Due: Smartsite Wed., 10/22, 11:55 p.m.
Names of Files to Submit: **64bitAdd.s, editDist.s , ReadMe.txt**

- All prompts for input and all output must match my prompts/output. We use a program to grade your work and tiny differences can cause your work to be marked as a 0.
  - If your work does happen to lose points you have 1 week from receiving your grade to request a regrade and we will be happy to go back and fix any typos in your prompt/output.
  - The best way to avoid being deducted points is to copy the prompt/unchanging portion of the outputs into your code. Make sure to get the spaces as well.
- You must also submit a file called ReadMe.txt. The format is as follows
  - student: first name last name, id
  - student: first name last name, id
  - …
  - notes: Any difficulties you had with the assignment and any comments you would like to make. Also if a program doesn't work please let us know.
- An example ReadMe.txt has been included with this assignment
- Submit your work on Smartsite by uploading each file separately. **DO NOT** upload any folders or compressed files such as .rar, .tar, .targz, .zip etc.
- **ALL** students in a group should upload the submission to SmartSite.
- If you have any questions please post them to Piazza

1. (Time 15 min)Write an assembly program called **64bitAdd.s** that adds two 64 bit numbers together. The first number will be referenced by the label **num1** and the second number will be referenced by the label **num2**. The upper 32 bits of the sum should be placed in **EDX** and the lower 32 bits in **EAX**. **AFTER** the last line of code that you wish to be executed in your program please place the label **done**. I have included a Makefile for you that will compile your program. The following table shows how the numbers will be laid out in memory.

| | | |
|---|---|---|
| num1: | Upper 32 bits of num1 | Lower 32 bits of num1 |
| num2: | Upper 32 bits of num2 | Lower 32 bits of num2 |

2. (Time 2 hours) 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. The label for the first string should be **string1** and the label for the second string should be **string2.** The edit distance between string1 and string2 should be placed in **EAX**. For each string please allocate space for 100 bytes. 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. **AFTER** the last line of code that you wish to be executed in your program please place the label **done**. I have included a python implementation for you to compare your answers against. You can run the program by running the following command: python3 editDist.py string1 string2. If you inspect the source code of the python program you will see that there are 2 different implementations. You can implement either in assembly. I have included a Makefile for you that will compile your program.

**IT IS OF VITAL IMPORTANCE THAT YOU NAME YOUR LABELS AS SPECIFIED AND MAKE THE APPROPRIATE AMOUNT OF SPACE FOR EACH!** 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.