COSC 1437 Graded Exercise #3

This batch mode program reads and processes a text file using command line input redirection. The program will also use command line arguments to select the sort key and sort order.

For example, this program can be run from the command line as in the examples below:
Note that the command line arguments can be entered in any order

```
GE3  name  asc < TestFile1.txt > NameAsc.txt
GE3  score asc < TestFile1.txt > ScoreAsc.txt
GE3  name  des < TestFile1.txt > NameDes.txt
GE3  score des < TestFile1.txt > ScoreDes.txt

GE3  asc name  < TestFile1.txt > NameAsc.txt
GE3  asc score < TestFile1.txt > ScoreAsc.txt
GE3  des name  < TestFile1.txt > NameDes.txt
GE3  des score < TestFile1.txt > ScoreDes.txt
```

The first line in the file used by this program is an unsigned int value indicating the number of lines of data that follow. Each additional line in the file contains a student name and a test score. The format of the text file is as follows:

```
100
“Smith, John J”         99
“Adams, Susan Ann”  86
```

Note that the student name is enclosed in double quotes. The opening quote is the first character in the line. Assume the name may be up to 30 characters including whitespace characters within the double quotes but not counting the double quote characters themselves. An unsigned int value representing the score follows the name after 1 or more tab or space characters separating the name and score.

Assume that the line length will be 80 characters or less not counting the newline characters at the end of the line. Lines are terminated with CR and LF characters (Windows style text files created using Notepad).

The program should do the following:

0. Process and save the command line arguments for later use.
1. Read the number of data items from the file.

2. Dynamically allocate one array of pointers to structures. The structure that is pointed to by the pointers in the array should have the following definition:

```
struct Student
{
   char * namePtr;
   unsigned int score;
};
```

3. While there are still data lines to be read from the file:

3-1. Read and parse a data line from the file. You will need to separate the name from the score and store these two items temporarily.

3-2. Dynamically allocate memory for a Student structure. Place the address of this Student structure in the array of structure pointers. You will also need to dynamically allocate just enough memory to store the name. Do not store the quote characters. Populate the fields of the Student structure with the two items from Step 3-1.

4. Based on the command line arguments, sort the array of structure pointers using the proper sort key and the proper sort order.

5. Send the sorted results to standard output. The output format should be similar to the input format (including the use of quotes) except the items should be properly sorted. It should be possible to use the output file created by this program as the input file to this program. The code that sends results data to standard output should ideally be encapsulated into one function to make code porting to other environments easier.

6. Free all dynamically allocated memory.