Luxor University
Faculty of computers and information

Programming Fundamentals

Lab Sheet #4

Objectives:

- Learn how to think on paper before coding.
- Learn how to construct your program step by step in detail (Algorithm).
- Learn how to present your program algorithm in an efficient and organized way.
- Start you first C console programs.
- Learn how to solve problems using (variables & data types, input & output and operators).
- Understand the difference between sequential execution and transfer of controls.
- Learn how to solve problems using (conditions).

Problems:

- Arithmetic
- Shapes with Asterisks
- Vowel or Constant Char?
- Table of Squares and Cubes
- Triangle
- Car-Pool Savings Calculator
- Palindrome Tester
- Counting 7s
- Sum of range
- Char Again!

Arithmetic

Problem statement:

Write a program that inputs three different integers from the keyboard, and then prints the sum, the average, the product, the smallest and the largest of these numbers.

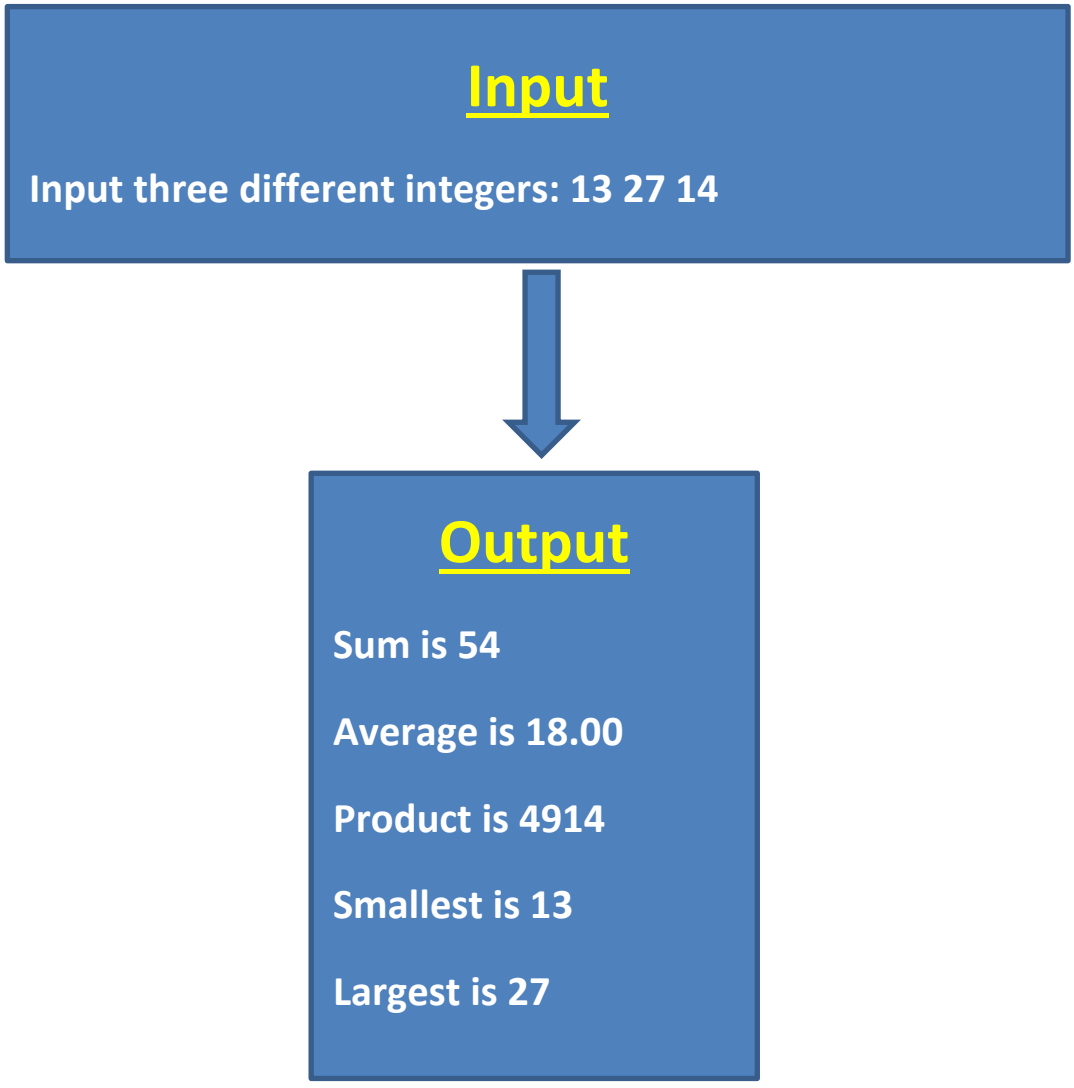
Input:

