input 1:**  
123  
  
**Sample Output 1:**  
345  
  
**Sample Input 2:**  
32768  
  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i=1,n, rem,res=0;

                        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;

                                    if(rem%2==0)

{

                                            res=(rem+2)\*i+res;

                                                 i=i\*10;

                                     }

                                     else

                                     {

                                                 res=(rem+2)\*i+res;

                                                 i=i\*10;

                                     }

                                     n = n/10;

                        }

                        System.out.println (res);

            }

}

**30.findMileage**

Read the question carefully and follow the input and output format.  
  
Given the cubic capacity(CC) of a bike. Write a function to return the mileage/liter for the given Cubic Capacity(CC). The mileage will be calculated as follows:  
  
if CC is between 100 and 125, mileage is 75  
if CC is between 126 and 135, mileage is 70  
if CC is between 136 and 150, mileage is 60  
if CC is between 151 and 200, mileage is 50  
if CC is between 201 and 220, mileage is 35  
  
First line of input consists of an integer that corresponds to CC of a bike. Output consist of an integer, which is the mileage.  
  
Print "Number too large" when the given input CC is greater than 220.  
Print "Number too small" when the given input CC is less than 100.  
  
Include a function named findMileage(int cc) whose return type is an integer, which is the mileage.  
  
**Sample Input 1:**  
1  
  
**Sample Output 1:**  
Number too small  
  
**Sample Input 2:**  
160  
  
**Sample Output 2:**  
50

import java.util.\*;

public class Main

{

            public static void main(String[] args)

            {

                        int cc,mil;

                        Scanner in=new Scanner(System.in);

                        cc = in.nextInt();

                        if(cc<100)

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

                        else if(cc>220)

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

                        else

{

                                    if(cc>=100 && cc<=125)

                                    mil=75;

                                    else if(cc>=126 && cc<=135)

                                    mil=70;

                                    else if(cc>=136 && cc<=150)

                                    mil=60;

                                    else if(cc>=151 && cc<=200)

                                    mil=50;

                                    else

                                    mil=35;

                                    System.out.println (mil);

                        }

            }

}

**31.sumPrimeArray**

Read the question carefully and follow the input and output format.  
  
John is working in a bank. He has created account details transaction in a file and protected it with a password. He sent the file to his manager for review. The file is protected with a password. The password is the sum of Prime numbers. Write a function to generate the password.  
  
**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, which is the sum.  
  
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.  
3) Print 0, when there are no prime numbers in a given input array.  
  
Include a function named sumPrimeArray(int array[], int size) whose return type is an integer, which is the prime sum.  
  
**Sample Input 1:**  
5  
1  
2  
3  
4  
5  
  
**Sample Output 1:**  
10  
  
**Sample Input 2:**  
3  
4  
8  
9  
  
**Sample Output 2:**  
0

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j, sum=0,count=0;

// int 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++)

                                                {

                                                            count=0;

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

                                                            {

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

                                                                        count++;

                                                            }

                                                            if(count==2)

                                                            sum=sum+a[i];

                                                }

                                                System.out.print(sum);

                                    //}

                        }

            }

}

**32. avgOddKeyValues**

Read the question carefully and follow the input and output format.  
  
  
Given an input array, First index represents key and second index represents the value and so on... Write code to find out the average of all values whose keys are odd numbers.  
  
**Input and Output Format :**  
First line of input consists of n, the next n lines correspond to the elements of the array. Output consist of the an integer.  
  
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.  
  
**Sample Input 1:**  
8  
1  
3  
2  
4  
3  
16  
4  
25  
**Sample Output 1:**  
9  
  
**Sample Input 2:**  
-3  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i, sum=0,count=0,avg;

// int 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=i+2)

                                                {

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

                                                            {

                                                                        sum=sum+a[i+1];

                                                                        count++;

                                                            }

                                                }

                                                avg = sum/count;

                                                System.out.print(avg);

                                    // }

                        }

            }

}

**33.Product of MaxMin Element**

Write a program to find the product of the maximum and minimum element in a given input array.

 Include a function named **productOfMaxMin** 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 product of maximum and minimum 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 product of maximum and minimum element in the array.

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

**Sample Input 1:**

8

2

12

3

4

6

8

10

9

**Sample Output 1:**

24

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,j,temp,prod;

//int 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++)

                                                {

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

                                                            {

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

                                                                        {

                                                                                   temp = a[j];

                                                                                    a[j] = a[i];

                                                                                    a[i] = temp;

                                                                        }

                                                            }

                                                }

                                                prod = a[0] \* a[n-1];

                                                System.out.print(prod);

                                    //}

                        }

            }

}

**34.registerAccountNumbers**

Read the question carefully and follow the input and output format.  
  
Given an array in which the elements are in xxxyy format, where first xxx digits represent the Branch code and the yy represents the account  
id.  Find out the No of accounts in the given branch code  
  
**Input and Output Format :**  
The first input n corresponds to the size of the array, the next n lines correspond to the elements of the array and the last line of the input corresponds to the branch code.  
Output corresponds to the number of accounts in the given branch code  
If the given branch code is not available, print 0.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid account Number" when there is any negative number available in the input array and terminate the program  
3) Print "Invalid branch code" when branch code is negative number and terminate the program  
  
Include a function named registerAccountNumbers (int size, int account\_numbers[], int branch\_code) that returns the no of accounts  
  
**Sample Input 1 :**  
6  
12345  
12370  
12324  
13355  
13333  
14575  
123  
  
**Sample Output 1 :**  
3  
  
**Sample Input 2 :**  
-6  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,bcode,code,temp,count=0;

//int 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 account number");

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                   // {

                                                bcode=in.nextInt();

                                                if(bcode<0)

                                                {

                                                            System.out.println("Invalid branch code");

                                                            System.exit(0);

                                                }

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

                                                {

                                                            temp=a[i];

                                                            code=temp/100;

                                                            if(code==bcode)

                                                            count++;

                                                }

                                                System.out.print(count);

                                   // }

                        }

            }

}

**35.newArraySum**

Read the question carefully and follow the input and output format.  
  
Given an input array which contains age of some employees, write a program to find the sum of ages of employees greater than 18.  
  
**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, which is the sum.  
  
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 newArraySum(int age[],int size) whose return type is an integer, which is the sum.  
  
**Sample Input 1:**  
5  
21  
22  
17  
10  
25  
**Sample Output 1:**  
68  
  
**Sample Input 2:**  
6  
50  
-36  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i, sum=0;

// int 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]>18)

                                                            sum=sum+a[i];

                                                }

                                                System.out.print(sum);

                                  //  }

                        }

            }

}

**36.Array Product**

Write a program to find the product of positive/nonnegative elements in a given array.

 Include a function named **calculateProduct** 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 product.

If the size of the array is negative or if it is greater than 10 or if any element in the array is more than 2 digits, 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 product of positive numbers in the array.

**Sample Input 1:**

8

1

-2

3

4

-6

8

10

-6

**Sample Output 1:**

960

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

200

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i, prod=1;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n < 0 || n>10)

                        {

                                    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]>=100 || a[i]<=-100)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    //if(flag!=1)

                                    //{

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

                                                {

                                                            if(a[i]>0)

                                                            prod=prod\*a[i];

                                                }

                                                System.out.print(prod);

                                    //}

                        }

            }

}

**37.Perfect Number**

Write a program to find whether the given number is a perfect Number.

A number is a perfect number if the sum of the proper divisors of the number is equal to the number itself.

Include a function named **findPerfect** that accepts an integer argument and returns an integer. The function returns

1.       1 if the input is a Perfect Number

2.       0 if the input is not a Perfect Number

3.       -1 if the input is a negative number or if it is greater than 32767

**Input and Output Format:**

Input consists of a single integer.

Output consists of a string.

Refer sample output for formatting specifications.

**Sample Input 1:**

6

**Sample Output 1:**

yes

**Sample Input 2:**

241

**Sample Output 2:**

no

**Sample Input 3:**

-9

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i,sum=0;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    // flag=1;

System.exit(0);

                        }

                        if(n>32767)

                        {

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

                                    // flag=1;

System.exit(0);

                        }

                       // if(flag==0)

                       // {

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

                                    {

                                                if(n%i==0)

                                                sum=sum+i;

                        //    }

                            if(sum==n)

                                    System.out.println("yes");

                            else

                                    System.out.println("no");

                        }

            }

}

**38.Squares of Even Digits**

Write a program to find the sum of the squares of even digits in a number.

Include a function named sumSquareEven that accepts an integer argument and returns an integer . The function returns -1 if the number is less than zero or if the number is greater than 32767.

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of an integer.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

3487

**Sample Ouput 1:**

80

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,n ,rem,res=0;

                        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;

                                     if(rem%2==0)

                                     res=res+(rem\*rem);

                                     n=n/10;

                        }

                         System.out.println(res);

            }

}

**39.sumPrimeArray**

Read the question carefully and follow the input and output format.  
  
John is working in a bank. He has created account details transaction in a file and protected it with a password. He sent the file to his manager for review. The file is protected with a password. The password is the sum of Prime numbers. Write a function to generate the password.  
  
**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, which is the sum.  
  
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.  
3) Print 0, when there are no prime numbers in a given input array.  
  
Include a function named sumPrimeArray(int array[], int size) whose return type is an integer, which is the prime sum.  
  
