National University of Computer and Emerging Sciences

Laboratory Manual

for

Computer Organization and Assembly Language Programming

(EL 213)

Course Instructor	Ms. Sana Fatima
Lab Instructor(s)	Mr. Hazoor Ahmad Mr. Muhammad Salman Mubarik
Section	BSR-3A
Semester	Fall 2023

Department of Computer Science

FAST-NU, Lahore, Pakistan

Exercise1: **[Moving Star]:** Write a function MovingStar that shows an asterisk '*' moving from Top-Left to Top-Right to Bottom-Right to Bottom-Left back to Top-Left. Your program should terminate after one round.

Exercise 2: Update code written in activity 1 such that the star travels the screen in an infinite loop.

Exercise 3: Write a Subroutine 'HollowTriangle', that takes 3 parameters on stack

1. Memory address of the character to be printed to form a triangle
2. Starting row of the triangle
3. Ending row of the triangle

The subroutine then prints the outline of a triangle, using the character specified in memory. It will start in the middle of the screen from the row mentioned as starting row and goes down to the ending row. Starting row will have only one character in it, in the middle of the screen - Column 40. All the rows between the starting and ending row will have two characters in them, with a certain number of spaces between the two characters, number of spaces will depend on row number. For example, the second row (of the triangle) will have one space; the third row will have three and so on. The ending row will be a row of the character.

Example:

Character: db '%'

StartingRow: db 5

EndingRow: db 11

The output will be

Row no	Col …	Col 34	Col 35	Col 36	Col 37	Col 38	Col 39	Col 40	Col 41	Col 42	Col 43	Col 44	Col 45	Col 46	Col …
1															
2															
3															
4															
5								%							
6							%		%						
7						%				%					
8					%						%				
9				%								%			
10			%										%		
11		%	%	%	%	%	%	%	%	%	%	%	%	%	
12															
13															
14															

Exercise 4: Write a function **printNumbers** that take multiple numbers on stack as input and prints them on screen. The function also takes total numbers to be printed as input.

For example, if numbers to be printed are 134, 189, 156 then call to **printNumbers** will be as follow

Start:

Push 134; first number to be printed

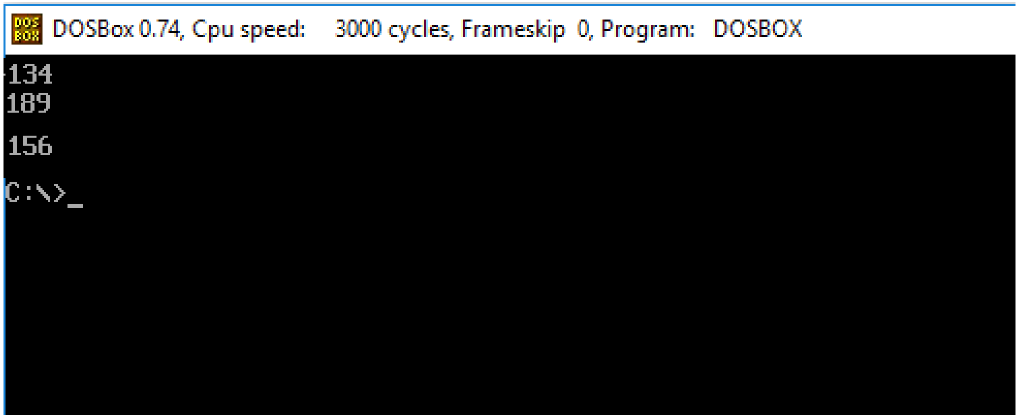
Push 189; second number to be printed

Push 156; third number to be printed

Push 3; total numbers to be printed

Call **printNumbers**

The screen should look like this after this call (be careful about the order in which numbers are printed, it should be same as shown on screen i.e number at end of stack is printed on first line and vice versa)


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: DOSBOX

```
134
189

156

C:\>_
```

You can use **printNumber** routine given in book for this question, and call it multiple times in **printNumbers** function.