3 integers.

Output:

Sum, the average, the product, the smallest and the largest of these numbers. **Print only 2 numbers after the decimal point in average.**

Example 1:



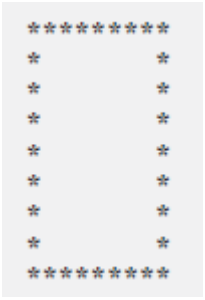
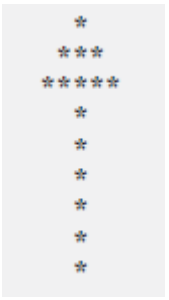
Shapes with Asterisks

Problem statement:

Write a program that prints the following shapes with asterisks.

Notes:

Write a program for each shape. Don't use loops.

Shape 1	
Shape 2	
Shape 3	
Shape 4	

Vowel or Constant Char?

Problem statement:

Write a C program to read a lower or upper case character (w) and determine if it is vowel or constant character.

Note:

Vowel characters are (a, e, i, o, and u) otherwise it is constant.

Input:

1 character (‘a’=<w<=’z’, ‘A’=<w<=’Z’).

Output:

If character is vowel print **VOWEL** otherwise print **CONSTANT**.

Example 1:

Example 2:

Table of Squares and Cubes

Problem statement:

Write a program that calculates the squares and cubes of the numbers from 0 to 5 and use **output formatting techniques** to print the following table of values:

```
number  square  cube
0       0       0
1       1       1
2       4       8
3       9       27
4       16      64
5       25      125
```

Note:

Don’t use loops

Triangle

Problem statement:

Write a C program to read the 3 sides of triangle. Using these sides determine if it is a non-degenerate triangle or not.

Input:

The 3 sides of triangle.

Output:

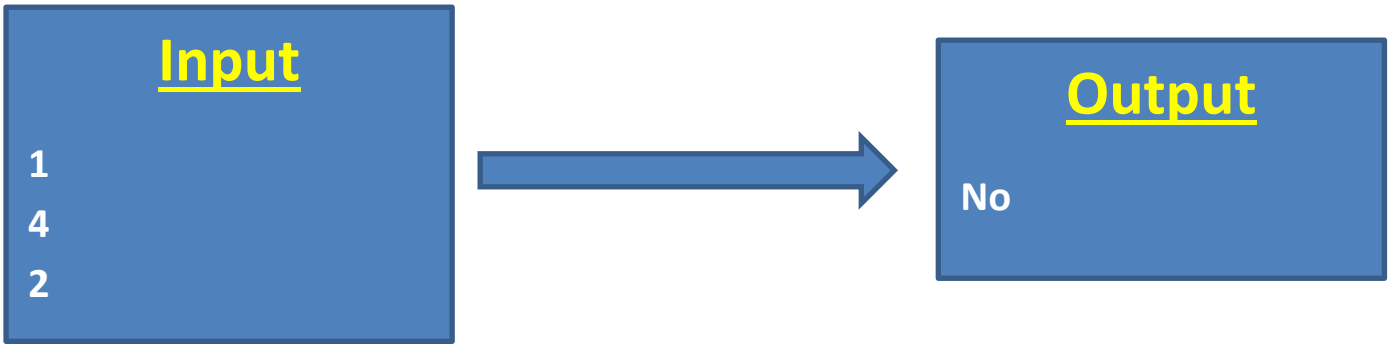
If is a non-degenerate triangle print **Yes** otherwise **No**.

