

# COSC2406 F18 – Assembly Language Programming

## Assignment 5 - Chapter 6

Due: Wednesday November 7<sup>th</sup>, 2018 by 11:55pm

Good formatting and commenting are essential for assembly program. A penalty of up to 20% will be applied if the programs are not properly formatted and commented.

FOR EACH QUESTION: It does help to write pseudo-code to identify the steps necessary to solve these problems. Transfer the refined algorithms to your program as comments and then write the Assembly code to perform the necessary steps. It is recommended that this step is done first, and coding second.

### A) [40] Count Digits and Letters in a text file

- a) Make a procedure, much the like the IsDigit procedure code shown at the end of Chapter 6, only the IsLetter procedure will return ZF=1 if the value in the AL register is a letter, either upper case or lower case. If the value in AL is not a letter, then ZF will return as ZF=0. The procedure must preserve AL.
- b) Write a program that asks the user for the name of a text file located in the same folder as this program's .asm file (such as another .asm file). Open the file using the OpenInputFile procedure. Include appropriate code to detect if the operation fails and report the error.
- c) Once open, read the file into a buffer, 100 characters at a time. Process the characters such that you could the number of digits and letters. You will want to keep track of how many individual letters and numbers – so you can report how many 0's, 1's, 2's...a's b's etc. that are in the file. I suggest using one or two arrays to hold the count values. You will also need some of the procedures from Question A.

Once the entire file has been processed, print the results in a manner similar to the output shown to the right.

Please enter the name of the file:  
Fibonacci.asm

Fibonacci.asm  
count of '0' = 3  
count of '1' = 8  
count of '2' = 7  
count of '3' = 1  
count of '4' = 0  
count of '5' = 0  
count of '6' = 0  
count of '7' = 0  
count of '8' = 0  
count of '9' = 0  
count of 'A' = 31  
count of 'B' = 24  
count of 'C' = 24  
count of 'D' = 20  
count of 'E' = 55  
count of 'F' = 20  
count of 'G' = 8  
count of 'H' = 4  
count of 'I' = 30  
count of 'J' = 1  
count of 'K' = 0  
...

## COSC2406 F18 – Assembly Language Programming

Your program should have the following custom procedures: (a) one to process the characters read into the buffer and count each letter and number; and (b) one to print the count results. Both these functions must preserve the registers: one will use USES and the other push/pop operations.



### B) [40] Coloured Rotating Text

For this program, you will ask the user for a String which will be a maximum of 40 characters long. You will then generate two random numbers using your BetterRandomNumber procedure from Assignment #4 – these numbers being between 0 and 15 inclusive. Save the numbers as the foreground and background colours. You will also set to variables called row and col to 5 – which will be the row and col numbers where the text printing will start.

Then, loop 300 times and inside the loop: a) set the cursor at row and col; b) set the text colour; c) write the string; d) advance the foreground colour and every time it reaches zero, decrement the background colour; e) rotate the string one character to the left – the first character will become the last character; f) advance the row value until it reaches 31 and then reset it back to 10 and increase the col value; and g) delay the program 1/20 of a second.

When the loop is done, set the cursor to row 30, and column 0.

Here is an algorithm defining this problem which will give you an example of how to create a comment-based algorithm to solve the problems. More comments may be necessary and you can add them as required.

```
; collect a string from user (max 40 char)

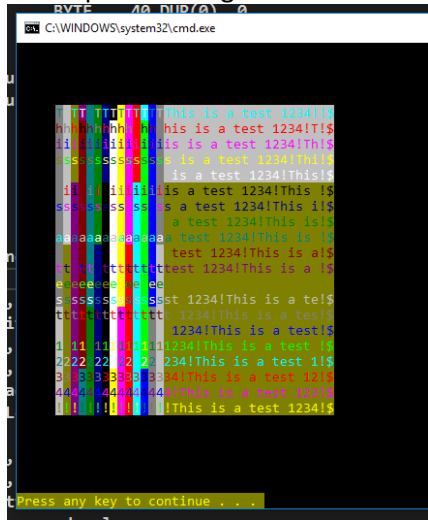
; generate random foreground and background colour, store in vars

; clear the screen
; set row, column variables initially to (5, 5)
; loop 300 times
; --> set cursor at (row, col)
; --> combine and set the colour
; --> write the string
; --> advance foreground colour, if hits zero, decrease background colour (wraps around)
; --> rotate string to left, first char to end
; --> advance cursor row value, if it hits 25, reset to 5 and increment col value
; --> delay 1/20 second

;set cursor to row 30, column zero
```

## COSC2406 F18 – Assembly Language Programming

Also required: at least two of the steps in the main loop (the 300 loop) must be placed into functions which preserve registers.



For all of these problems, a right-side column of comments must be included to explain the logic of your code. Writing the logic in Java or pseudo-code is acceptable. When I read your comments, I should clearly see your algorithm – not a description of each assembly instruction. Also, there must be proper comments at the start of every procedure as shown on pages 146-47, and comments with your name, course code, date, assignment/question number, and problem description at the top of every program. Marks will be deducted for missing commenting. *Failure to comment the code properly or to format like the code in the textbook will result in a penalty of up to 25%.*

Demonstrate and explain your working programs to the TA, Glenn Driver (gdriver@algomau.ca) by Wednesday Nov 14<sup>th</sup>. The demonstration and explanation of your code will represent 25% of the assignment grade – the other 75% of the grade will come from grading the code and how well the programs meet the specification and requirements of the question.

### IMPORTANT:

- If you do not demonstrate your programs to the TA by the end of the demonstration date, you will receive a grade of zero.***
- If you are unable to explain most or all of your code and the logic of your programs, then the maximum grade possible will be 50%.***