

COP 3404
Systems Software
Project 1 (50 points)

Project 1 due 9/14/18 at 11:59 PM (nearly midnight)

This is essentially a simple data structure project.

For this project you are to implement the symbol table that will be used by the assembler during pass 1 and pass 2. It should be constructed as an efficient hashing table. You should construct a "main" routine that will invoke the symbol table operations. The main routine should read a file name off the command line.

The file will consist of a character string and an optional number one per line.

For example the file might look like:

```
moss 25
eno
fred
gorge 18
```

The actions should follow the following rules.

- 1) upon seeing moss 25
 - hash moss creating a location in an array.
 - if moss already exists, report an error, for example:
(ERROR moss already exists at location 8)
 - if moss does not exist, store the name and its number
when moss is stored print a lines such as:
stored moss 25 at location 8 (where 8 is the array index where moss is stored
reporting collisions, if necessary)
- 2) upon seeing eno
 - hash eno to find the location in the array where eno may
or may not exist.
 - if eno does not exist, report an error
(ERROR eno not found)
 - if eno does exist, report the location in the array and its number.
(eno found at location 12 with value 433)

Clearly, the 12 and 433 are numbers created for this example. 433 would have been the number associated with eno when it was stored, like 25 was the number associated with moss, in the above example.

You will have at least one printed line for each input line in the file.
(Maybe more when printing the occurrence of collisions)
You must be able to handle collisions.
You must write a hashing function, you may not use one built in the language.

You should include appropriate documentation associated with your project in a file called "text" containing name, purpose of project, external files for input and output, and general description of how the problem is solved. You should use a makefile to compile and link your program even if you code in java or other languages. Create good modular code.

COP 3404
Systems Software
Project 1 (50 points)

None of this program should be interactive; no menus, prompts or other action requested by your program should be employed.

You may include any other files as you desire in your shar. Be sure to test the integrity of your shar by creating a new directory, copying your shar file to the new directory, unsharing, and making the project. Only after this testing procedure has been accomplished should you turnin your project. A heavy penalty will be assessed for projects that do not make (compile). Even if you use a language like python, you must include a makefile.

You should not shar a directory, ie when I unshar your project a new subdirectory should NOT be created.