**Sample Input 1:**  
5  
1  
2  
3  
4  
5  
  
**Sample Output 1:**  
10  
  
**Sample Input 2:**  
3  
4  
8  
9  
  
**Sample Output 2:**  
0

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j, sum=0,count=0;

// int 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++)

                                                {

                                                            count=0;

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

                                                            {

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

                                                                        count++;

                                                            }

                                                            if(count==2)

                                                            sum=sum+a[i];

                                                }

                                                System.out.print(sum);

                                    //}

                        }

            }

}

**40. Array Multiplication in Reverse**

A company wanted to know the reward points of the employee so that at the end of every month they will credit some amount along with their salary. Each employee has 2 separate lists, in first list records will be sorted in employee’s employee number in ascending order. Second list records will be sorted in employee’s employee number in descending order. Hence the management has decided to multiply both the reward points and credit the amount based on the points. Here they followed the formula for multiplying the first entry value in the first list with the last entry value in the second list and second entry from the first list with the second last record from the second list. Repeat the same for all the entries in the lists.

 Include a function named **arrayProduct** that accepts 3 arguments and ints return type is void. The first argument is the input array 1, the second argument is  the input array 2 and the third argument is an int that corresponds to the size of the array. The output array is stored in a global variable named output1.

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

**Input and Output Format:**

Input consists of utmost 2n+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 first array. The last 'n' integers correspond to the elements in the second array. If any of the inputs are invalid, then terminate the program.

 Output consists of n integers that correspond to the elements in the result array.

 Assume that the maximum size of the array is 20.

**Sample Input 1 :**

5

23

2

5

32

76

2

2

21

42

4

**Sample Output 1 :**

92

84

105

64

152

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j;

// int 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);

                                                }

                                    }

                                     int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                    //{

                                                int c[]=new int[20];

                                                for(i=0,j=n-1;i<n&&j>=0;i++,j--)

{

                                                            c[i] = a[i]\*b[j];

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

                                                }

                                  //}

                        }

            }

}

**41.University Type**

 Write a program to find if the student is eligible for first, second or third grade universities by finding the average of their marks given in the input integer array.

 Grade should be calculated as given below :

Average >80 First Grade University

Average >60 Second Grade University

Otherwise Third Grade University

 Include a function named **calculateGrade** that accepts 2 arguments and returns an integer. 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 integer that corresponds to the university type. The function returns 1 if the student is eligible for First Grade univesity, returns 2 if the student is eligible for Second Grade University, returns 3 if the student is eligible for Third Grade University and returns -1 if the average is greater than 99.

If the size of the array is negative or if any element in the array is negative or if the average marks scored by the student is greater than 99 , 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 string --- “First Grade University” or “Second Grade University” or “Third Grade University” or “Invalid Input”.

Assume that the maximum size of the array is 20.

**Sample Input 1:**

5

92

87

78

74

80

**Sample Output 1:**

First Grade University

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, sum=0,avg=0;

// int 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++)

                                                            sum=sum+a[i];

                                                avg=sum/n;

                                                if(avg>99)

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

                                                else if(avg>80)

                                                System.out.print ("First Grade University");

                                                else if(avg>60)

                                                System.out.print ("Second Grade University");

                                                else

                                                System.out.print ("Third Grade University");

                                   //}

                        }

            }

}

**42.Interchange Array**

 Write a program to interchange the first element in the array with the last element in the array. Repeat the process till the middle of the array.

 Include a function named **interchangeArray** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size 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 the interchanged array.

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

**Sample Input 1:**

4

2

1

3

4

**Sample Output 1:**

4

3

1

2

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,temp=0;

// int 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,j=n-1;i<n/2;i++,j--)

                                                {

                                                            temp = a [i];

                                                            a [i] = a [j];

                                                            a [j] = temp;

                                                }

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

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

                                    //}

                        }

            }

}

**43.Armstrong Number**

 Write a program to find whether the given input number is an Armstrong Number.

 Include a function named **checkArmstrong** that accepts an integer and returns an integer. The function returns

1.       yes if the input is an Armstrong number

2.       no if the input is not an Arnstrong number

3.       Invalid Input if the input is a negative number or if the input is not a 3-digit number.

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

153

**Sample Output 1:**

yes

**Sample Input 2:**

161

**Sample Output 2:**

no

**Sample Input 3:**

2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, sum=0,temp,rem;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0||n>999)

                        {

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

                                    flag=1;

                        }

                       // if(flag==0)

                        {

                                     temp=n;

                                    while (n != 0)

                                    {

                                                rem = n%10;

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

                                                n = n/10;

                                    }

                                     if (temp == sum)

                                     System.out.println("yes");

                                    else

                                    System.out.println("no");

                        //}

            }

}

**44.Factorial**

Write a program to find the factorial of a given number.

Include a function named **findFactorial** that accepts an integer argument and returns an integer that corresponds to factorial. If the input value is negative or greater than 10, the function returns -1.

If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

4

**Sample Output 1:**

24

**Sample Input 2:**

-67

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, fact=1,i;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0||n>10)

                        {

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

                                    flag=1;

                        }

                        //if(flag==0)

                        //{

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

                                    {

                                                fact = fact \* i;

                                    }

                                    System.out.println (fact);

                        //}

            }

}

**45.Salary Calculation**

Jim got his salary. His salary calculations are as follows.

From his Basic amount he gets 50% of his basic for house Rent allowances and 75% of his basic as special allowances . If the number of days he worked is 31 he gets 500 extra. Write a program to calculate his gross salary after calculating all his salary split up.

Include a function named **calculateGross** that accepts 2 integer arguments and returns a float. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days Jim has worked. The function returns a float that corresponds to the gross salary.

 Print Invalid Input and terminate the program in the following cases:

1.       Basic salary is greater than 10000

2.       Number of working days is greater than 31

3.       Basic salary is negative

4.       Number of working days is 0 or negative

**Input and Output Format:**

Input consists of 2 integers. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days he has worked.

Output consists of a single float that corresponds to Jim's gross salary. The gross salary is displayed correct to 2 decimal places.

**Sample Input 1:**

5000

30

**Sample Output 1:**

11250 .00

**Sample Input 2:**

5000  
0  
**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int basic=0,days=0;

                        double salary;

                        Scanner in=new Scanner(System.in);

                        basic=in.nextInt();

                        days= in.nextInt();

                        if(basic>10000 || days>31 || basic<0 || days==0 || days<0)

{

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

                                    System.exit(0);

                        }

                        if(days==31)

                        salary=(basic\*50)/100+(basic\*75)/100+basic+500;

                        else

                        salary=(basic\*50)/100+(basic\*75)/100+basic;

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

            }

}

**46.Perfect Square**

 Write a program to find whether the given input number is a perfect square without using sqrt function.

 Include a function named **checkPerfectSquare** that accepts an integer and returns an integer. The function returns

1.       1 if the input is a perfect square

2.       0 if the input is not a perfect square

3.       -1 if the input is a negative number

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

36

**Sample Output 1:**

yes

**Sample Input 2:**

40

**Sample Output 2:**

no

**Sample Input 3:**

**-**2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,fact=1;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    //flag=1;

System.exit(0);

                        }

                       // if(flag==0)

                       // {

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

                                    {

                                                fact=fact\*i;

                                                if(fact==n)

                                                {

                                                            System.out.println("yes");

                                                            System.exit(0);

                                                }

                                    }

                                    System.out.println("no");

                      //  }

            }

}

**47.Find Index**

 Write a program to find the index of a particular number in a given input array.

Include a function named **findIndex** that accepts 3 arguments and returns an int. The first argument is the input array, the second argument is an int that corresponds to the size of the array and the third argument is the element to be searched for. The function returns the corresponding index if the search element is present in the array and returns -1 if the search element is not present in 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+2 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.

The last integer corresponds to the element whose count needs to be found.

Output consists of an integer that corresponds to the index of the search element if it is present.

Else, print 'not found'.

Refer sample output for formatting specifications.

Assume that the maximum number of elements in the array is 20 and that all elements in the array are unique.

**Sample Input 1:**

8

2

1

3

8

6

12

10

19

8

**Sample Output 1:**

3

**Sample Input 2:**

8

2

1

3

8

6

12

10

19

80

**Sample Output 2:**

not found

**Sample Input 3:**

-5

**Sample Output 3:**

Invalid Input

**Sample Input 4:**

5

23

2

-200

**Sample Output 4:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,se,index=0,f=0;

// int 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)

                                   // {

                                                se=in.nextInt();

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

                                                {

                                                            if(a [i]==se)

{

                                                                        f=1;

                                                                        index = i;

                                                                        System.out.print(index);

                                                                        System.exit(0);

                                                }

                                                }

                                                if(f==0)

                                                System.out.print("not found");

                                    //}

                        }

            }

}

**48.Descending Order Sort**

 Write a program to sort the given array in descending order.

 Include a function named **sortArray** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size of the array. If the size of the array is negative or if any of the elements in the array are 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 array.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

8

1

6

3

5

8

10

4

9

**Sample Output 1:**

10

9

8

6

5

4

3

1

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,temp=0;

// int 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++)

{

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

{

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

{

                                                                                    temp = a[i];

                                                                                    a[i] = a[j];

                                                                                    a[j] = temp;

                                                                        }

                                                            }

                                                }

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

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

                                    //}

                        }

            }

}

**49.endWithThree**

Read the question carefully and follow the input and output format.  
  
Given an input array, Find out the count of numbers that ends with 3.  
  
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 the count of numbers that ends with 3.  
  
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.  
  
**Sample Input 1:**  
5  
23  
353  
33  
12  
14  
**Sample Output 1:**  
3  
  
**Sample Input 2:**  
5  
1  
7  
3  
-8  
**Sample Output  2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j, temp,count=0,rem;

// int 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++)

{

                                                            temp = a[i];

                                                            rem = temp%10;

                                                            if(rem==3)

                                                            count++;

                                                }

                                                System.out.print(count);

                                   // }

                        }

            }

}

**50.feeCalculation**

Read the question carefully and follow the input and output format.  
  
Student Fees is calculated according to the student's 10th marks. The student will get discount in fees as follows :  
  
Marks discount(%)  
>90         -  50%  
81-90      - 25%  
70-80      - 10%  
<70         -  0%  
Calculate the fees according to above table.  
  
