**Final Project Report**

**Group II**

Phạm Vũ Minh - 20176818

**Project 9: Ascii Character Graphics Display**

**1. Requirements:**

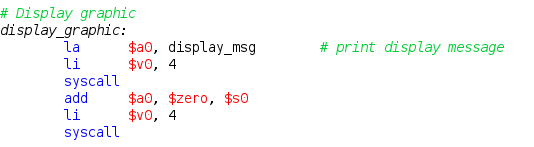
We use MIPS functions to manipulate and display a ascii graphics on the console.

As required, a certain memory region is assigned to store the graphics and all changes to the graphics are done only in this region.

**2. Detail analysis:**

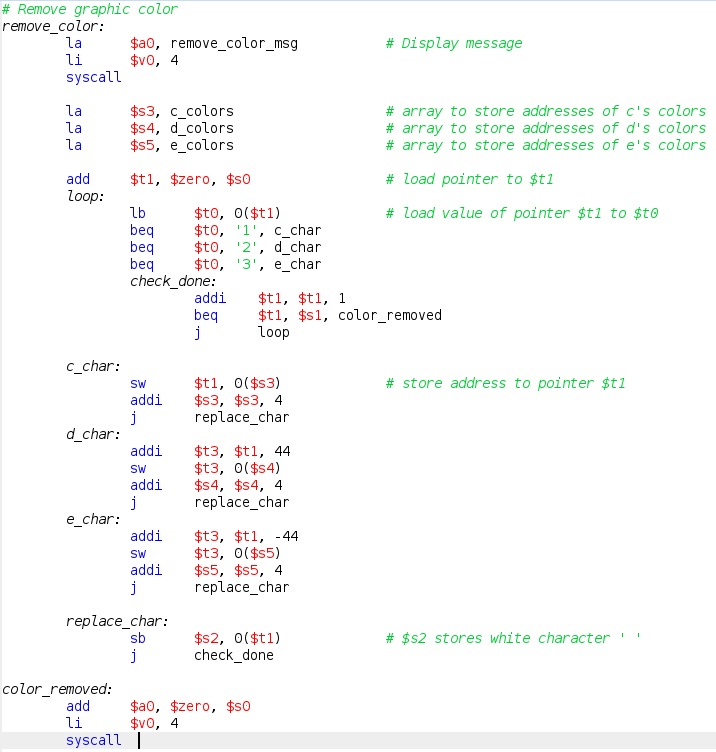
1. Display the graphics

The graphics are stored as a single .*asciiz* label, then it is loaded to $s0 and printed on the console using simple MIPS system call.



1. Remove colors in the graphics

At the beginning, colors in the graphics are represented by the numbers 1, 2 and 3. To remove them, we create a pointer to iterate through the memory assigned to the graphics, byte by byte. If the values in this pointer is 1, 2 or 3, replace them by storing white character ‘ ‘ in its place



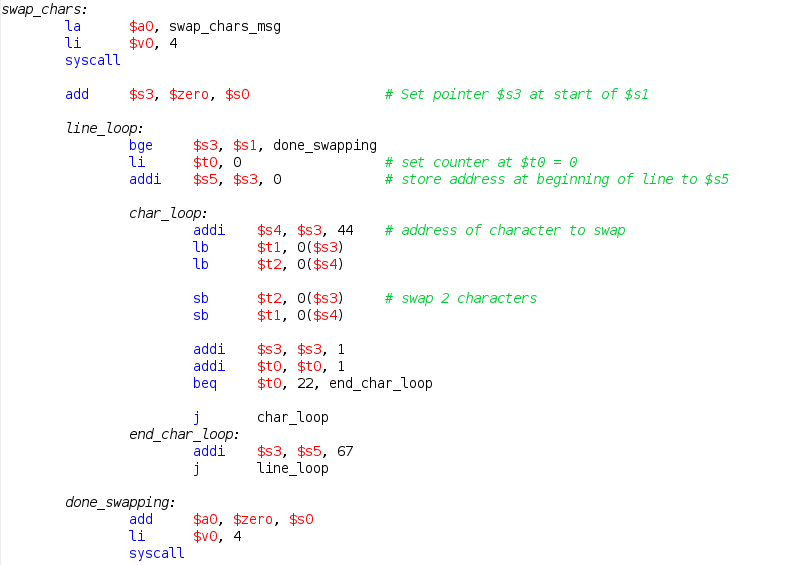
Addresses of these colors characters are also saved and used in later section.

1. Swapping the letters of the graphics to display ECD

The graphics is considered to contain multiple lines, separated by new line character ‘\n’ at the end.

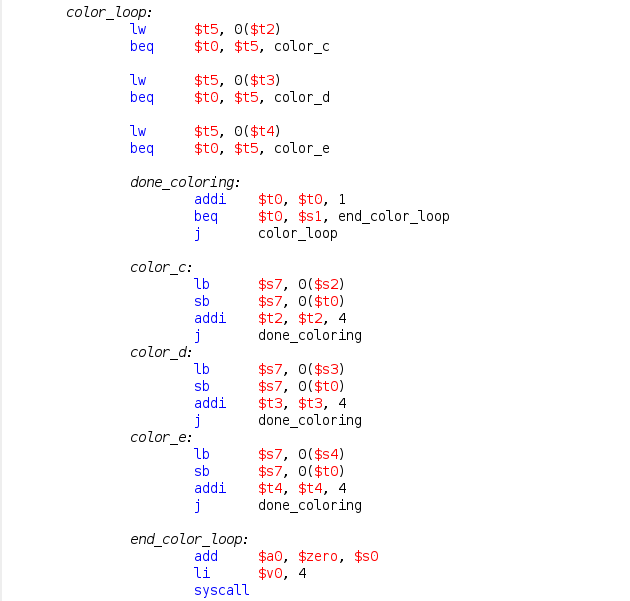
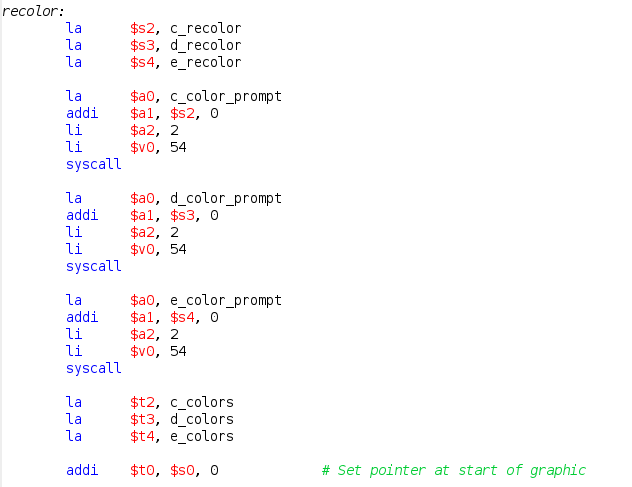
In each line, the letters ‘D’, ‘C’ & ‘E’ is separated into 3 neighboring regions, each 22 characters long. Consider each line to be a string *s*, so *s[0] - s[21]* stores letter ‘D’, so *s[22] - s[43]* stores letter ‘C’, so *s[44] - s[65]* stores letter ‘E’.

We simply iterate through each line and swap the value of ‘D’ and ‘E’ regions



1. Recolor graphics

We prompt the user for colors to fill each letter of the graphics. Then we simple iterate through the graphics and use the colors addresses stored from earlier to save the new colors.



**3. Test results:**

