Lab Manual for Computer Organization and Assembly Language

Lab-7

Nested Loops

Table of Contents

1. Introduction

A nested loop is a loop within a loop, an inner loop within the body of an outer one. How this works is that the first pass of the outer loop triggers the inner loop, which executes to completion. Then the second pass of the outer loop triggers the inner loop again. This repeats until the outer loop finishes.

2. Objective

- To know more about Assembly language, such as how to repeat a block of statements using Loop Instructions.

3. Concept Map

This section provides you the overview of the concepts that will be discussed and implemented in this lab.

3.1 Nested Loop Instruction

The JMP instruction can be used for implementing loops. For example, the following code snippet can be used for executing the loop-body 10 times.

```
MOV     CL, 10
L1:
```

```
<Loop-Body>
          Mov cl,5
          L2:
          Loop L2
<end Loop Body>
```

```
Loop    L1
```

The processor instruction set, however, includes a group of loop instructions for implementing iteration. The basic LOOP instruction has the following syntax −

```
LOOP    label
```

Where, *label* is the target label that identifies the target instruction as in the jump instructions. The LOOP instruction assumes that the **ECX register contains the loop count**. When the loop instruction is executed, the ECX register is decremented and the control jumps to the target label, until the ECX register value, i.e., the counter reaches the value zero.

The above code snippet could be written as –

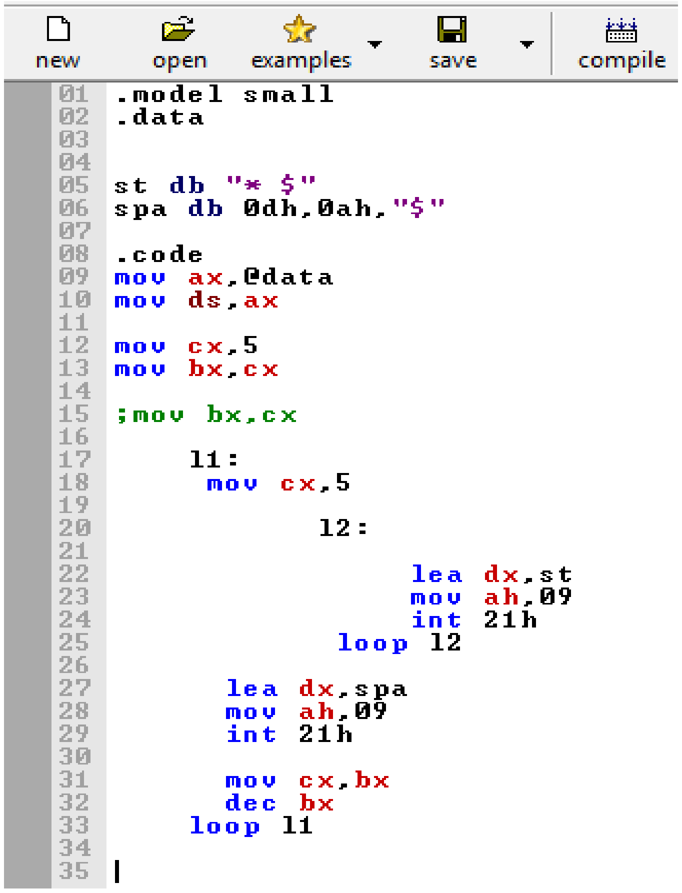
```
mov ECX,10

l1:

<loop body>

loop l1
```

Explanation:
If you need to code a loop within a loop, you must save the outer loop counter's ECX value.

```
           mov ecx, 0

outerLoop:

           cmp ecx, 10
           je done
           mov ebx, 0

innerLoop:
           mov eax, ecx        ; do your thing here
           add eax, ebx

           cmp ebx, 10
           je innerLoopDone
           inc ebx
           jmp innerLoop

innerLoopDone:

           inc ecx
           jmp outerLoop
done:
```

