**Lab Manual – Display Memory**

**Important Instructions:**

* **Make proper subroutines**
* **Use Delay Function, that we did in class (if required)**

**Activity 1 [Moving Star]:** Write a function **MovingStar** that moves an asterisk ‘\*’ **by one cell** following these rules:

* If current location of star is ith row and jth column, next location will be ith row and (j+1)th column i.e. the star will be moved rightward by one cell.
* CurrentRow and CurrentCol (global variables) will be (1,0) initially i.e. 2nd row and 1st Column. Global variable Direction will be ‘R’ (Rightward) initially.
* Call your subroutine **MovingStar** from main 80 times (with delay), it should show the star moving in 1st row. Call clrscr function (available in book) before testing MovingStar function. You are not required to pass any parameter to MovingStar or return anything. Your subroutine will update the global variables as required.

Modify the **MovingStar** function with following rules:

* If ‘\*’ is at rightmost column then next location will be jth column and (i+1)th row, i.e. if star is in last column it should move downward **by one cell** (Direction ‘D’).
* Call your function **MovingStar** from main (80+24) times (with delay), it should start ‘\*’ from top-left cell, show it moving through first row then traversing the last column.

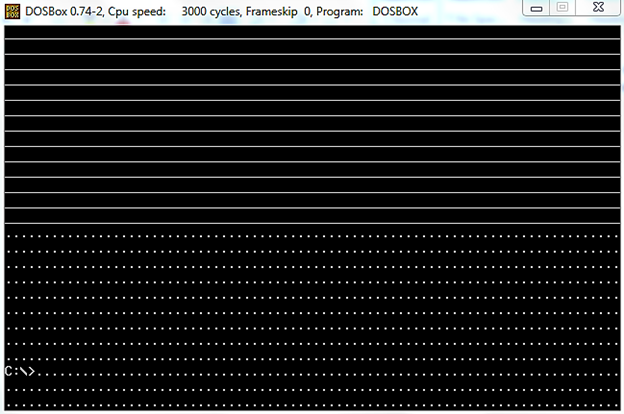
Modify the MovingStar function with Leftward (‘L’) and Upward (‘U’) traversal **by one cell** in last row and first column respectively.

Call your function from main (80+24+80+24) times (with delay). It should show the star moving in the boundary of the screen.

**Activity 2:** Call your function MovingStar in infinite loop (with delay). Your program will display the star traversing the boundary for indefinite interval of time and command prompt will not come back.

**Activity 3:** Code to clear screen is given in example 7.1. Your task is to modify this code and print ‘\_’ (underscore) on first 13 rows of screen and ‘.’ In rest of the rows. Required output is given below. Properly calculate the cells required with each character.

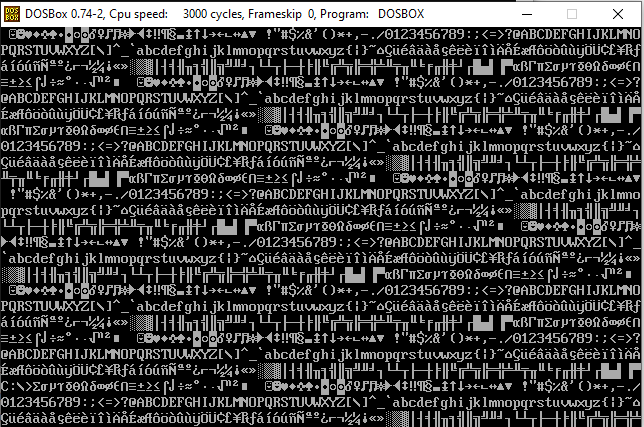
**Required Output:**

****

We did following code in class:

|  |
| --- |
| ; clear the screen  [org 0x0100]    mov ax, 0xb800 ; load video base in ax  mov es, ax ; point es to video base  mov di, 0 ; point di to top left column  ; es:di pointint to --> 0xB800:0000 (B8000)  nextchar: mov word [es:di], 0x0720 ; clear next char on screen  add di, 2 ; move to next screen location  cmp di, 4000 ; has the whole screen cleared  jne nextchar ; if no clear next position  mov ax, 0x4c00 ; terminate program  int 0x21 |

**Activity 4:** Update above code such that it prints all characters (ASCII 0 to 255) on screen starting from location zero onwards and fills all the screen as shown in the figure below:



**Activity 5:** Update the code written in Question 1 to print blinking characters with high intensity as shown in the figure below:

