This is a warm-up project whose objective is for you to set up the infrastructure you need for subsequent assignments. Help is available through **piazza** on our course t-square site, during **TA office hours** (http://www.ece.gatech.edu/academic/courses/ece2035/help/index.html), and by sending **email** to ece2035-help@ece.gatech.edu. There are also helpful links to tutorials and primers here: http://www.ece.gatech.edu/academic/courses/ece2035/assignments/index.html.

HW1-1: The goal of this part is to use a Linux/Unix environment and become familiar with its facilities. Toward this end, you must either acquire/access and run an appropriate Linux/Unix distribution that fits your computing environment or connect to a machine in the Klaus 1448 Linux cluster. Then perform the following exercises.

1. Create and edit a text file that contains the following:
   - Tell one interesting fact about yourself.
   - Which Linux/Unix distribution are you using (e.g., Ubuntu 14.04.03)?
   - The method you are using to run Linux/Unix. Some examples:
     - Unix underlying Mac OS X
     - VMware on Windows 10 (recommended for Windows 10 users)
     - Virtual Box on Windows 8.1
     - Dual booting Ubuntu 12.04 and Windows 7
     - Red Hat on Linux machines in Klaus room 1448
     - Running native Ubuntu 14.04.3 LTS
2. Find a suitable application to capture a screenshot which shows your text file open in an editor, running under Linux/Unix.
3. Submit your screenshot in **jpeg format** in a file of **size no greater than 200K**.

In order for your solution to be properly received and graded, there are a few requirements.

1. The file must be named **`HW1-1.jpg`**.
2. The file must be less than 200K bytes.
3. Your solution must be properly uploaded to the T-square site before the scheduled due date, **5:00pm on Friday, 29 January 2016**.

HW1-2: The goal of this part of the project is to modify a short C program, compile it using the GNU C Compiler `gcc`, and run it. A program shell `HW1-2-shell.c` is provided. You must copy/rename it to `HW1-2.c` and modify it to compute how many elements in the first integer set (`SetA`) are also in the other (`SetB`). Your program should print out the number of elements in the intersection (`IntersectionSize`) using the print statement in the shell code. `SetA` always contains eight integers and `SetB` always has ten integers. Both sets are globally declared. Assume the two sets are "sets" in that there are no duplicate elements within a set but they may be in any order.

You should open a "terminal window" to run `gcc` under Linux/Unix (type `man gcc` for compiler usage or look up GCC online documentation on the internet). Note that in the terminal window, you can enter any of the Linux commands (such as `ls`, `cd`, `cp`; for reference see http://users.ece.gatech.edu/~linda/2035/Linux_Cmd_Cheatsheet.pdf). Use the Linux command `cd` to change your current working directory to the directory in which you placed the shell program. For example,

```
> cd ~/Documents/2035/hw1
```

You can list the files in that directory using

```
> ls -la
```

You can copy a file using `cp` or rename a file using `mv` (move a file to a new file). For example:

```
> cp HW1-2-shell.c HW1-2.c
```

You can use any of the available text editors normally found on linux, including `vi`, `vim`, and `emacs`. Using the text editor of your choice modify the `HW1-2.c` program to compute the intersection size as described above.

Once you write your program, you can compile and run it using the Linux command line:

```
> gcc HW1-2.c -g -Wall -o HW1-2
> ./HW1-2
```

You should become familiar with the compiler options specified by these flags.

In order for your solution to be properly received and graded, there are a few requirements.

1. The file must be named **`HW1-2.c`**.
2. Your name and the date should be included in the header comment.
3. The starting *shell* program should not be modified except for the replacement of the comment "*`// insert your code here`*" and the addition of declared local variables. It is especially important not to remove or modify any print statements since they will be used in the grading process.
4. Your solution must include proper documentation and appropriate indentation.
5. Your solution must be properly uploaded to T-square before the scheduled due date, **5:00pm on Friday, 29 January 2016.**

HW1-3: The goal of this part is for you to install MiSaSiM, modify a short assembly program `HW1-3-shell.asm`, simulate, test and debug it in MiSaSiM. The MiSaSiM simulator can be installed according to the instructions at www.ece.gatech.edu/~scotty/misasim/. Copy or rename the shell program to `HW1-3.asm` and modify it to compute the number of elements in common in `SetA` and `SetB` and store that number at the memory location labeled `Result`. Again, assume that `SetA` has eight integers and `SetB` has ten, and that the two sets are "sets" in that there are no duplicate elements within an individual set but they may be in any order.

In order for your solution to be properly received and graded, there are a few requirements.

1. The file must be named **`HW1-3.asm`**.
2. Your name and the date should be included in the beginning of the file.
3. The starting *shell* program should not be modified except for the replacement of the comment "# write your code here..."
4. Your program must store its result at the memory location labeled `Result` when it returns. This answer is used to check the correctness of your code.
5. Your program must return to the operating system via the **`jr`** instruction. *Programs that include infinite loops or produce simulator warnings or errors will receive zero credit.*
6. Your solution must include proper documentation.
7. Your solution must be properly uploaded to T-square before the scheduled due date, **5:00pm on Friday, 29 January 2016.**

Honor Policy: In all programming assignments, you should design, implement, and test your own code. **Any submitted assignment containing non-shell code that is not fully created and debugged by the student constitutes academic misconduct.** The only exception to this is that you may use code provided in the examples on the course website as a starting point for your programs (http://www.ece.gatech.edu/academic/courses/ece2035/examples/index.html).