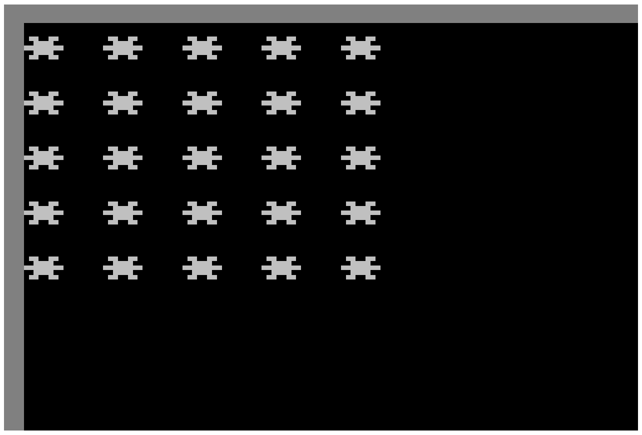
3.2 Example

The following program prints the square pattern.



```
.model small
.data


st db "* $"
spa db 0dh,0ah,"$"

.code
mov ax,@data
mov ds,ax

mov cx,5
mov bx,cx

;mov bx,cx

    l1:
     mov cx,5

           l2:

               lea dx,st
               mov ah,09
               int 21h
            loop l2

      lea dx,spa
      mov ah,09
      int 21h

      mov cx,bx
      dec bx
    loop l1

```

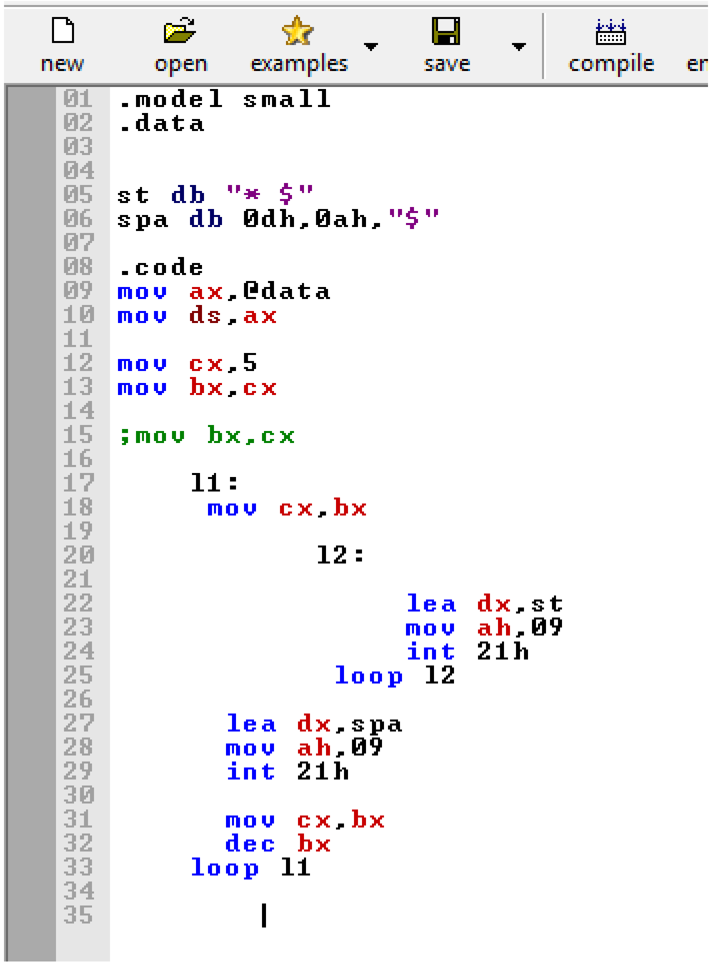
When the above code is compiled and executed, it produces the following **Output.**



```
*   *   *   *   *

*   *   *   *   *

*   *   *   *   *

*   *   *   *   *

*   *   *   *   *
```