Note:

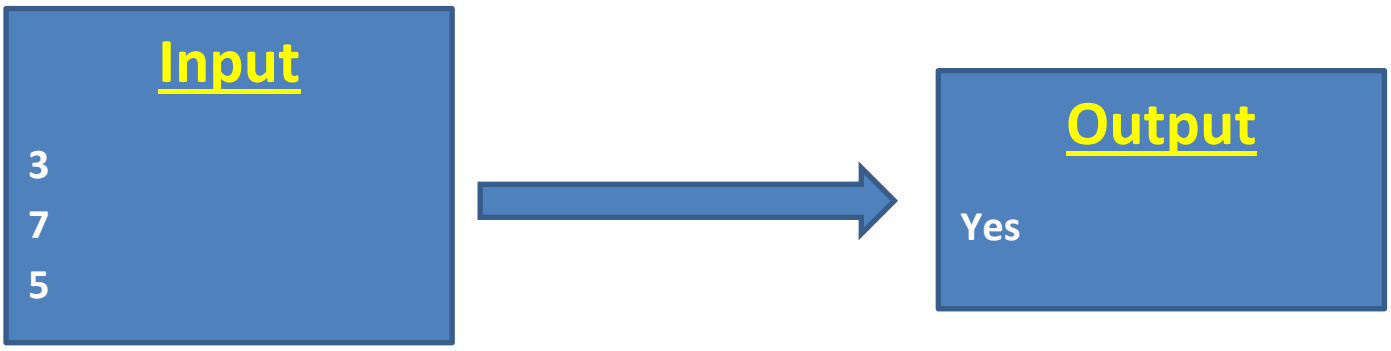
The non-degenerate triangle has this feature the summation of any 2 sides has to be greater than the third. For example let's say we have 3 sides (a, b, c) to be a non-degenerate triangle:

- a + b > c
- a + c > b
- c + b > a

Example 1:



Example 2:



Car-Pool Savings Calculator

Problem statement:

Create an application that calculates your daily driving cost, so that you can estimate how much money could be saved by carpooling, which also has other advantages such as reducing carbon emissions and reducing traffic congestion.

Input:

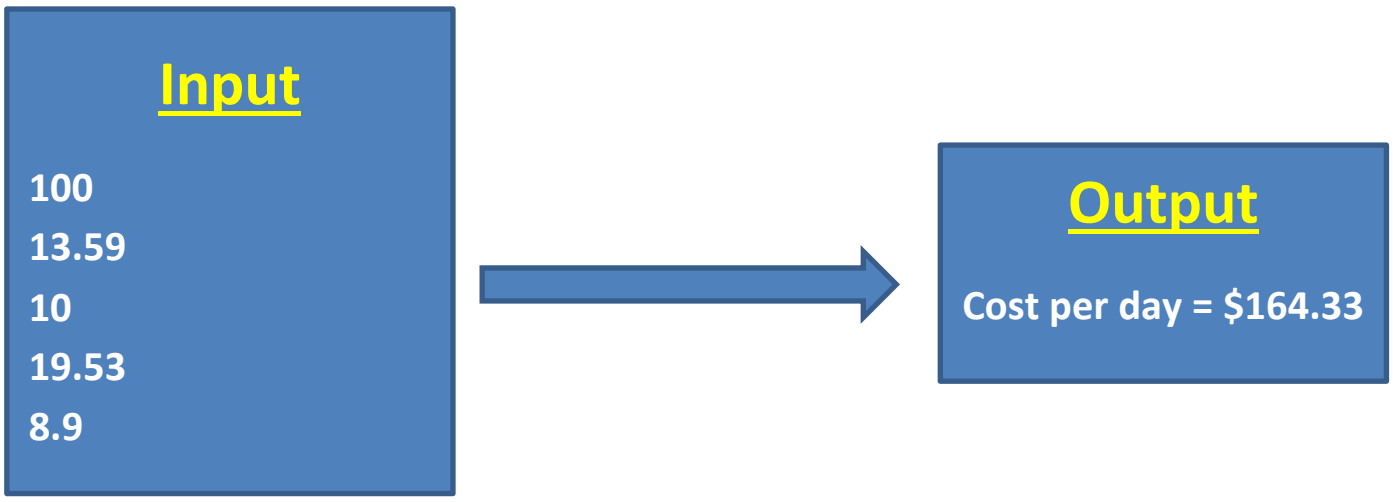
The application should input the following information:

