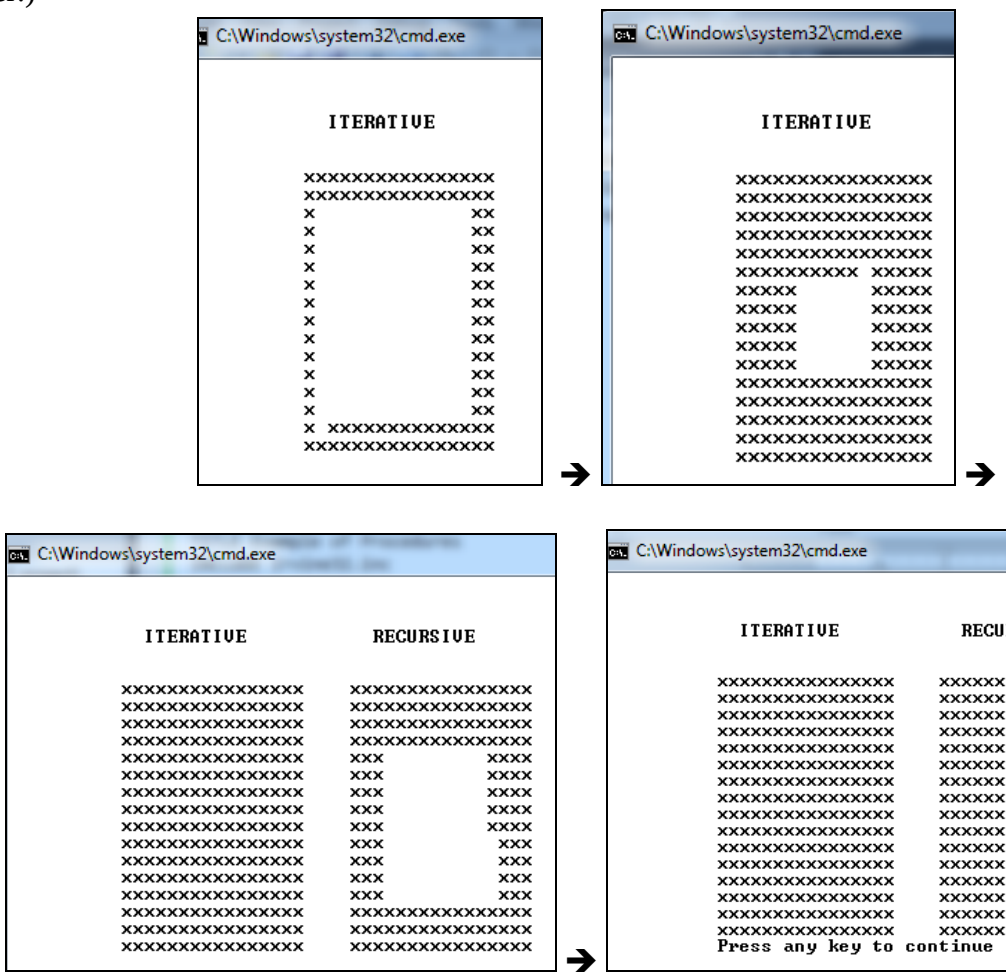
Lab 4: Procedures

Requirements:

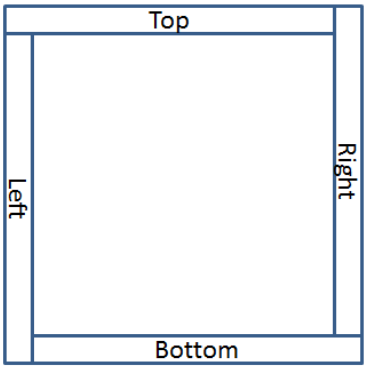
1. This assignment as well as other assignments in this class must be finished on Windows operating system.
2. Zip your program(s) submit on Canvas
3. Due 11:59pm Oct. 16, 2013

Assignments:

In this assignment, please write a program to print square matrices of 'x'. The printing job starts with the left-most corner and proceeds in a spiral way. The program should be paused for 10 milliseconds between printing each two symbols. Below are three snapshots of an execution of the program. (Note the edge length of square is an even number.)



The square can be considered as a set of rings. The program can print the square ring by ring. The printing of each ring consists of the printing of top edge, right edge, bottom edge, and left edge. See the below picture.



In each ring, the four edges are equally long. Each ring is defined by the two corners: top-left and bottom-right. These corners are where the edges start. The length of the edges is (right-top-column) – (left-top-column). If two rings are next to each other, the length of edges in the inner ring is two less than the length of the edges in the outer ring.

There are two ways to construct the program: recursive and iterative. The **iterative** way should obviously build a loop for printing the square one ring at a time. Given the following procedure documentation:

```
; Receives BL (top-left x), BH (top-left y)
;         AL (bottom-right x), AH (bottom-right y)
; Precondition: (BL-AL)==(AH-BH), (BL-AL+1) is even
; Returns: nothing
; Description: Prints a square of constant symbol. The top-left corner of
;              the square is at (BL, BH). The bottom-right corner of the square
;              is at (AL, AH). The program pauses for constant delay_time milliseconds.
;              Algorithm: iteration
spiral_print proc USES EAX EBX EDX ECX ESI
        ...
spiral_print endp
```

, the number of rings should be (AL-BL+1)/2. The loop for printing the rings can enclose four inner loops for printing the edges. But that may make the outer loop very long and hard to understand. In this assignment, please implement procedures for printing edges. Below are examples:

```
; Receives: DH (row), DL (col), ESI (# of prints)
; Returns: nothing
; Description: prints symbol for ESI times from (DL, DH)
;              each print increments DH
vertical_print proc USES EDX ECX EAX ESI
        ...
vertical_print endp

; Receives: DH (row), DL (col), ESI (# of prints)
; Returns: nothing
; Description: prints symbol for ESI times from (DL, DH)
;              each print increments DL
horizontal_print proc USES EDX ECX EAX ESI
        ...
horizontal_print endp

; Receives: DH (row), DL (col), ESI (# of prints)
; Returns: nothing
; Description: prints symbol for ESI times from (DL, DH)
;              each print decrements DH
vertical_print_rev proc USES EDX ECX EAX ESI
        ...
vertical_print_rev endp
```

```
; Receives: DH (row), DL (col), ESI (# of prints)
; Returns: nothing
; Description: prints symbol for ESI times from (DL, DH)
;              each print decrements DL
horizontal_print_rev proc USES EDX ECX EAX ESI
       ...
horizontal_print_rev endp
```

The **recursive** algorithm for this program is based on the idea that the printing of the square can be done by printing the boundary edges and then printing a smaller square if it still exists. The iterative and recursive solutions are very similar except the iterative solution uses a loop to repeat printing the rings, while the recursive one will print the boundary ring and then make a recursive call.

Your program needs to contains both iterative and recursive solutions.

You are given an incomplete program. You are required to use the data segment and the main procedure.

Grading Policies:

Your iterative solution	50%
Your recursive solution	50%