Department of Electrical and Computer Engineering
University of Texas at Austin

EE 306
Fall, 2003
Instructor: Tony Ambler
TAs: Sundar Balasubramaniam, Ming Ding, Ashwin Gunasekar, Aamir Hasan, Harshal Ingole, Ritu Kar, Paroma Sen, Sujata Sharma, Raghuveer Simha

Programming Assignment 4
Due: Midnight – 5th December, 2003

Instructions
Download program4.asm and modify it. Submit the .asm file, not the .obj file.
Write your name at the beginning of the program file you submit. You will be penalized if you don’t. Be sure to write it as a comment, like so
; Last name, First name
No late submissions will be accepted. So make sure you start early.
Programs you submit must be your own work. You may discuss algorithms with your classmates. You may not discuss your program with others. Do not show your code to any of your fellow students. Plagiarism of any kind will not be tolerated.

Shaping Subroutines

Subroutines are primarily used for two reasons - to break down a large problem into smaller parts, and to reuse code. In this assignment, you will use a subroutine called print to write several subroutines of your own. Your subroutines will then be called by the main program to draw shapes. Drawing is done by using spaces ( ) and stars (*). You do not have to write the main program, just the subroutines specified.

Description
There are 4 subroutines in this program – PRINT, HLINE, VLINES AND EQTRIANGLE.
PRINT – this subroutine is already written for you and should be used in the other subroutines. It prints a sequence of spaces and stars. The interface (specification) of this subroutine is
R1 – number of leading blanks (spaces)
R2 – number of leading stars
R3 – number of intermediate stars
R4 – number of trailing stars

So if print is called (jumped to) with R1 = 2, R2 = 3, R3 = 3 and R4 = 2, it will print the following to the console. There is **no newline at the end.**

```
***   **
```

HLINE – should print a horizontal line of the specified length, followed by a newline. The interface is
R1 – number of leading spaces
R2 – length (in terms of number of stars) of the line

V_LINES – should print a set of vertical lines. It also has a newline at the end. The interface is
R1 – number of vertical lines
R2 – length of each line
R3 – spacing between lines
R4 – leading spaces before 1st line

With R1 = 3, R2 = 4, R3 = 3 and R4 = 6, this should print
```
*   *   *
*   *   *
*   *   *
*   *   *

```

E_QTRIANGLE – should print an equilateral triangle. Again, there should be a newline at the end. The interface is
R1 – length of each side
R2 – leading spaces before base of triangle

With R1 = 4, and R2 = 6, this should print
```
*
* *
* *
* * *
* * * *

```

Note that the number of stars in the base will not be equal to the length. It will be (2*length - 1).
Specifications
You should write your subroutines in the program4.asm file provided (download program4.asm and modify it). The PRINT routine is already written. The 4 .FILLS at the beginning of the file should be filled in with the addresses of the subroutines. The address of the print routine is x4010. You will have to fill in the rest. These addresses will depend on your implementation and will likely be different from other students’. These addresses will be used by the main program to call your subroutines. It is absolutely essential that you get them right.

Write the HLINE, V_LINES, and EQTRIANGLE subroutines. These subroutines should use the PRINT subroutine. Believe me, PRINT makes life much easier.
Do not forget to put the newline character at the end of the subroutine.

Hints
Save the values of registers you use in your subroutines. Go through the PRINT subroutine. It will help you write your own subroutines.
Make sure you understand what you need to do. We have tried to make the description fairly detailed. But talk to Dr Ambler or the TAs if you have any questions. The crux of this assignment is understanding the interface. Writing the subroutines is not difficult.
There is a simple main program in main.asm (posted along with assg). It just calls the HLINE subroutine. HLINE is a good starting point. We’ll post a more sophisticated main program soon.
We’ll also post clarifications as and when ambiguities show up. So check the web page regularly in the next few days.