ELEC 379 : DESIGN OF DIGITAL AND MICROCOMPUTER SYSTEMS
1999/2000 WINTER SESSION, TERM 2

Assignment 2 - 8086 Programming

due
Monday, February 21 2000

Question 1

Write a program in 80x86 assembly language that reads one character at a time from the keyboard. Those characters that are vowels (a,e,i,o,u and y) (and only those characters) are stored in a buffer. This buffer should be 10 bytes long. Vowels entered after the buffer is full should be discarded. The program should stop reading characters when a ’$’ is entered.

After the ’$’ character is entered, the program should output a carriage return and line feed, followed by the contents of the buffer, converted upper case, with a space character between each vowel, and a final carriage return and line feed.

Your code should make use of only these DOS interrupts: int 21H function 1 (AH=1) to read a character, int 21H function 2 (AH=2) to print a character, and int 20H to return control to DOS.

For example, if the input is:

```
John Brown (12345678). The lAzY brown fOx jumps on the qUick dog’s back$
```

then your program should print:

```
O O E A Y O O U O E
```

Your program must follow the documentation conventions for assembly-language programs given on the course web page.

Your program should make use of at least four functions defined by you that do the following: (1) read a character, (2) print a character, (3) determine if a character is a vowel, (4) convert a character to upper-case. Your functions should use the callee-saves calling convention.

Submit a listing of your code and a print-out of its output. You can redirect the output of your program to a file using ’>’ (e.g. `asg2 >`*filename*). Submitting output that was not generated by the program in the accompanying listing will be considered academic dishonesty and will result in a mark of zero or worse.

Assemble and test your code using the free valarrow DOS assembler and linker available from the course web page.

Bonus marks will be awarded for the shortest and for the easiest-to-read solutions.

You may find it useful to first write and test your program in a high-level language (e.g. C) and then convert it to assembly language. To start a new line you must print carriage return (0DH) and line feed (0AH) characters.