

ECE 33: Introduction to Computer Engineering: Fall 2015

Programming Assignment II : DUE: Oct 28, 2015

This assignment is supposed to teach you the use of buffers to store data, complex loops and manipulation of non-numeric data.

Write a program in 8085 assembly language that would read a line of text and insert a word into it at a given word position.

The program should begin with a welcoming message. It should then ask the user to type in a sentence. The sentence will have a maximum of 40 characters. It will end with a period and may contain symbols such as a comma or a dollar sign. It should then ask the user for a word (user will complete the word by hitting <CR>). It should also read in a number (maximum two digits) specifying the insertion position. (user will complete the number by hitting <CR>). The program should then print the sentence by inserting the new word in the original sentence at a word position supplied by the user. In particular, the new word should be inserted after i words in the sentence where i is the number typed in by the user. If i is larger than the total number of words, the new word should be added as the last word in the sentence (just before the period).

Two typical sessions with this program are shown below.:

The Sentence Editor

Type a sentence: I am feeling good.

Now give a word: not

To insert after: 2

Corrected sentence: I am not feeling good.

The Sentence Editor

Type a sentence: I am going.

Now give a word: inside

To insert after: 7

Corrected sentence: I am going inside.

Note that in this second example, the user wants the word inserted after 7 words in the sentence while the sentence has only 3 words. Consequently the word is inserted at the end of the sentence. (It should be inserted before the period as shown).

For extra credit, (1) allow the word to be inserted before the sentence if the user indicates so by giving the insertion position as 0, and (2) allow the sentence to be ended with a question mark in addition to a period. This is shown in the following example.

```
The Enhanced Sentence Editor
Type a sentence: Isn't it nice?
Now give a word: why
To insert after: 0
```

```
Corrected sentence: Why isn't it nice?
```

Note that the first letter of the new word has now been capitalized and the first letter of the previous first word (now, the second word) is made small. If you are attempting this extra credit, please make it clear in the screen title (“The Enhanced Sentence Editor” as against “The Sentence Editor”).

Name the program prog2. Upload the file prog2.asm to the coursesite by 11:59, Oct 28, 2015. Your submission time will be the time stamp on the web submission. Your program should clearly identify you in the program header.

The programs will be graded for efficiency (compactness of the code and speed of execution), correct programming style, proper labels, good comments and pleasing interaction with the user (user interface). Please use your own imagination while designing the screen layout and messages. See the three example programs (example1.asm - example3.asm) to learn the correct programming style.