

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

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

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

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

Include a function named newNumber(int number) that returns a integer

Sample Input 1:

1325

Sample Output 1:

213

Sample Input 2:

-13

Sample Output 2:

Number too small

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

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

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

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

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

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 Output 1:

12

Sample Input 2:

-8

Sample Output 2:

Invalid Input