

Assembly Language Lab: The Guessing Game

CSC 242 Lab 1

Handed Out: see course calendar

Due Date: see course calendar

Instructions:

Your program will simulate a simple guessing game. The program has stored the value 6. The program will continually ask the user to guess a number between 0 and 9 (see sample program input/output below.) The user enters the guess from the keyboard.

If the guess is larger than 6; the program should output: Too big.

If the guess is smaller than 6; the program should output: Too small.

When the user finally guesses correctly, the program will output: Correct!

You took # guesses.

where # is the number of guesses it took the user.

If after 9 guesses, the user has not correctly guessed the number, the program should output: Game over. Correct answer is 6.

You do not need to check for invalid input (alphabetical characters, punctuation, etc). If your program encounters any non digit character, the program should output Invalid input. Invalid input should still count as a guess.

You should use TRAP x20 (GETC), to read in a character from the keyboard. See Table A.3 in Appendix A of the textbook for a listing of all trap vectors. You can use the pseudo-op .STRINGZ to store a string into your program.

Hints and suggestions

Remember, all input and output functions use ASCII characters. You are responsible for making any conversions that are necessary. Before sitting down at the computer, give some serious thought to what this program involves. Break it into sections. You might even want to draw a simple flow chart. Comment your code! Not only does this help you debug your program, but it will make it easier to remember what you did, should you need to look back at your code for future projects (hint, hint).

Your program should be written in LC-3 Assembly language, and tested using the LC-3 Simulator. The completed program should work using input from the keyboard and output to the monitor. Don't forget the ASCII code x0A. It causes the cursor to go to the next line (Newline).

Sample Input/Output

Guess a number: 5
Too small.
Guess again: 9
Too big.
Guess again: a
Invalid input.
Guess again: 6
Correct! You took 4 guesses.

Writing and testing your program

Your code should have a comment block at the beginning of the file containing your name, your email, student number, and a brief description of the program. Your description should serve as a general summary of your program's approach to the problem and will aid in grading. It is in your best interest to make all of your ideas clear through this summary and through commenting within your code.

IMPORTANT NOTE: The first line of your program must specify the memory address of where you want your program to be placed (using the .ORIG pseudo-op).

We request you place your program at x3000.

When and how to submit: The programs will be submitted in class by turning a hard copy of the programs with your full name and date. The program will be evaluated during class. The instructor or TA will evaluate documentation and logic outside of class.

Grading: 50% for technical approach to be illustrated through the program description and comments in the code, 30% properly executing; 20% for formatting and programming style.