**Note:**  
Formula : fees - (fees\* discount(%))  
Print "Invalid mark" if the mark is greater than 100  
Print "Invalid fee" if the fee is greater than 32767  
Print "Invalid input" if any of the input is negative  
  
**Input and Output Format :**  
First line of input represents the fee, second line of input represents the marks of student.  
  
**Sample Input 1:**  
10000  
95  
**Sample Output 1:**  
5000  
  
**Sample Input 2:**  
15896  
101  
**Sample Output 2:**  
Invalid mark

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int fees,marks;

                        double discount,fee\_final=0;

                        Scanner in=new Scanner(System.in);

                        fees = in.nextInt();

                        marks = in.nextInt();

                        if(marks>100)

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

                        else if(fees>32767)

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

                        else if(marks<0 || fees<0)

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

                        else

{

                                    if(marks>90)

                                    discount = .50;

                                    else if(marks>80 && marks<=90)

                                    discount = .25;

                                    else if(marks>=70 && marks<=80)

                                    discount = .10;

                                    else

                                    discount = 0;

                                    fee\_final = fees-(fees\*discount);

                                    System.out.printf("%.0f",fee\_final);

                        }

            }

}

**51.sumTwoFive**

Read the question carefully and follow the input and output format.  
  
Given an integer array find the sum of elements which end with 2 or 5.  
  
**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, the sum.  
  
1) Print "Invalid array size" when size of the array is negative .  
2) Print "Invalid input" when there is any negative number available in the input array and terminate the program.  
  
Include a function named sumTwoFive(int array[], int size) whose return type is an integer, the sum  
  
**Sample Input 1:**  
5  
22  
35  
5  
2  
10  
  
**Sample Output 1:**  
64  
  
**Sample Input 2:**  
-6  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, temp,sum=0;

// int 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++)

{

                                                            temp = a[i];

                                                            rem = temp % 10;

                                                            if(rem==2 || rem==5)

                                                            sum = sum+a[i];

                                                }

                                                System.out.print(sum);

                                   // }

}

            }

}

**52.Sum of Odd Digits**

Write a program to find the sum of the odd digits in a number.

Include a function named **sumOddDigits**that accepts an integer argument and returns an integer that corresponds to the sum of the odd digits. The function returns -1 if the input is less than zero or if it is greater than 32767.

If the function returns -1, print “Invalid Input”.

**Input and Output Format:**

The input consists of an integer.

The output consists of an integer that corresponds to the sum of the odd digits in the number.

**Sample Input 1:**

3487

**Sample Ouput 1:**

10

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rem,sum=0;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    flag=1;

                        }

                        if(n>32767)

                        {

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

                                   // flag=1;

                        }

                        //if(flag==0)

                        //{

                                    while (n != 0)

                                    {

                                                rem=n%10;

                                                if(rem%2!=0)

                                                sum=sum+rem;

                                                n=n/10;

                                    }

                                    System.out.print(sum);

                       // }

            }

}

**53.LCM**

Write a program to calculate the LCM of the 2 given integers.

Include a function named **calculateLCM** that accepts 2 integer arguments and returns an int that corresponds to the LCM of the 2 numbers.

Print Invalid Input and terminate the program in the following cases:

1.       Any of the 2 inputs is greater than 1000

2.       Any of the 2 inputs is negative

**Input and Output Format:**

Input consists of 2 integers.

Output consists of a single integer that corresponds to the LCM.

**Sample Input 1:**

10

8

**Sample Output 1:**

40

**Sample Input 2:**

50000

**Sample Output 2:**

Invalid Input

import java.util.\*;

public class Main

{

            public static void main(String[] args)

            {

                        int a, b, i, c=0;

                        Scanner in=new Scanner(System.in);

                        a = in.nextInt();

                        b = in.nextInt();

                        if(a>1000 || b>1000 || a<0 || b<0)

                        {

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

                                    System.exit(0);

                        }

                         c=a\*b;

                         int d=c;

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

                        {

                                    if(i%a==0 && i%b==0 && i<d)

                                    d=i;

                        }

                        System.out.println(d);

            }

}

**54.Product of Prime Digits**

Write a program to find the product of the prime digits in the given input number.

Include a function named **productPrimeDigits** that accepts an integer argument and returns an integer that corresponds to the product of the prime digits in the integer.

The function returns -1 if the input number is negative or greater than 32767.

If the function returns -1, print Invalid Input.

 Please note that 1 and 0 are neiher prime nor composite.

**Input and Output Format:**

Input consists of an integer.

Output consists of an integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

324

**Sample Output 1:**

6

**Sample Input 2:**

-67

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,n, rem,prod=1,f=0,count=0;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    flag=1;

                        }

                        if(n>32767)

                        {

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

                                    flag=1;

                        }

                       // if(flag==0)

                       // {

                                    while (n != 0)

                                    {

                                                rem=n%10;

                                                count=0;

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

{

                                                            if(rem%i==0)

                                                            count++;

                                                }

                                                if(count==2)

{

                                                            f=1;

                                                            prod = prod\*rem;

                                                }

                                                n=n/10;

                                    }

                                    if(f==1)

                                    System.out.println (prod);

                       // }

            }

}

**55.dailyAllowance**

Read the question carefully and follow the input and output format.  
  
A Sales person daily allowances calculated as follows .  
Item              Money (rupees)  
Shirt                   15  
Saree                 10  
other items          5  
  
Given an input array in which the first index represents no.of shirts sold, second index represents the no of sarees sold  and the third index represents the other items sold for a particular day, Calculate the total allowances.  
  
Inlcude a function named dailyAllowance(int items[], int size) that returns an integer, which is the total allowances.  
  
Business Rules:  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid item count" when there is any negative numbers available in the input array and terminate the program  
3) Print "Array size greater than 3"  when size of the array is greater than 3 and terminate the program.  
  
**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 the total allowance.  
  
**Sample Input 1**:  
3  
10  
5  
10  
**Sample Output 1:**  
250  
  
**Sample Input 2:**  
4  
  
**Sample Output 2:**  
Array size greater than 3

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n < 0)

                        {

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

                                    System.exit(0);

                        }

                        else if(n>3)

{

System.out.print("Array size greater than 3");

                                    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 item count ");

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                  //  {

int shirt=15,saree=10,other=5,allowance=0;

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

{

                                                            allowance = (a[0]\*shirt) + (a[1]\*saree) + (a[2]\*other);

                                                }

                                                System.out.print(allowance);

                                   // }

                        }

            }

}

**56.generateNumber**

Read the question carefully and follow the input and output format.  
  
Given an input number keep the prime digits as it is and the remaining digit in given input number should be replaced with next number(add +1 to the digit).  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
note : If any digit in the given input is 9 replace with 0. Consider 1 is not a prime number.  
  
Include a function generateNumber(int number) that returns the generated number  
  
**Input and Output Format :**  
Input is a single integer.  
Output is the generated number  
  
**Sample Input 1:**  
6234  
  
**Sample Output 1:**  
7235  
  
**Sample Input 2:**  
32768  
  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i=1,n, rem,res=0;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    //flag=1;

System.exit(0);

                        }

                        if(n>32767)

                        {

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

                                    //flag=1;

System.exit(0);

                        }

                       // if(flag==0)

                        //{

                                    while (n != 0)

                                    {

                                                rem=n%10;

                                                if(rem==2 || rem==3 || rem==5 || rem==7)

{

                                                            res=res+(rem\*i);

                                                            i=i\*10;

                                                }

                                                else if(rem==9)

{

                                                            res=res+0;

                                                            i=i\*10;

                                                }

                                                else

{

                                                            res=res+((rem+1)\*i);

                                                            i=i\*10;

                                                }

                                                n = n/10;

                                    }

                                    System.out.println (res);

                        //}

            }

}

**58.findEmployeeID**

Read the question carefully and follow the input and output format.  
  
In a given input array, elements are given in this Format PPPII, where PPP represents the Project Code and II represents employee id . Find out the Employee Ids who are working in the given project code.  
  
**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 corresponds to the project code. Output consists of the employee id's.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid employee id" when there is any negative numbers available in the input array and terminate the program  
3) Print "Invalid project code" when branch code is negative number and terminate the program  
  
Include a function named findEmployeeID(int size,int employee\_ids[],int project\_code). Its return type is void.  
The output array is stored in a global variable named working\_employee.  
  
**Sample Input 1:**  
5  
12345  
12334  
23457  
23478  
12546  
123  
**Sample Output 1:**  
45  
34  
  
**Sample Input 2:**  
5  
12345  
12334  
23457  
-23478

**Sample Output 2:**  
Invalid employee id

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, pcode=0,count=0,temp=0,code=0;

// int 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 employee id ");

                                                            System.exit(0);

                                                }

                                    }

                                    //if(flag!=1)

                                    //{

int b[]=new int[20];

                                                 pcode=in.nextInt();

                                                if(pcode<0)

{

                                                            System.out.print ("Invalid project code");

                                                            System.exit(0);

                                                }

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

{

                                                            temp=a[i];

                                                            code=temp/100;

                                                            if(code==pcode)

{

                                                                        b[count]=a[i]%100;

                                                                        count++;

                                                            }

                                                }

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

                                                System.out.println("%d\n",b[i]);

                                    //}

                        }

            }

}

**59.Array Addition in Reverse**

In an export company, the clothes are stored in boxes and are shipped in three different ships. Each box has a number on it. The boxes are stored in ships in specific pattern as below:

1. First box number in Ship1 + Last box number in Ship2 = first box number in Ship3.

2. Second box number in Ship1 + Second Last box number in Ship2 = Second box number in ship3.

Similarly all the boxes are numbered in ship3. Assuming the count of boxes in each ship is equal, find the box numbers in ship3.

Include a function named **arrayAddition** that accepts 3 arguments and its return type is void.  The first argument is the input array 1, the second argument is the input array 2 and the third argument is an int that corresponds to the size of the array.  The output is stored in an array variable named output1.

