**Lab Manual – Display Memory & String Instructions**

**Important Instructions:**

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

**Activity 1 [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.

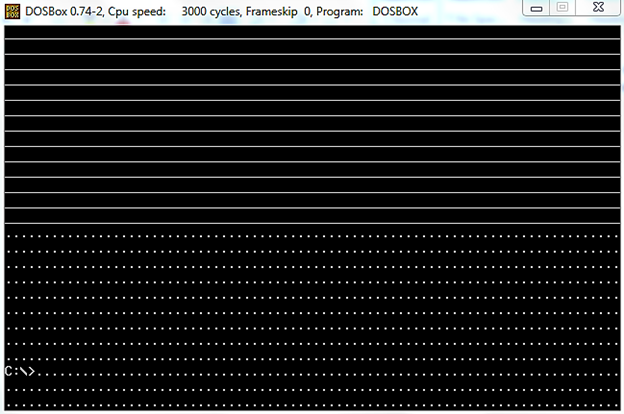
**Activity 2:** Update code written in activity 1 such that the star travels the screen in an infinite loop.

**Activity 3 [Scroll Up and Down]:** We did examples ofScroll Up and Scroll Down in class. Using that code, write a program that scrolls the Introduction Screen 3 rows Up and 3 rows down in an infinite loop without losing any data of screen.

**Practice Problems – Display Memory**

**1-** 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:**

****

**Practice Problems – String Instructions**

1. Write a program that finds total number of occurrences of a character from a null terminated string. For example total occurrences of ‘a’ in string “I am a student of coal” are 3.
2. Write a program that prints tokens of a string on Screen. Best use string instructions studied so far.

**Sample Run:**

|  |
| --- |
| String: I am a student of coal  Output:  I  am  a  student  of  coal |

1. [MOVS] Write a program that copies first 12 lines of AFD Introduction screen in last 12 lines.
2. [SCAS] Write a program that takes a c-string *myStr* and two characters *charToFind* and *charToReplace* from user and replaces all the occurrences of *charToFind* with *charToReplace* in *myStr*. Your program should create a space of 50 characters on heap in order to save *myStr*.

**Sample output:**

|  |
| --- |
| **InputString:** ddsdfhgrtsdfhjghjksdd  **CharToFind:** d  **CharToReplace:** $  **ModifiedString:** $$s$fhgrts$fhjghjks$$ |

1. Write a program that takes a character *ch* and a CString *myStr* from user and removes all the occurrences of *ch* from *myStr*.

**Sample Output:**

|  |
| --- |
| **myStr:** cabccdefcfdcxyzcc  **ch:** ‘c’  **Modified String:** abdeffd xyz |

1. TrimStart(char\* str)

Write a function that takes a string and removes all the space in start of the string.

**Sample Output:**

|  |
| --- |
| **Before TrimStart**  **str:** “ Hello How are you?”  **After TrimStart**  **str:** “Hello How are you?” |

1. String Compression

Write a function that compresses a string by removing consecutive occurrences of same character.

**Sample Run:**

|  |
| --- |
| **String Before Compression:**  **Str:** “ggggdddddddyyyyakxxxuww”  **String after Compression:**  **Str:** “gdyakxuw” |

1. Write a function that searches a substring from a string and highlights the found substring. If the string is not found it will not highlight anything.

**Sample Run:**

|  |
| --- |
| **String:** “I am a student of COAL”  **Substring:** “student”  **Printed String after Search:** “I am a student of COAL” |