**Files to submit**: editDist.s
**Time it took Matthew to complete**: 2 hours (but I was super rusty when I did it. Probably 45 – 60 mins now)

- All programs must compile without warnings when using the -Wall and -Werror options
- Submit only the files requested
  - Do **NOT** submit folders or compressed files such as .zip, .rar, .tar, .targz, etc
- If submitting in a group on Grade Scope please make sure to mark your partner.
  - Only one of you has to submit there
- Your program must match the output exactly to receive credit.
  - Make sure that all prompts and output match mine exactly.
  - Easiest way to do this is to copy and paste them
- All input will be valid unless stated otherwise
- Print all real numbers to two decimal places unless otherwise stated
- The examples provided in the prompts do not represent all possible input you can receive.
- All inputs in the examples in the prompt are underlined
  - You don't have to make anything underlined it is just there to help you differentiate between what you are supposed to print and what is being given to your program
- If you have questions please post them on Piazza

Write an assembly program called **editDist.s** that calculates the edit distance between 2 strings. An explanation of what edit distance is can be found here while accompanying pseudo code can be found here.

1. The label for the first string should be **string1** and the label for the second string should be **string2.**
2. The edit distance between string1 and string2 should be placed in **EAX**.
3. For each string please allocate space for 100 bytes.
    1. While you must allocate space for 100 bytes in your final submission you will likely find it easier to work with the .string directive for testing and debugging.
4. **AFTER** the last line of code that you wish to be executed in your program please place the label **done**.
    1. Make sure that there is an instruction after the done line and a new line after that instruction. If you don't your output won't match mine.
5. I have included a C implementation of the edit distance program. I highly recommend translating this solution into assembly as it will make your life much easier.
    1. As a note remember that constants cannot be swapped. Pay careful attention to this in your solution
    2. Use subroutines. It makes life easier (in my opinion)
6. I have included a Makefile that will compile your program. Your program must be able to be compiled by this Makefile when you submit it
7. **IT IS OF VITAL IMPORTANCE THAT YOU NAME YOUR LABELS AS SPECIFIED AND MAKE THE APPROPRIATE AMOUNT OF SPACE FOR EACH VARIABLE!** I will be using gdb to test your code and if your labels do not match then the tests will fail. You must also make sure to include the done label AFTER the last line of code you want executed in your program so that I know where to set break points.