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

**Input and Output Format:**

Input consists of utmost 2n+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 first array(i.e box numbers in ship 1). The last 'n' integers correspond to the elements in the second array (i.e box numbers in ship 2). If any of the inputs are invalid, then terminate the program.

Output consists of n integers that correspond to the box numbers in ship 3.

Assume that the maximum size of the array is 20.

**Sample Input 1 :**

5

23

2

5

32

76

2

9

21

42

4

**Sample Output 1 :**

27

44

26

41

78

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j;

// int 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);

                                                }

                                    }

                                     int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

                                                           // flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                   // {

                                                int c[]=new int[20];

                                                for(i=0,j=n-1;i<n&&j>=0;i++,j--)

{

                                                            c[i] = a[i]+b[j];

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

                                                }

                                  //  }

                        }

            }

}

**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.  
**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);

                        }

            }

}

**61.Duplicate Elements**

 Tim is working as a data entry staff in a college. His manager wants him to delete the duplicate student id from the entry. Help Tim in writing a program to delete the duplicate elements.

 Include a function named **eliminateDuplicate** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The output array is stored in a global variable named output1 and the number of elements in the output array is stored in the global variable named output 2.

 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 array.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

8

1

6

3

5

6

8

5

9

**Sample Output 1:**

1

6

3

5

8

9

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j, k;

// int 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++)

{

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

{

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

{

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

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

                                                                                    n--;

                                                                        }

                                                                        else

                                                                        j++;

                                                            }

                                                //}

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

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

                                    }

                        }

            }

}

**62.Second Smallest**

 Write a program to find the second smallest of all divisors of the given number.

 For example, the divisors of 21 are 1,3,7 and 21. The second smallest divisor is 3.

 Include a function named **secondSmallest** that accepts an integer argument and returns an integer. The function returns the second smallest divisor or returns -1 if it is a negative number or if it is greater than 32767.

 If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer.

Output consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

21

**Sample Output 1:**

3

**Sample Input 2:**

-241

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

50000

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,n;

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);

                        }

                        else

                        {

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

{

                                                if(n%i==0)

{

                                                            System.out.println(i);

                                                            System.exit(0);

                                                }

                                    }

                        }

}

}

**63.secondMaxMinDiff**

Read the question carefully and follow the input and output format.  
  
Given an input array, find the difference b/w second largest and second smallest element in the array.  
  
Hint : There is no repetition of element in the array.  
  
**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, which is the difference b/w second largest and second smallest..  
  
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.  
  
Include a function named secondMaxMinDiff(int[] array, int n) whose return type is an integer, which is the difference b/w second largest and second smallest.  
  
**Sample Input 1:**  
5  
1  
2  
3  
4  
5  
  
**Sample Output 1:**  
2  
  
**Sample Input 2:**  
4  
-3  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j, temp=0,diff=0;

// int 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++)

{

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

{

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

{

                                                                                    temp = array[i];

                                                                                    array[i]=array[j];

                                                                                    array[j]=temp;

                                                                        }

                                                            }

                                                }

                                                diff=a [1]-a [n-2];

                                                System.out.print(diff);

                                    //}

                        }

            }

}

**64.newNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to find the difference between consecutive digits in the given input integer and display it.  
  
**Input and Output Format:**  
Input consists of an integer and output the difference between the consecutive digits.  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
  
  
**Sample Input 1:**1325  
**Sample Output 1:**  
213  
  
**Sample Input 2:**-13  
**Sample Output 2:**  
Number too small

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rem,i=1,rem1=0,diff=0,digit=0;

                        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);

                        }

                        else

                        {

                                    while(n>10)

{

                                                rem=n%10;

                                                n=n/10;

                                                rem1=n%10;

                                                if(rem>rem1)

                                                diff=rem-rem1;

                                                else

                                                diff=rem1-rem;

                                                digit=digit+(diff\*i);

                                                i=i\*10;

                                    }

                                    System.out.println(digit);

                        }

            }

}

**65.Digit Counting**

 In a game show everybody got one coupon with some code. They need to count the digits in the code and send SMS to the given number.

Write a program to find the number of digits in the given number.

 Include a function named **countDigits** that accepts an integer argument and returns an integer that corresponds to the number of digits. If the input is a negative number, the function returns -1.

 If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer.

Output consists of an integer that corresponds to the number of digits in the input.

**Sample Input 1:**

250

**Sample Output 1:**

3

**Sample Input 2:**

**-**2345

**Sample Output 2:**

Invalid Input

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, count=0;

                        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)

{

                                   count++;

                                   n=n/10;

                         }

                         System.out.println (count);

           }

}

**66.maxScoreCount**

Read the question carefully and follow the input and output format.  
  
Student1 and Student2 are of same class and have recieved their scores for different subjects. Given each subject scores of the Student1 & Student2, Find out in how many subjects student1 has scored more marks than student2.  
  
**Input and Output Format :**  
First line of input corresponds to the n, the number of subjects. The next n lines correspond to the scores of Student 1 and the next n lines correspond to the scores of student 2. Output is the number of subjects student1 scored more marks than student2.  
  
Print "Invalid size" when size of the array is a negative number and terminate the program  
Print "Invalid score"  when there is any negative score and terminate the program  
  
Include a function named maxScoreCount(int size,int student1[],int student2[]) that returns an integer, the number of subjects student1 scored more marks than student2  
  
**Sample Input 1:**  
7  
45  
23  
67  
34  
88  
13  
67  
33  
56  
89  
44  
67  
89  
55  
  
**Sample Output 1:**  
3  
  
**Sample Input 2:**  
-3

**Sample Output 2:**  
Invalid size  
  
**Sample Input 3:**  
2  
30  
-60  
**Sample Output 3:**  
Invalid score

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, count=0;

// int 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 score");

                                                            System.exit(0);

                                                }

                                    }

                                    int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                    {

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

{

                                                            if(a[i]>b[i])

                                                            count++;

                                                }

                                                System.out.print(count);

                                    }

                        }

            }

}

**67.nextPrime**

Read the question carefully and follow the input and output format.  
  
Write a program to find out the Next Prime to the given number.  
  
Hint: number is always less than 100.  
  
**Input and Output Format :**  
  
First line of input consists of n, the number. Output is a single integer that displays the next prime.  
  
Print "Number too large" when the given input number is greater than 32767 .  
Print "Number too small" when given input is a negative number.  
  
Include a function named nextPrime(int num) whose return type is an integer, the next prime.  
  
**Sample Input 1:**  
9  
  
**Sample Output 1:**  
11  
  
**Sample Input 2:**  
98987  
  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,j,n;

                        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);

                        }

                        else

                        {

                                    int next=0,count=0;

                                    for(i=n+1;i<100;i++)

{

                                                count=0;

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

{

                                                            if(i%j==0)

                                                            count++;

                                                }

                                                if(count==2)

                                                {

                                                            next=i;

System.out.println(next);

                                                }

                                    }

                                    if(next==0)

System.exit(0);

                        }

            }

}

**68.Array Average**

Write a program to find the average of the array.

 Include a function named **avgArray** 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:**

7

1

9

8

4

6

4

5

**Sample Output 1:**

5.29

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, sum=0;

// int flag=0;

                        double 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)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                    //{

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

{

                                                            sum=sum+a[i];

                                                }

                                                avg=sum/n;

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

                                    //}

                        }

            }

}

**69.Strong Number**

Write a program to find whether the given input number is a Strong Number

Strong Number : (In a number sum of Factorial of individual digits equals to the same number).

Include a function named **checkStrong** that accepts an integer and returns an integer. The function returns

1.       1 if the input is a Strong Number

2.       0 if the input is not a Strong Number

3.       -1 if the input is a negative number

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

145

**Sample Output 1:**

yes

**Sample Input 2:**

141

**Sample Output 2:**

no

**Sample Input 3:**

**-**2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,num,sum=0,i=0,f=0,r;

// int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

System.exit(0);

                                    //flag=1;

                        }

                        if(n>32767)

                        {

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

System.exit(0);

                                    //flag=1;

                        }

                        //if(flag==0)

                        //{

                                    num=n;

                                    while(n!=0)

                                    {

                                                i=1;

                                                f=1;

                                                r=n%10;

                                                while(i<=r)

                                                {

                                                            f=f\*i;

                                                            i++;

                                                }

                                                sum=sum+f;

                                                n=n/10;

                                    }

                                    if(sum==num)

                                    System.out.println("yes");

                                    else

                                    System.out.println("no");

                        //}

            }

}

**70.Sum of Even Digits**

In the computer science department, HOD wants to divide students into two groups based on their roll numbers. He decided to find the sum of the even digits in the roll number of each student and decided to split them into 2 groups based on this. Write a program to find the sum of the even digits in a number.

 Include a function named **addEvenDigits** to find the sum of even digits in a number. This function accepts an integer argument and returns an integer. The function returns -1 if the roll number is less than zero or if the roll number is greater than 32767. Refer function specifications given at the end of the problem for further details.

 If the roll number is less than 0 or if it exceeds 32767, print “Invalid Input”.

**Input and Output Format:**

The input consists of an integer that corresponds to the roll number.

The output consists of an integer that corresponds to the sum of the even digits in the roll number.

**Sample Input 1:**

3487

**Sample Ouput 1:**

12

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,n,flag=0;

                        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;

                                  if(rem%2==0)

                                  res=res+rem;

                                  n=n/10;

                         }

                         System.out.println(res);

            }

}

**71.commonElementsSum**

Read the question carefully and follow the input and output format.  
  
Given 2 integer arrays , write a program to find the sum of common elements in both the arrays.  
  
If there are no common elements print 0.  
  
**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, which is the sum  
  
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.  
  
