**La Salle College   
Algorithms & Programming (420AP1AS)  
  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Michelle M. Khalifé**

**Assignment 3 – for loop patterns**

In order to print the patterns below, you will need to use a nested loop. A nested loop is a loop (the inner loop) within another loop (the outer loop).

Each loop will carry a different responsibility (count vs. range).

For instance, a loop that outputs \*\*\*\*\* wants to print an asterisk five times (count), whereas a loop that outputs 12345 wants to print number 1 through 5 (range). You do NOT want to: printf(“\*\*\*\*\*”); &/or printf(“12345”); This is hard-coding.

The algorithm below is in pseudo-code. It prints pattern #1. Implement it and modify the values of i & j to figure out which part of the pattern each loop is responsible for.

for i ß 1 to 5 // outer loop

        for j ß 1 to 5 // inner loop

            print j

endfor

        print \n

endfor

|  |  |  |
| --- | --- | --- |
| **PATTERN #1**  12345 12345 12345 12345 12345 | **PATTERN #2**  12345 23456 34567 45678  56789 | **PATTERN #3**  00000  .0000  ..000  ...00  ....0 |
| **PATTERN #4**    1 22 333 4444 55555 | **PATTERN #5**  1 12  123  1234  12345 | **PATTERN #6**    5  54  543  5432  54321 |
| **PATTERN #7**  1  21  321  4321  54321 | **PATTERN #8**  1  2 3  4 5 6  7 8 9 10  11 12 13 14 15 | **PATTERN #9**    X  XX  XXX  XXXX  XXXXX |

*Hint: Adopt an incremental approach. Start by capturing the shape of the pattern (rectangle, triangle, …). To do so, replace its content by one character, as displayed in pattern #9. Once you’ve captured the shape, you can then adapt your code to work in the pattern’s actual content.*