CS 100 Software Construction
Homework 1

- Linux (10 pts):
  - make a directory in your home directory called cs100
  - create a subdirectory called hw, then a subsubdirectory called hw1
  - put all your files created for this homework in ~/cs100/hw/hw1
- C++ (40 pts):
  - Write a class, called *Stack*, in a file called stack.h, that implements a stack of up to 1000 characters. You should include the following member functions in your class. You need not handle any errors (e.g., by throwing exceptions), but you should detect them and print an error message if they occur.

```
#define STACK_CAPACITY 1000
class Stack
{
  public:
    Stack(); // constructor for a stack
    void push( char c ); // adds c to the top of the
stack
    char pop(); // removes top element, returns it
    char top(); // returns the top element
    bool isEmpty(); // returns true iff the stack is
empty
    ~Stack(); // destructor for a stack
};
```

  - Write a main, in a file called stack.cc which includes stack.h, that repeatedly reads a character string from *cin* into a string variable and outputs the reverse by pushing the characters onto an instance of your stack class, then printing them as they are removed from the stack.
  - Modify your program to exit on end of file (when the user types a ^D, on Unix).
  - Compile command, on well.cs.ucr.edu (failure to compile = no credit)

```
$ g++ -o stack stack.cc
```

- Systems Programming (50 pts):
  - Write a program that implements a rolodex. Write the entire program in one file called rolodex.cc. Your program should repeatedly do the following: present a text menu of commands, read a single letter command, and execute the command (using a switch statement). Commands to implement are:
    - i - insert a new address (prompt for name, address, phone number, retain letter case)
    - f - find a given address (prompt for last name, case-sensitive)
    - d - delete a given address (prompt for last name)
    - p - print all cards in the rolodex (output should look nice and they should be in ascending alphabetical order by last name)
    - l - load addresses from a given rolodex file (prompt for file name)

        - s - save the addresses to a specified rolodex file (prompt for file name)
        - q - quit the program (if rolodex is modified, but not saved, your program should ask the user if they want to save or discard changes, then call save() if the user confirms they want to save)
        - Compile command, on well.cs.ucr.edu (failure to compile = no credit)

$ g++ -o rolodex rolodex.cc

- What to submit:
    - A Unix script showing correct execution of all your commands. See man script for more information.
    - package using tar cfv HW1 to create a single file to submit
    - test that your tarfile extracts correctly with tar xfv HW1
    - Test the extracted archive you created on Well.cs.ucr.edu or one of the lab machines so the TA can do the same with your submitted work.