**Sample Input 1:**  
4  
1  
2  
3  
4  
2  
3  
6  
7  
  
**Sample Output 1:**  
5  
  
**Sample Input 2:**  
3  
8  
6  
-7  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,flag=0,k=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[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);

                                                }

                                    }

                                    int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

                                                            flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    if(flag!=1)

                                    {

int common[]=new int[n];

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

{

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

{

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

{

common[k]=a[i];

k++;

}

}

}

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

sum=sum+common[i];

System.out.println(sum);

}

}

}

}

**72.Palindromic Number**

Write a program to find whether the given input number is a palindrome.

Include a function named checkPalindrome that accepts an integer and returns an integer. The function returns

1 if the input is a palindrome

0 if the input is not a palindrome

-1 if the input is a negative number

Print Invalid Input if the function returns -1.

**Input and Output Format:**

Input consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

2002

**Sample Output 1:**

yes

**Sample Input 2:**

167

**Sample Output 2:**

no

**Sample Input 3:**

-2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, sum=0,rev=0,rem,temp;

                        Scanner in=new Scanner(System.in);

                       n=in.nextInt();

                        if(n < 0)

                        {

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

                                    System.exit(0);

                        }

else

{

temp=n;

while(temp!=0)

{

rem=temp%10;

rev=rev\*10+rem;

temp=temp/10;

}

if(rev==n)

System.out.print("yes");

else

System.out.print("no");

}

}

}

**73.sumEvenIndex**

Read the question carefully and follow the input and output format.  
  
Write a program to find the sum of the indexes (positions) of even numbers in the Array. Consider 0 index as 1 and 1 index is 2 and so on……  
Note : Assume Array Index Starts From 1  
  
**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, which is the sum.  
  
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.  
  
**Sample Input 1:**  
7  
4  
2  
7  
9  
1  
10  
13  
  
**Sample Output 1:**  
9  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i, flag=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[20];

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

{

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

sum=sum+i;

}

System.out.print(sum);

}

}

}

}

**74.sumEvenOddProduct**

Read the question carefully and follow the input and output format.  
  
Write a program to find the sum of  product of even digits  and product of odd digits in a given number.  
  
If number contains only even numbers or odd numbers take the other numbers product as 1.  
  
**Input and Output Format :**  
Input consists of a single integer. Output consist of the sum of even digit product and odd digit product.  
  
Print "Number too large" when the given input number is greater than 32767  
Print "Number too small" when the given input number is a negative number.  
  
  
**Sample Input 1:**  
4564  
**Sample Output 1:**  
101  
  
{Hint :   (4\*6\*4) + (5)  =  96 +5 = 101}  
**Sample Input 2:**  
1357  
**Sample Output 2:**  
106  
  
**Sample Input 3:**  
981357  
**Sample Output 3:**  
Number too large

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, sum=0,rem,flag=0,evenprod=1,oddprod=1,res;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

flag=1;

}

if(n>32767)

{

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

flag=1;

}

if(flag==0)

{

while(n!=0)

{

rem=n%10;

if(rem%2==0)

evenprod=evenprod\*rem;

else

oddprod=oddprod\*rem;

n=n/10;

}

res = evenprod + oddprod;

System.out.println (res);

}

}

}

**75.Sum of the Digits**

In a lucky draw everybody got one coupon with some code. They need to sum the digits in the code and send SMS to the given number. Write a program to find the sum of digits in a number.

Include a function named **sumDigits**that accepts an integer argument and returns an integer that corresponds to the sum of the digits. The function returns -1 if the input is less than zero or if the roll number is greater than 32767.

If the function returns -1, print “Invalid Input”.

**Input and Output Format:**

The input consists of an integer.

The output consists of an integer that corresponds to the sum of the digits in the number.

**Sample Input 1:**

3487

**Sample Ouput 1:**

22

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, rem,flag=0, res=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

flag=1;

}

if(n>32767)

{

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

flag=1;

}

if(flag==0)

{

while(n!=0)

{

rem=n%10;

res=res+rem;

n=n/10;

}

System.out.print(res);

}

}

}

**76.Sum of squares of prime numbers**

Given an integer n, write a program to find the sum of squares of prime numbers upto and including n.

Include a function named **sumSquarePrime**that accepts an integer argument and returns an integer that corresponds to result. The function returns -1 if the input is less than zero or if the number is greater than 32767.

If the function returns -1, print “Invalid Input”.

Please note that 1 is neither prime nor composite.

**Input and Output Format:**

The input consists of an integer.

The output consists of an integer that corresponds to the sum of the squares of prime numbers.

**Sample Input 1:**

10

**Sample Ouput 1:**

87

**Sample Input 2:**

-8

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

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

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

flag=1;

}

if(n>32767)

{

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

flag=1;

}

if(flag==0)

{

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

{

count=0;

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

{

if(i%j==0)

count++;

}

if(count==2)

sum=sum+(i\*i);

}

System.out.println (sum);

}

}

}

**77.3/5 Number**

Write a program to find whether the given number is a 3/5 Number.

A number is a 3/5 Number if the product of the digits in the number is divisible by 3 or 5.

Include a function named **divisibleByThreeFive** that accepts an integer argument and returns an integer. The function returns

1. 1 if it is a 3/5 Number
2. 0 if it is not a 3/5 Number
3. -1 if it is a negative number

**Input and Output Format:**

Input consists of a single integer.

Output consists of a string.

Refer sample output for formatting specifications.

**Sample Input 1:**

251

**Sample Output 1:**

yes

**Sample Input 2:**

241

**Sample Output 2:**

no

**Sample Input 3:**

-9

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, flag=0,rem, prod=1;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

flag=1;

}

if(n>32767)

{

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

flag=1;

}

if(flag==0)

{

while(n!=0)

{

rem=n%10;

prod=prod\*rem;

n=n/10;

}

if(prod%3==0 || prod%5==0)

System.out.println ("yes");

else

System.out.println ("no");

}

}

}

**78.aboveAverageMarks**

Read the question carefully and follow the input and output format.  
  
Given an input array that represents the marks of students, find out the marks which are greater than or equal to average mark of all students.  
  
**Input and Output Format:**  
First line of input consists of n, the number of elements in the input array.  
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 numbers available in the input array and terminate the program.  
  
Include a function named aboveAverageMarks(int array[], int size) whose return type is void.  
The output array is stored in a global variable named above\_average.  
  
**Sample Input 1:**  
5  
10  
20  
30  
40  
50  
**Sample Output 1:**  
30  
40  
50  
  
**Sample Input 2:**  
4  
-3  
**Sample Output 2:**  
Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i, flag=0,sum=0,avg=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[20];

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++)

sum=sum+a[i];

avg=sum/n;

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

{

if(a[i]>=avg)

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

}

}

}

}

}

**79.changeNumber**

Read the question carefully and follow the input and output format.  
  
Tom needs to generate a new number from the given input with the following conditions.Consider Input is always a 3 digit number.  
  
conditions:  
(i) Middle digit comes first.  
(ii) Last digit should come in middle  
(iii) First digit should come as a last digit

Business rule:  
1. Print "Invalid input" if input is negative number.  
2. Print "Not a 3 digit number" if the given number is not a 3 digit number.  
158

581  
Include a function named changeNumber(int number) that returns an integer.  
  
**Input and Output Format:**  
Input consists of an integer.  
Refer business rules and sample output for output format.  
  
**Sample Input 1:**  
123  
**Sample Output 1:**  
231  
  
  
**Sample Input 2:**  
1234  
**Sample Output 2:**  
Not a 3 digit number

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, flag=0,rem, d1,d2,d3,i=1,temp=1,res=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

flag=1;

}

else if(n<100 || n>999)

{

System.out.println ("Not a 3 digit number");

flag=1;

}

if(flag==0)

{

do

{

rem=n%10;

d3=rem;

n=n/10;

rem=n%10;

d2=rem;

n=n/10;

rem=n%10;

d1=rem;

}while(n<0);

do

{

res=res+(d1\*i);

i=i\*10;

res=res+(d3\*i);

i=i\*10;

res=res+(d2\*i);

}while(n<0);

System.out.println (res);

}

}

}

**80.differentElements**

Read the question carefully and follow the input and output format.  
  
Given two input arrays find out the elements which are not common.  
  
**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, which contains the elements that are not common between the first and second array.  
  
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 differentElements(int set1[], int set2[], int size) whose return type is void.  
The output array is stored in a global variable named not\_common.  
  
**Sample Input 1:**  
5  
1 2 3 4 5  
5 6 4 8 7  
  
**Sample Output 1:**  
1  
2  
3  
6  
8  
7  
  
**Sample Input 2:**  
5  
1 4 8 9 4  
-8  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,f=0,k=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[n];

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

                                    {

                                                a[i] = in.nextInt();

                                                if(a[i] < 0)

                                                {

                                                            f=1;

                                                            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)

                                                {

                                                            f=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    if(f!=1)

                                    {

int not\_common[]=new int[20];

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)

{

not\_common[k]=a[i];

k++;

}

}

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

{

int flag1=0;

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

{

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

flag1=1;

}

if(flag1==0)

{

not\_common[k]=b[i];

k++;

}

}

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

System.out.println(not\_common[i]);

}

}

}

}

**81.registerAccountNumbers**

Read the question carefully and follow the input and output format.  
  