- Total miles driven per day.
- Cost per gallon of gasoline.
- Average miles per gallon.
- Parking fees per day.
- Tolls per day.

Output:

Display the user's cost per day of driving to work. **Print only 2 numbers after the decimal point.**

Example 1:



Palindrome Tester

Problem statement:

A palindrome is a number or a text phrase that reads the same backward as forward. For example, each of the following five-digit integers is a palindrome: 12321, 55555, 45554 and 11611. Write a program that reads in a five-digit integer and determines whether or not it's a palindrome.

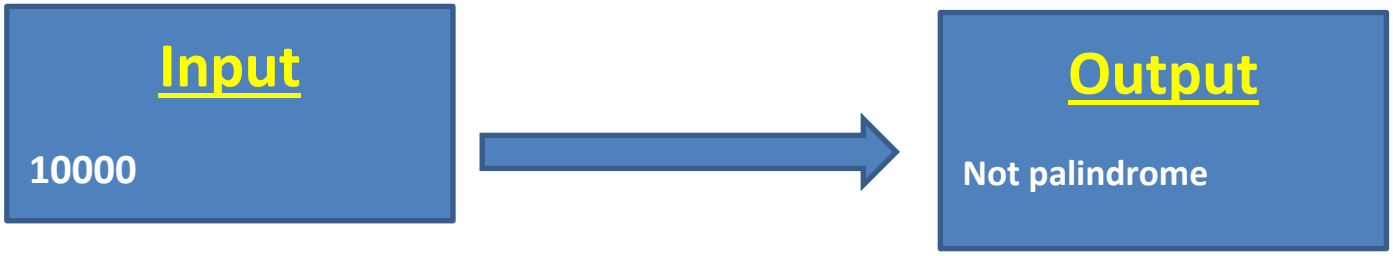
Input:

A five digits integer.

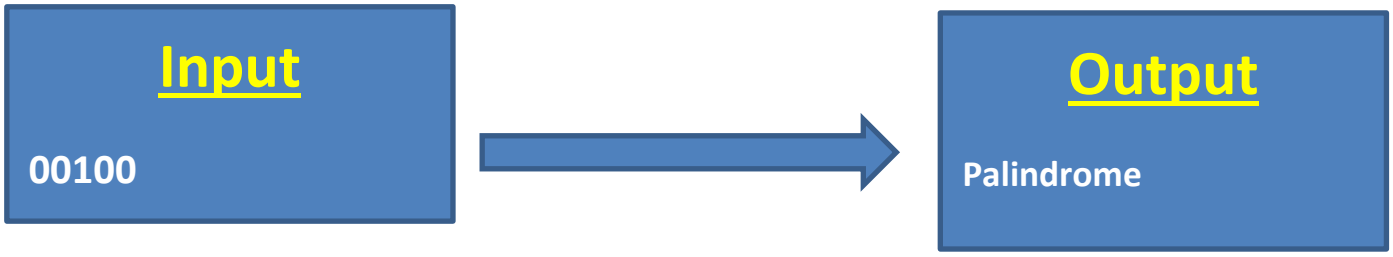
Output:

If number is palindrome print **Palindrome** otherwise print **Not palindrome**.

Example 1:



Example 2:



Counting 7s

Problem statement:

Write a program that reads an integer of 5 digits and determines and prints how many digits in the integer are 7s.

