Assignment 4

Name:____

Choose one (1) of the three (3) programs. Make sure to write the flowchart.

Sum of N Consecutive Positive Integers

Define the function

Function Name:	`SumOfNRecursive()`
Parameter(s):	n: `int`
Return:	int

Function Name:	`SumOfNFormula()`
Parameter(s):	n: `int`
Return:	int

where both `SumOfNRecursive()` and `SumOfNFormula()` returns sum of n consective integers from 1 to n if n is positive; otherwise, they return zero (0). However, `SumOfNRecursive()` should derive the solution recursively; whereas, `SumOfNFormula()` uses an arithmetic formula to derive the solution.

In the main function,

1. prompt the user to enter two (2) integers.
2. display the outputs of the calls to `SumOfNRecursive()` with the user's inputs.
3. display the outputs of the calls to `SumOfNFormula()` with the user's inputs.

A possible output of the program is:

```
Enter two numbers:  -5 12

Outputs from recursive function
S(-5) = 0
S(12) = 78

Outputs from formula function
S(-5) = 0
S(12) = 78
```

Green text are inputs.

Positive Integer Perfect Squares

Define the functions

Function Name:	`squareRecursive()`
Parameter(s):	n: `int`
Return:	int

Function Name:	`square()`
Parameter(s):	n: `int`
Return:	int

where both `squareRecursive()` and `square()` returns the square of n; however, `squareRecursive()` finds it recursively and `square()` finds it with an arithmetic equation.

In the main function,

1. assign four (4) int variables random numbers between 1 and 99 inclusively.
2. display for each variable a statement that states if the calls to `square()` and `squareRecursive()` are identical.
3. display a statement that states that the function are identical for positive integers if the calls for each variable identical; otherwise, state that there are not identical.

Note: Include libraries ctime and cstdlib to use `srand()`, `rand()` and `time()` for generating random numbers.

A possible output of the program is:

```
For n = 5, both square() and squareRecursive() produced 25
For n = 23, both square() and squareRecursive() produced 529
For n = 17, both square() and squareRecursive() produced 289
For n = 56, both square() and squareRecursive() produced 3136
The functions square() and squareRecursive are identical for positive integers
```

Base N Notation

Define the functions

Function Name:	`BaseConvert()`
Parameter(s):	*value*: `int` *base*: `int`
Return:	nothing

Function Name:	`NumberGenerator()`
Parameter(s):	*value*: `int` *base*: `int`
Return:	nothing

where both `BaseConvert()` calls `NumberGenerator()` if *base* is between 2 and 9; otherwise, it does nothing. And `NumberGenerator()` prints *value* in base notation *base*. In the main function,

1. prompt the user to enter three (3) bases.
2. initialize two (2) int variables to random numbers between 1 and 99. Make sure they are different.
3. display each variable in all three bases.

Note: Include libraries ctime and cstdlib to use `srand()`, `rand()` and `time()` for generating random numbers.

A possible output of the program is:

```
Enter three bases:  2 5 8

28 in base 2 is 011100
28 in base 5 is 0103
28 in base 8 is 034

53 in base 2 is 0110101
53 in base 5 is 0203
53 in base 8 is 065
```

Green text are inputs.