Given an array in which the elements are in xxxyy format, where first xxx digits represent the Branch code and the yy represents the account  
id.  Find out the No of accounts in the given branch code  
  
Input and Output Format :  
The first input n corresponds to the size of the array, the next n lines correspond to the elements of the array and the last line of the input corresponds to the branch code.  
Output corresponds to the number of accounts in the given branch code  
If the given branch code is not available, print 0.  
  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program  
2) Print "Invalid account Number" when there is any negative number available in the input array and terminate the program  
3) Print "Invalid branch code" when branch code is negative number and terminate the program  
  
Include a function named registerAccountNumbers (int size, int account\_numbers[], int branch\_code) that returns the no of accounts  
  
**Sample Input 1 :**  
6  
12345  
12370  
12324  
13355  
13333  
14575  
123  
  
**Sample Output 1 :**  
3  
  
**Sample Input 2 :**  
-6  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i ,bcode=0,count=0,temp=0;

// int 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 account number");

                                                            System.exit(0);

                                                }

                                    }

                                   //if(flag!=1)

                                    //{

                                                int code=in.nextInt();

                                                if(code<0)

                                                {

                                                            System.out.print("Invalid branch code");

                                                            System.exit(0);

                                                }

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

{

                                                            temp=a [i];

                                                            bcode=temp/100;

                                                            if(bcode==code)

                                                            count++;

                                                }

System.out.println(count);

//}

                        }

            }

}

**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]);

}

}

}

**83.findLargest**

Read the question carefully and follow the input and output format.  
  
Write a program to find the largest of the 3 given numbers.  
  
**Input and Output Format :**  
Input consists of 3 integers. Output consist of an integer that is the maximum.  
  
Print "Number too large" when any of given input numbers is greater than 32767 .  
Print "Number too small" when given input is a negative number.  
  
Include a function named findLargest(int num1, int num2, int num3) whose return type is an integer, which is the largest.  
  
**Sample Input 1:**  
2  
3  
4  
  
**Sample Output 1:**  
4  
  
**Sample Input 2:**  
98974  
  
**Sample Output 2:**  
Number too large  
  
**Sample Input 3:**  
-32767  
  
**Sample Output 3:**  
Number too small

**84.sumThreeLargest**

Read the question carefully and follow the input and output format.  
  
Write a program to find the sum of first ,second and third largest element in the given array.  
  
**Input and Output Format:**  
First line of input consists of n, the number of elements. Next n lines correspond to the array elements . Output consists of an Integer, the sum.  
  
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.  
  
**Sample Input 1:**  
8  
1  
2  
2  
3  
4  
5  
5  
4  
  
**Sample Output 1:**  
12  
  
**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,temp=0, sum=0;

//int 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[20];

                                    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++)

{

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

{

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

{

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

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

                                                                                    n--;

                                                                        }

                                                                        else

                                                                                    j++;

                                                            }

                                                }

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

{

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

{

                                                                        temp = a[i];

                                                                        a[i] = a[j];

                                                                        a[j] = temp;

                                                            }

                                                }

                                                sum=a[0]+a[1]+a[2];

                                                System.out.println(sum);

                                    //}

                        }

            }

}

**85.highestProfitYear**

Read the question carefully and follow the input and output format.  
  
An array holds information of a company profit margin and year. Find out the year in which highest revenue was earned. Assume the first index of array indicates year and the next index indicates the amount of money earned by the company and so on.  
  
**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, which is the sum  
  
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.  
  
  
**Sample Input 1:**  
6  
2012  
10000  
2011  
5000  
2009  
4000  
  
**Sample Output 1:**  
2012  
  
**Sample Input 2:**  
8  
2015  
89745  
-2015  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i ,max=0,highest=0;

//int 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 account number");

                                                            System.exit(0);

                                                }

                                    }

                                   // if(flag!=1)

                                    //{

                                                for(i=1;i<n;i=i+2)

{

                                                            if(a[i]>max)

{

                                                                        max=a[i];

                                                                        highest=a[i-1];

                                                            }

                                                }

                                                System.out.print(highest);

                                    //}

                        }

            }

}

**86.countNoOfConnections**

Read the question carefully and follow the input and output format.  
  
Given an input array, the elements are of format XYYY . where X represents the connection type. YYY represents the connection id.  
  
If X is 2 -> means 2G connection  
If X is 3 -> means 3G connection  
If X is 4 -> means 4G connection  
You need to find the number of type of connections.  
  
Note:  
If a particular connection type (starting with 2 or 3 or 4) is not available represent with zero in the corresponding position.  
  
  
**Business Rules :**  
1) Print "Invalid array size" when size of the array is a negative number and terminate the program.  
2) Print "Invalid connection" when there is any negative number available in the input array and terminate the program  
  
**Input and Output Format :**  
First line of input corresponds to n, next n lines corresponds to the elements of the array  
Output consists of the the number of type of connections. [1st line of the output corresponds to the number of 2G connections, 2nd line corresponds to the number of 3G connections and 3rd line corresponds to the number of 4G connections]  
  
**Sample Input 1 :**  
5  
2333  
3101  
2102  
4567  
3123  
  
**Sample Output 1:**  
2  
2  
1  
  
**Sample Input 2 :**  
2  
-2234  
  
  
**Sample Output 2:**  
Invalid connection  
  
import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, x,temp,two=0,three=0,four=0,other;

//int 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[20];

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

                                    {

                                                a[i] = in.nextInt();

                                                 if(a[i] < 0)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    //if(flag!=1)

                                    //{

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

{

                                                            temp=a[i];

                                                            x=temp/1000;

                                                            if(x==2)

                                                            two++;

                                                            else if(x==3)

                                                            three++;

                                                            else if(x==4)

                                                            four++;

                                                            else

                                                            other=0;

                                                }

                                                int out[]=new int[3];

                                                out[0]=two;

                                                out[1]=three;

                                                out[2]=four;

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

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

//}

                        }

            }

}

**87.Sum of positive numbers in Array**

Write a program to find the sum of positive numbers in an array.

 Include a function named**addPositives** 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 the sum of the positive numbers in the array. If the size of 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.Assume that the maximum size of the array is 20.

**Sample Input 1:**

5

3

5

-2

6

-6

**Sample Output 1:**

14

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,sum=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n < 0)

                        {

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

                                    System.exit(0);

                        }

                        else

                        {

                                    int a[]=new int[n];

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

                                    {

                                                a[i] = in.nextInt();

                                                if(a [i]>0)

                                                sum=sum+a [i];

                                    }

                                    System.out.print(sum);

                        }

            }

}

**88.Find Index**

Write a program to find the index of a particular number in a given input array.

 Include a function named **findIndex** that accepts 3 arguments and returns an int. The first argument is the input array, the second argument is an int that corresponds to the size of the array and the third argument is the element to be searched for. The function returns the corresponding index if the search element is present in the array and returns -1 if the search element is not present in 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+2 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. The last integer corresponds to the element whose count needs to be found.

Output consists of an integer that corresponds to the index of the search element if it is present.

Else, print 'not found'. Refer sample output for formatting specifications. Assume that the maximum number of elements in the array is 20 and that all elements in the array are unique.

**Sample Input 1:**

8

2

1

3

8

6

12

10

19

8

**Sample Output 1:**

3

**Sample Input 2:**

8

2

1

3

8

6

12

10

19

80

**Sample Output 2:**

not found

**Sample Input 3:**

-5

**Sample Output 3:**

Invalid Input

**Sample Input 4:**

5

23

2

-200

**Sample Output 4:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, se,f=0,index=0;

//int 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[20];

                                    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)

                                    //{

                                                se=in.nextInt();

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

{

                                                            if(a [i]==se)

{

                                                                        f=1;

                                                                        index = i;

                                                                        break;

                                                            }

                                                }

                                                if(f==1)

                                                System.out.print(index);

                                                else

                                                System.out.print("not found");

                                    //}

                        }

            }

}

**89.Store Consequtives**

Write a program to obtain a new array that contains the consequtive values of the given input array. The output array is named as output1.

 Include a function named **storeConsequtives** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size of the array .  The output array is stored in a global variable named output1.

 If the size of the array is negative or if any of the elements in the array are 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 array.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

4

2

5

1

4

**Sample Output 1:**

3

6

2

5

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, se,f=0,index=0;

//int 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[20];

                                    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++)

{

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

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

                                                }

                                    //}

                        }

            }

}

**90.findGrade**

Read the question carefully and follow the input and output format.  
  
In a school examination the result of students are published in the form of an array where first index is the student id and the second index is the total marks in mathematics third index is student id and fourth index is total marks in mathematics and so on... Write a method that assigns the student id as the key and the grade in mathematics as the value to the output array based on the following conditions:  
  
If(Marks >=90 ) : 1  
If(Marks >=80 and <90 ) : 2  
If(Marks >=70 and <80) : 3  
If(Marks <70 ) : 0  
  
Hint: Array size is always is 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 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.  
  
Include a function named findGrade(int array[], int size) whose return type is void.  
The output array is stored in a global variable named grade.  
  
**Sample Input 1:**  
10  
1  
65  
2  
74  
3  
86  
4  
95  
6  
69

**Sample Output 1:**  
1  
0  
2  
3  
3  
2  
4  
1  
6  
0  
  
**Sample Input 2:**  
4  
-3

**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i;

//int 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[20];

                                    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)

                                   // {

                                                int grade[]=new int[20];

                                                for(i=1;i<n;i=i+2)

{

                                                            if(a[i]>=90)

{

                                                                        grade[i-1]=a[i-1];

                                                                        grade[i]=1;

                                                            }

                                                            else if(a[i]>=80 && a[i]<90)

{

                                                                        grade[i-1]=a[i-1];

                                                                        grade[i]=2;

                                                            }

                                                            else if(a[i]>=70 && a[i]<80)

{

                                                                        grade[i-1]=a[i-1];

                                                                        grade[i]=3;

                                                            }

                                                            else

{

                                                                        grade[i-1]=a[i-1];

                                                                        grade[i]=0;

                                                            }

                                                }

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

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

                                    //}

                        }

            }

}

**91.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

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, sumodd=0,sumeven=0;

// int 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)

                                    //{

                                                double avg,av;

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

                                                {

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

                                                            sumeven=sumeven+a[i];

                                                            else

                                                            sumodd=sumodd+a[i];

                                                }

                                                avg = (sumeven+sumodd);

                                                av=avg/2;

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

                                    //}

                        }

            }

}

**92.convertToBinary**

Read the question carefully and follow the input and output format.  
  