Input:

1 integer of 5 digits.

Output:

How many digits in the integer are 7s.

Example 1:

Example 2:

Sum of range

Problem statement:

Write a C program to compute the summation of range [L, R] **inclusive**.

Note:

Don't use loops.

Input:

The first number or range (L) and the last number of range (R). While (1=<L,R<=10^9, L<=R)

Output:

the summation of range [L, R] **inclusive**.

Example 1:

Explanation:

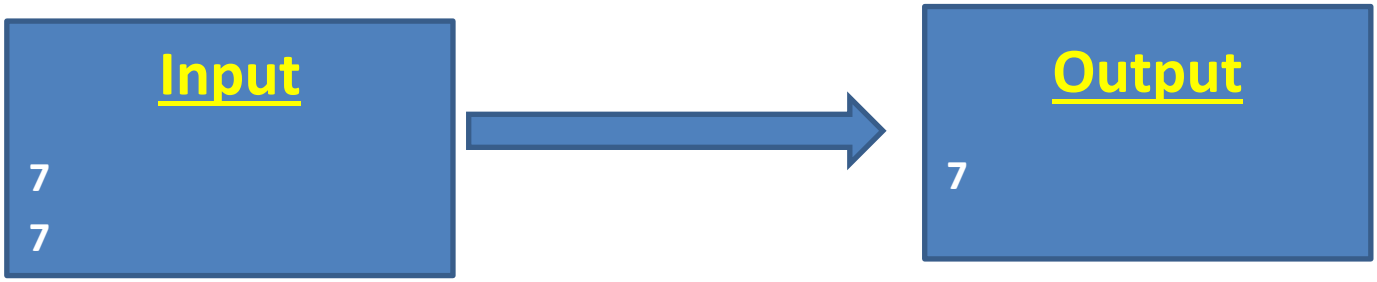
[1, 3] = 1 + 2 +3 =6.

Example 2:

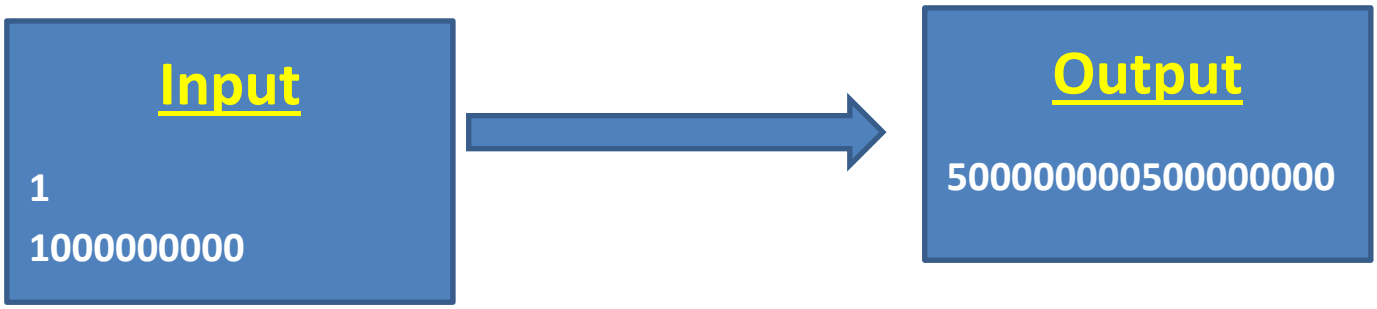
Explanation:

[4, 8] = 4 + 5 + 6 + 7 + 8 =6.

Example 3:



Example 4:



Example 5:

Char Again!

Problem statement:

Write a C program that prints the integer equivalents of some uppercase letters, lowercase letters, digits and special symbols. And if this character is a letter change his case (from upper to lower, from lower to upper).

Input:

A character (w).

Output:

Print the integer equivalent. And if this character is a letter change his case (from upper to lower, from lower to upper).

Example 1:

Example 2:

Example 3: