CSCI-14 Assignment 10 – structs. (100 points total) Due 5/1/18

1. Create a struct MenuItem containing fields for name (a string) and price (a float) of a menu item for a diner. Create a ReadItem() function that takes an input stream and a MenuItem (both by reference) and prompts the user for the fields of the MenuItem, loading the values into the struct's fields. Create a PrintItem() function that takes an output stream (by reference) and a MenuItem (by value) and prints the MenuItem fields to the stream in a reasonable one-line format. Create a main() that declares a MenuItem struct, calls ReadItem(), then calls PrintItem(). Test with a few different items. (20 points)

2. Copy your program and modify it to include an array of 10 MenuItem structs. Modify ReadItem() to not prompt for, but instead just read the data for a struct, and call it once for each MenuItem. Then call PrintItem() for each struct to create a menu. Use files, opened and closed in main(), for all input and output. Create a test file with the 10 items and prices, and test with it. Send me the test file and the output file. (30 points)

3. Copy your program and modify it so that after it reads the MenuItems from the file (as per part 2), it prints a menu (on the screen) for the user, collects an order, and prints a diner check. You should not have to modify PrintItem(). Write a function that takes the array of MenuItems and an array of ints, prints the menu (number the items 1 through 10), and then loops, prompting the user to select items until they have either selected all 10 items on the menu or entered -1 to indicate they are through ordering. Don't allow the user to select an item that isn't on the menu or an item they have already selected. You may assume the user will enter at least one valid selection. Keep the user's selections in the array of integers. Return the number of entries the user made. Then print a diner check with the items the user selected and the total price of the meal. Use another function to print the check, passing the two arrays, the number of items selected, and the output stream to the function. Use the PrintItem() function to print the items, both on the check and on the menu. You may assume no tax. Print all output to the screen. (50 points)

Hints:

If you use an `istream&` for an input stream parameter, you may pass either `cin` or an input file handle to the function. Similarly, if you use an `ostream&` for the output stream parameter, you may pass in either `cout` or an output file handle to the function. The only restriction is that you may only use operations that would work on `cin` or `cout` in the function. For example, you may not `open()` or `close()` the file inside the function.

You may use either C-strings or C++ string objects for the menu item names. Remember, to read a C-string containing white space characters, use

```
in.getline( data, MAX, '\n' );
```

where `in` is either `cin` or an input file handle (or a parameter of `istream &` type), `data` is an array of `char` at least `MAX+1` in length, `MAX` is the maximum number of characters to read from the input stream, and `'\n'` is the character at which to stop reading. Also, to read a C++ string object containing white space characters, use

```
getline( in, data );
```

where `in` is either `cin` or an input file handle (or a parameter of `istream &` type) and `data` is a C++ string object. Here, you don't have to worry about the maximum length of the string being read.

To number the items in the third exercise, print the number before calling PrintItem(). PrintItem() should not end the line itself.