

1 Instructions

Create a folder on the desktop using your person-number as the name. Only files in this folder will be your submission. To pass the practical part your Visual Studio project must compile and meet the criteria when tested. Put your theory answers in a plain text file called `theory.txt` in your submission folder. Create the text file with Notepad (not with Word).

2 Practical Question

You will write a program that calculates and displays Pascal's Triangle. This structure is normally drawn as an equilateral triangle of this form:

```

      1
     1 1
    1 2 1
   1 3 3 1
  1 4 6 4 1
 1 5 10 10 5 1
    
```

Each number in the triangle is the sum of the two numbers above it. For ease of drawing on a console you will draw it shifted to one side as follows:

```

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
    
```

The rule for calculating each number is simple: every number in the first column and the last column of each row is a number one. Every other number is calculated by adding the numbers directly above and to the top-left. For example the number 4 on the 5th row appears because the number 3 is directly above it and the number 1 is to its top-left.

Your program should operate in the following way:

- Prompt the user for the starting row number and input an integer.
- Prompt the user for the ending row number and input an integer.
- Print each row from the starting row to the finishing row on the console.
- Each number in a cell should be justified to an appropriate width.

The example above shows the triangle between row 1 and row 6. Each number is justified to be width two as the largest number in the range chosen is two digits long. In order to know the biggest number in the range you will need to generate the rows first, check for the largest number and count how many digits it contains. If you do not know how to do this properly then make sure you handle numbers up to four digits long when you check. Once you know the width of each number scan through the range again and output the triangle.

3 Theory Questions

1. What are the differences in memory allocation between the stack and the heap?
2. Describe one technique to store non-rectangular 2D data in arrays.
3. In what ways are pointers and arrays similar to one another?
4. Where are local variables allocated?
5. What is the life-time of a local variable, describe the construction and destruction of scopes in your answer?
6. Explain dereferencing of pointers in terms of memory locations being accessed.
7. What is meant by recursion?
8. What parts are necessary in a function definition? Explain giving an example.
9. Describe how an object oriented design can help encapsulation?