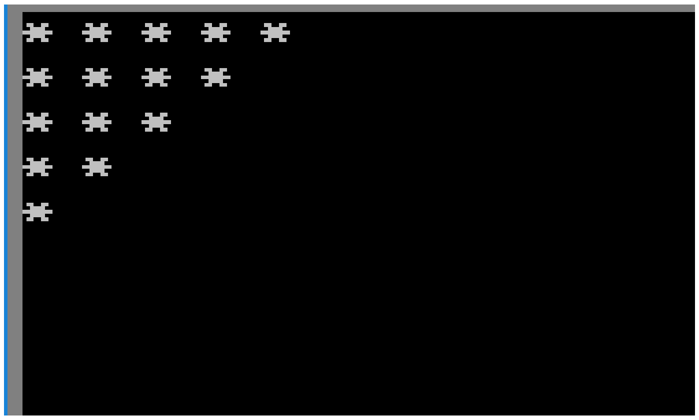
4. Walkthrough Task

Write an assembly program that can print a Triangle



```
.model small
.data


st db "* $"
spa db 0dh,0ah,"$"

.code
mov ax,@data
mov ds,ax

mov cx,5
mov bx,cx

;mov bx,cx

    l1:
     mov cx,bx

              l2:

                      lea dx,st
                      mov ah,09
                      int 21h
               loop l2

      lea dx,spa
      mov ah,09
      int 21h

      mov cx,bx
      dec bx
    loop l1

```

When the above code is compiled and executed, it produces the following **Output.**



```
*  *  *  *  *
*  *  *  *
*  *  *
*  *
*
```

5. Procedure& Tools

In this section you will study how to setup and MASM Assembler.

5.1 Tools

- Download emu 8086 from (http://www.emu8086.com/files/emu8086v408r11.zip)
- Just extract the emu8086.15.zip on C
- Install emu8086

6. Practice Tasks

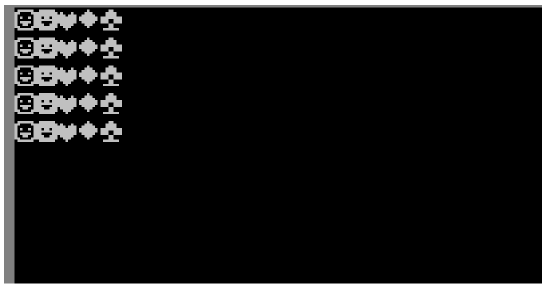
This section will provide more practice exercises which you need to finish during the lab. You need to finish the tasks in the required time. When you finish them, put these tasks in the following folder:
\\fs\assignments$\

6.1 Practice Task 1 **[Expected time = 15mins]**

Write an assembly code that prints the numbers from 1 to 5, 5 times on the screen. Each sequence of numbers from 1 to 5 is separated by new line.
Ex: 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5 1 2 3 4 5

6.2 Practice Task 2 **[Expected time = 15mins]**

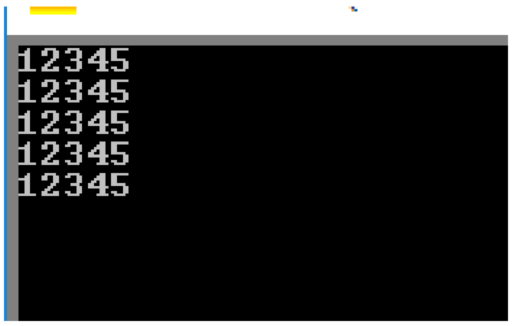
Write an assembly language program using the Loop instruction to print following pattern



```
☺☻♥♦♣
☺☻♥♦♣
☺☻♥♦♣
☺☻♥♦♣
☺☻♥♦♣
```

6.3 Practice Task 3 **[Expected time = 15mins]**

Write an assembly language program using the Loop instruction to print following pattern



```
12345
12345
12345
12345
12345
```

6.4 Out comes

After completing this lab, student will be able to setup emu 8086. He/ She will also be able to compile and run basic Assembly programs.

7. Evaluation Task (Unseen) [Expected time = 30mins for tasks]

The lab instructor will give you unseen task depending upon the progress of the class.

8. Evaluation criteria

The evaluation criteria for this lab will be based on the completion of the following tasks. Each task is assigned the marks percentage which will be evaluated by the instructor in the lab whether the student has finished the complete/partial task(s).

Table 3: Evaluation of the Lab

Sr. No.	**Task No**	**Description**	**Marks**
1	4	Problem Modeling	20
2	6	Procedures and Tools	10
3	7	Practice tasks and Testing	35
4	8	Evaluation Tasks (Unseen)	20
5		Comments	5
6		Good Programming Practices	10

9. Further Reading

This section provides the references to further polish your skills.

9.1 Slides

The slides and reading material can be accessed from the folder of the class instructor available at \\fs\lectures$