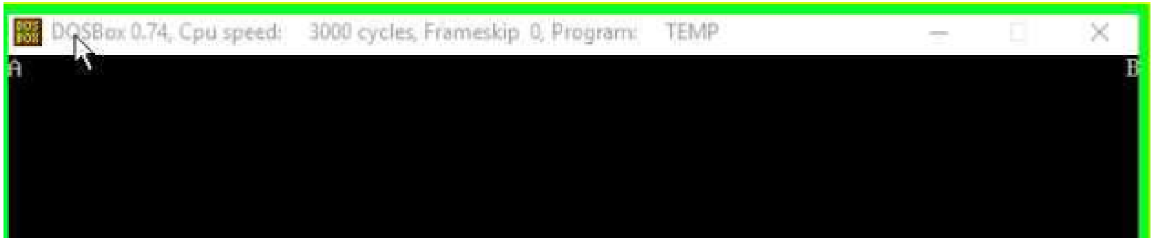
Your function should work with any number of inputs not just 3 inputs

Post Lab

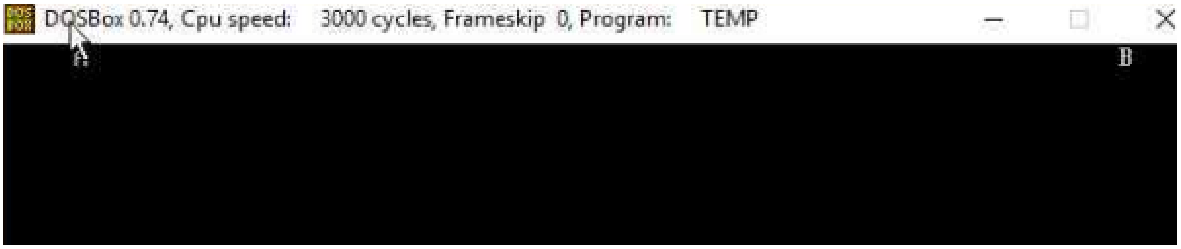
Write an infinite loop that shows ***A*** moving from top left to middle and ***B*** from top right to middle of the screen and then back simultaneously. Use two empty nested loops with large counters to introduce some delay so that the movement is noticeable.

Next few figures show this code in action

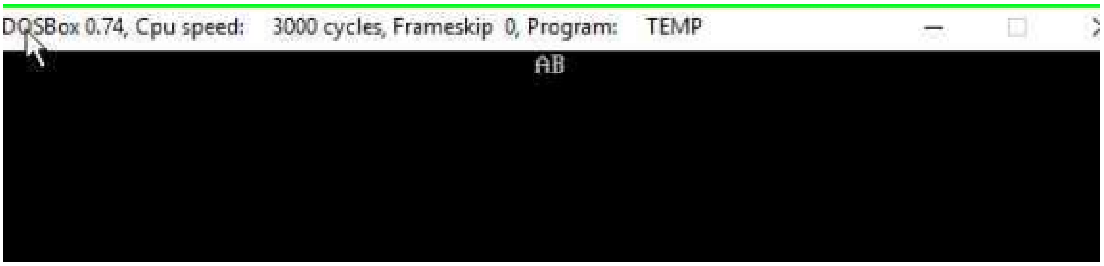
1. Start with A and B in top left and right corners resp.


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TEMP
A
B

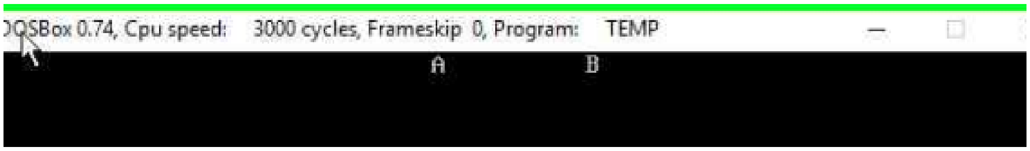
2. Moving towards center


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TEMP
A
B

3. A and B Meeting at center


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TEMP
AB

4. A and B Moving towards their respective corners again


DOSBox 0.74, Cpu speed: 3000 cycles, Frameskip 0, Program: TEMP
A
B

A and B will keep moving like that infinitely