Kate is a military officer. He needs to send top-secret code to the other soldiers in a different place. He need to encrypt it. We need to write a function to convert the given decimal number to the corresponding binary number.  
  
Input consist of single integer. Output consists of binary number.  
  
1) Print "Number too small" when input is negative.  
2) Print "Number too large" when input value is greater than 100.  
  
Include a function named convertToBinary(int num) whose return type is void.  
  
**Sample Input 1:**  
12  
  
**Sample Output 1:**  
1100  
  
**Sample Input 2:**  
101  
  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n,i=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

                                    System.exit(0);

                        }

                        else if(n>100)

                        {

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

                                    System.exit(0);

                        }

                        else

                        {

                                    int a[]=new int[20];

                                    while(n>0)

                                    {

                                                a[i]=n%2;

                                                i++;

                                                n=n/2;

                                    }

                                    for(int j=i-1;j>=0;j--)

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

}

            }

}

**93.reverseNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to find  the reverse of a given input integer  
  
**Input and Output Format :**  
Input consists of an integer, n.  Output consist of the reverse of the number n.  
  
Print "Number too large" when the given input number is greater than 32767  
Print "Number too small" when the given input numbers is a negative number.  
  
**Sample Input 1:**  
1234  
  
**Sample Output 1:**  
4321  
  
**Sample Input 2:**  
326357  
  
**Sample Output 2:**  
Number too large

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rev=0 ,rem;

//int flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

                        {

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

System.exit(0);

                                    //flag=1;

                        }

                        if(n>32767)

                        {

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

System.exit(0);

                                    //flag=1;

                        }

                        //if(flag==0)

                        //{

                                    while (n != 0)

                                    {

                                                rem =n%10;

                                                rev = (rev \*10)+rem;

                                                n=n/10;

                                    }

                                     System.out.println (rev);

//}

            }

}

**94.Sum of Prime Cubes**

Given an input integer n, write a program to find the sum of the cubes of the prime numbers upto n. (including n)

Please note that 1 is neither prime nor composite.

The function returns -1 if the input is a negative number or if it is greater than 3000.

If the function returns -1, print Invalid Input.

**Input and Output Format:**

Input consists of a single integer.

Output consists of a single integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

5

**Sample Output 1:**

160

**Sample Input 2:**

-241

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

50000

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int i,n,j, k,sum=0,flag=0;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0 || n>3000)

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

                        else

                        {

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

                                    {

                                                int flag=0;

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

                                                {

                                                            if(i%j==0)

                                                            flag=1;

                                                }

                                                if(flag==0)

                                                {

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

                                                }

                                    }

                                    System.out.println(sum);

                        }

            }

}

**95.Negative Elements**

Write a program to remove all the negative elements of the input array , sort the positive elements and stores them in an output array .

Include a function named **eliminateNegatives** that accepts 2 arguments and its return type is void. The first argument is the input array and the second argument is an int that corresponds to the size of the array. The output array is stored in a global variable named output1 and the size of the output array is stored in a global variable named output2.

If the size of 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 array.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

8

1

6

3

5

-6

8

-15

9

**Sample Output 1:**

1

3

5  
  
6

8

9

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i,j,k=0, temp;

                        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();

                                    int output1[]=new int[n];

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

                                    {

                                                if(a [i]>0)

                                                {

                                                            output1[k]=a [i];

                                                            k++;

                                                }

                                    }

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

                                    {

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

                                                {

                                                            if(output1[i]>output1[j])

                                                            {

                                                                        temp = output1[i];

                                                                        output1[i]=output1[j];

                                                                        output1[j]=temp;

                                                            }

                                                }

                                    }

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

                                    System.out.println(output1[i]);

                        }

            }

}

**96.Savings Calculation**

Jim got salary for this month and he spends 50% of his salary for food and 20% of his salary for travel. If the number of days he worked is 31 he gets a bonus of Rs.500. Write a program to find how much he can save in his pocket after spending all these?

Include a function named **calculateSavings** that accepts 2 integer arguments and returns a float. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days Jim has worked. The function returns a float that corresponds to the amount that Jim could save.

Print Invalid Input and terminate the program in the following cases:

1.      Basic salary is greater than 9000

2.      Number of working days is greater than 31

3.      Basic salary is negative

4.      Number of working days is negative

**Input and Output Format:**

Input consists of 2 integers. The first integer corresponds to Jim's basic salary and the second integer corresponds to the number of days he has worked.

Output consists of a single float that corresponds to Jim's savings. Jim's savings is displayed correct to 2 decimal places.

**Sample Input 1:**

7000

30

**Sample Output 1:**

2100 .00

**Sample Input 2:**

50000

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int basic=0,days=0;

float spent,savings;

                        Scanner in=new Scanner(System.in);

                        basic=in.nextInt();

                        days=in.nextInt();

                        if(basic>9000 || days>31 || basic<0 || days<0)

{

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

                                    System.exit(0);

                        }

                        spent=(basic\*70)/100;

                        if(days==31)

                        savings = basic+500-spent;

                        else

                        savings = basic-spent;

                        System.out.printf(“%.2f”,savings);

}

}

**97.studentMarks**

Read the question carefully and follow the input and output format.  
  
The Given input is of the format XXXYY , where XXX is Id , YY is marks. Write a code to display the id and marks separately as given in the output formats. [Refer sample input and output]  
  
**Input and Output Format :**  
Input consists of a number. Refer sample output for output format.  
  
Print "Number too large" when any of given input numbers is greater than 32767 .  
Print "Number too small" when given input is a negative number.  
  
Include a function named studentMarks(int number) whose return type is void.  
  
**Sample Input 1:**  
12345  
  
**Sample Output 1:**  
123  
45  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid input

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n=0,result;

                        Scanner in=new Scanner(System.in);

                        n = in.nextInt();

                        if(n<0)

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

                        else if(n>32767)

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

                        else

                        {

int id=0,mark=0,rem=0,i=1,flag=1,j=1;

                                    while(n!=0)

{

                                                if(flag<=2)

{

                                                            rem=n%10;

                                                            mark=mark+(rem\*i);

                                                            i=i\*10;

                                                            n=n/10;

                                                            flag++;

                                                }

                                                else

{

                                                            rem=n%10;

                                                            id=id+(rem\*j);

                                                            j=j\*10;

                                                            n=n/10;

                                                            flag++;

                                                }

                                    }

                                    System.out.println(id);

System.out.println(mark);

}

}

}

**98.maximumSum**

Read the question carefully and follow the input and output format.  
  
Given an Integer array, find out sum of Even and odd Numbers individually and find the maximum.  
  
**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 maximum of odd and even sum.  
  
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.  
  
**Sample Input 1:**  
5  
12  
13  
14  
15  
16  
  
**Sample Output 1:**  
42  
  
**Sample Input 2:**  
-13  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, i, oddsum=0,evensum=0,max=0;

//int 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]%2==0)

                                                            evensum = evensum + a[i];

                                                            else

                                                            oddsum = oddsum + a[i];

                                                            if(evensum>oddsum)

                                                            max = evensum;

                                                            else

                                                            max=oddsum;

                                                }

                                                System.out.print(max);

                                    //}

                        }

            }

}

**99.newNumber**

Read the question carefully and follow the input and output format.  
  
Write a program to find the difference between consecutive digits in the given input integer and display it.  
  
Input and Output Format:  
Input consists of an integer and output the difference between the consecutive digits.  
  
Print "Number too small" if the number is less than 0  
Print "Number too large" if the number is greater than 32767  
  
  
**Sample Input 1:**  
1325  
**Sample Output 1:**  
213  
  
**Sample Input 2:**  
-13  
**Sample Output 2:**  
Number too small

 import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int n, rem,i=1,rem1=0,diff=0,digit=0;

                        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);

                        }

                        else

                        {

                                    while(n>10)

{

                                                rem=n%10;

                                                n=n/10;

                                                rem1=n%10;

                                                if(rem>rem1)

                                                diff=rem-rem1;

                                                else

                                                diff=rem1-rem;

                                                digit=digit+(diff\*i);

                                                i=i\*10;

                                    }

                                    System.out.println(digit);

                        }

            }

}

**100.nonWorkingDoctors**

Read the question carefully and follow the input and output format.  
  
A doctor survey results information is stored in 2 arrays. First array represents all doctors ids (working and non -working both). Second array represents only working doctor's id . Please find the doctor ids who are not working .  
  
**Input and Output Format :**  
First line of input corresponds to n1, the size of first array and next n1 lines correspond to the elements of the first array. The next line corresponds to n2, the size of second array and next n2 lines correspond to the elements of the second array  
Output is the id's of doctor who are not working  
  
Print "Invalid array size" when size of the array is a negative number and terminate the program  
Print "Invalid id" when there is any negative numbers available in the input array and terminate the program.  
  
Include a function named nonWorkingDoctors(int total[],int working[],int n,int m) whose return type is void.  
  
**Sample Input 1:**  
7  
7  
2  
3  
4  
5  
6  
1  
3  
3  
4  
5

**Sample Output 1:**  
7  
2  
6  
1  
  
**Sample Input 2:**  
7  
7  
2  
3  
4  
5  
6  
-1

**Sample Output 2:**  
Invalid id

import java.util.Scanner;

public class Main

{

            public static void main(String[] args)

            {

                        int m,n, i,j;

//int flag=0;

                        Scanner in=new Scanner(System.in);

                        m=in.nextInt();

                        if(m < 0)

                        {

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

                                    System.exit(0);

                        }

                        else

                        {

                                    int a[]=new int[m];

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

                                    {

                                                a[i] = in.nextInt();

                                                if(a[i] < 0)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    n=in.nextInt();

                                    if(n<0)

                                    {

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

                                                System.exit(0);

                                    }

                                    int b[]=new int[n];

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

                                    {

                                                b[i] = in.nextInt();

                                                if(b[i] < 0)

                                                {

                                                            //flag=1;

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

                                                            System.exit(0);

                                                }

                                    }

                                    //if(flag!=1)

                                    //{

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

                                                {

                                                            flag=0;

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

                                                            {

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

                                                                         flag=1;

                                                            }

                                                            if(flag==0)

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

                                                }

                                    //}

                        }

            }

}

**101.firstMiddleSame**

Read the question carefully and follow the input and output format.  
  
Write a program to check if the first and middle element in an array is the same, if so display “100” in output or else display the first element of the array.  
  
Note: Array size is always odd.  
  
**Input and Output Format :**  
The first line of the input consists of an integer, n that corresponds to the number of elements in the array.  
The next 'n' lines correspond to the elements in the array.  
  
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.  
3) Print "Size must be odd" when the size of the array is even.  
  
**Sample Input 1:**  
7  
8  
4  
5  
8  
3  
2  
1  
  
**Sample Output 1:**  
100  
  
**Sample Input 2:**  
10  
4  
5  
6  
-8  
  
**Sample Output 2:**  
Invalid input  
  
**Sample Input 3:**  
3  
4  
8  
6  
  
**Sample Output 3:**  
4

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i;

//int flag=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n < 0)

{

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

System.exit(0);

}

else if(n%2==0)

{

System.out.print("size must be odd");

}

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)

//{

int mid = n / 2;

if(a[0] == a[mid])

System.out.println("100");

else

System.out.println(a[0]);

//}

}

}

}

**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);

}

}

}

**103.Array Sorting**

Write a program to sort the first half of the input array elements in ascending order and the second half of the input array elements in descending order.

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 n integers that correspond to the elements in the sorted array.

Assume that the maximum size of the array is 20.

**Sample Input 1:**

7

1

9

8

4

6

4

5

**Sample Output 1:**

1

4

8

9

6

5

4

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-5

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i,j, temp=0;

// int 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)

//{

int mid;

if(n%2==0)

mid=n/2;

else

mid=n/2+1;

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

{

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

{

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

{

temp = a [i];

a [i] = a [j];

a [j] = temp;

}

}

}

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

{

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

{

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

{

temp = a [i];

a [i] = a [j];

a [j] = temp;

}

}

}

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

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

//}

}

}

}

**104.occurenceDigit**

Read the question carefully and follow the input and output format.  
  
Given a number n, count the occurences of a number,x in n.  
  
**Input and Output Format :**  
The first line of the input consists of an integer n, the second line consists of an integer, which is the digit whose occurence is to be calculated. Output is an integer that gives the count.  
  
Print "Number too small" if any of the 2 input numbers is less than 0 and terminate the program.  
Print "Number too large" if any of the 2 input numbers is greater than 32767 and terminate the program.  
  
**Sample Input 1:**  
1122  
1  
**Sample Output 1:**  
2  
  
  
**Sample Input 2:**  
-2134  
**Sample Output 2:**  
Number too small

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, digit;

// int flag=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0)

{

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

//flag=1;

System.exit(0);

}

if(n>32767)

{

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

//flag=1;

System.exit(0);

}

//if(flag==0)

//{

digit=in.nextInt();

if(digit<0)

{

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

System.exit(0);

}

else if(digit >32767)

{

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

System.exit(0);

}

else

{

int rem,count=0;

while(n!=0)

{

rem=n%10;

if(rem==digit)

count++;

n=n/10;

}

System.out.println(count);

}

//}

}

}

**105.consecutiveDifference**

Read the question carefully and follow the input and output format.  
  
Given input as integer array in which consecutive elements should have a difference of 4 or more than 4.  
  
If the above condition matches display “1” else “0”  
  
**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, which is the either 1 or 0  
  
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.  
  
**Sample Input 1:**  
6  
1  
5  
10  
6  
2  
7  
  
**Sample Output 1:**  
1  
  
**Sample Input 2:**  
-8  
  
**Sample Output 2:**  
Invalid array size

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i,j, res=0,diff=0,f=0;

// int 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-1;i++)

{

f=0;

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

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

else

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

if(diff<4)

{

f=1;

break;

}

}

if(f==1)

System.out.print("0");

else

System.out.print("1");

//}

}

}

}

**106.Change using Minimal Coins / Notes**

Ram needs to pay the school fees of his 6 year old kid. As he is busy with his work, he is not finding time to go to the school to make payment. His kid's school doesn't accept online payment. So he decided to send the fee amount through his kid. The available denominations of rupees or coins is 500, 100, 50, 10, 5 and 1. Can you write a program to find the minimal number of coins or notes to be given to his kid?

**Input and Output Format:**

Input consists of a single integer that corresponds to the fee amount to be paid.

Output consists of an integer that corresponds to the minimal number of coins or rupee notes.

Print Invalid Input if the input value is negative.

**Sample Input 1:**

682

**Sample Output 1:**

8

**Sample Input 2:**

**-**2345

**Sample Output 2:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int rupees,count=0, c1,c2,c3,c4,c5,c6;

Scanner in=new Scanner(System.in);

rupees = in.nextInt();

if(rupees<0)

{

count=-1;

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

System.exit(0);

}

else

{

c1=rupees/500;

rupees=rupees-(500\*c1);

c2=rupees/100;

rupees=rupees-(100\*c2);

c3=rupees/50;

rupees=rupees-(50\*c3);

c4=rupees/10;

rupees=rupees-(10\*c4);

c5=rupees/5;

rupees=rupees-(5\*c5);

c6=rupees/1;

rupees=rupees-(1\*c6);

count=c1+c2+c3+c4+c5+c6;

System.out.print(count);

}

}

}

**107.singleDoubleTripCount**

Read the question carefully and follow the input and output format.  
  
In a given input array find out number of single digits, double digits and triple digits.  
  
**Input and Output Format :**  
First line of input consists of an integer, the size of the array. Next line correspond to the elements of the array. Output consist of an integer array.  
  
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.  
  
  
The output array is stored in a global variable named count. The size of the output array is 6.  
**Sample Input 1:**  
5  
1  
2  
23  
34  
456  
  
**Sample Output 1:**  
1  
2  
2  
2  
3  
1  
  
**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, c1=0,c2=0,c3=0;

// int 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)

//{

int count[]=new int[6];

count[0]=1;

count[2]=2;

count[4]=3;

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

{

if(a[i]>=0 && a[i]<=9)

count[1]=++c1;

else if(a[i]>=10 && a[i]<=99)

count[3]=++c2;

else if(a[i]>=100 && a[i]<=999)

count[5]=++c3;

else

;

}

for(i=0;i<6;i++)

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

//}

}

}

}

**108.Sum of multiples**

Given 2 integers n and m, write a program to find the sum of all numbers divisible by m upto and including n (i.e from 1 to n).

The first argument corresponds to n and the second argument corresponds to m. The function returns -1 if the any of the two inputs is less than 0.

If the function returns -1, print “Invalid Input”.

**Input and Output Format:**

The input consists of 2 integers. The 1st integer corresponds to n and the 2nd integer corresponds to m.

The output consists of an integer that corresponds to the sum of the multiples.

**Sample Input 1:**

11

5

**Sample Ouput 1:**

15

**Sample Input 2:**

-8  
2

**Sample Output 2:**

Invalid Input

import java.util.\*;

public class Main

{

public static void main(String[] args)

{

int n=0,m=0,sum=0,i;

Scanner in=new Scanner(System.in);

n = in.nextInt();

m = in.nextInt();

if(n<0 || m<0)

{

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

System.exit(0);

}

else

{

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

{

if(i%m==0)

sum = sum + i;

}

System.out.print(sum);

}

}

}

**109.Search Element**

Write a program to find whether a particular number appears in a given input array.

Include a function named **isElementPresent** that accepts 3 arguments and returns an int. The first argument is the input array, the second argument is an int that corresponds to the size of the array and the third argument is the element to be searched for. The function returns 1 if the search element is present in the array and returns 0 if the search element is not present in 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+2 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. The last integer corresponds to the element whose count needs to be found.

Output consists of a string that is either 'yes' or 'no'.

Refer sample output for formatting specifications.

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

**Sample Input 1:**

8

2

1

3

8

6

8

10

8

8

**Sample Output 1:**

yes

**Sample Input 2:**

-5

**Sample Output 2:**

Invalid Input

**Sample Input 3:**

5

23

2

-200

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class Main

{

public static void main(String[] args)

{

int n, i, se,f=0;

// int 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)

//{

se=in.nextInt();

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

{

if(a[i]==se)

{

f=1;

break;

}

}

if(f==1)

System.out.print("yes");

else

System.out.print("no");

//}

}

}

}

**110.Binary Conversion**

Write a program to convert a given input decimal number to binary.

Include a function named **convertToBinary** that accepts an integer argument and its return type is void. Print the binary representation of the number in this function.

If the input value is negative or greater than 100, print Invalid Input and terminate the program.

**Input and Output Format:**

Input consists of a single integer.

Output consists of the binary representation of the given integer.

Refer sample output for formatting specifications.

**Sample Input 1:**

12

**Sample Output 1:**

1100

**Sample Input 2:**

64

**Sample Output 2:**

1000000

**Sample Input 3:**

**-**2345

**Sample Output 3:**

Invalid Input

import java.util.Scanner;

public class HelloWorld

{

public static void main(String[] args)

{

int n,i=0;

Scanner in=new Scanner(System.in);

n = in.nextInt();

if(n<0 || n>100)

{

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

System.exit(0);

}

else

{

int a[]=new int[20];

while(n>0)

{

a[i]=n%2;

i++;

n=n/2;

}

for(int j=i-1;j>=0;j--)

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

}

}

}

**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);

}

}