Problem
Code a program to check the file formats for an on-line Movie Review System.

Discussion
There are three types of file formats to be reviewed, (i.e., checked and validated): 1. category file; 2. movie files; and 3. review files. A category file will have a `.vrs` extension. There can be only one category file in the system, but multiple Movie and Review files. The Category file will contain information about all of the other system Movie files. Movie files will have a `.mov` extension and will include brief information about the Review files. The Review files will contain complete movie information and the actual review. See "Input" section for a discussion of the specific Category, Movie and Review file formats. The interface for this program will be extremely simple consisting of a query and answer dialog with the user. This program will later be incorporated into a Movie Review System.

Execution
The program should initially present the user with a startup screen displaying the name of the program, a brief 3-4 line explanation of the program and the programmer's name and email address. After the user hits the return key, the startup screen is to be cleared. A brief help screen should appear explaining the program, which is cleared upon the pressing of the return key.

Following the brief help screen the user is prompted for the name of the Category file. The name of the Category file is input and opened for reading. By default the program is to assume that the Category file is located within the current directory. (The user is responsible for entering a path name specification if the file is located in a different directory.) If the Category file does not exist, the program is to halt with an appropriate error message. If the Category file does exist, it is read, and checked for format errors. The Category file, all Movie files referenced by the Category file and all Review files referenced by the Movie files, (that do not contain format errors), are checked for existence, opened, validated and line by line corresponding to their file format. For each file that is checked and validated, the program should include a brief entry in the report file, (and echo it to the screen). See the Output section for the report file description. After the Category file, Movie files and Review files have been processed, the program should issue a brief termination message before returning to the operating system.

Structure Charts
A copy of the initial structure chart design must be submitted on **Friday Sept. 12th**. A final structure chart reflecting design changes and corresponding to the code must be produced and submitted with the executable. The initial chart will be compared against the final chart to determine the quality of the design.

Input
The format of the Category file requires that the first line contain a positive integer in columns 1-10. Any remaining data on line 1 after column 10 is ignored. (This integer represents the number of lines that follows in the file, which contain the brief Movie category records.) All other lines, beginning with line 2, of the file, the movie records, are organized as described in the following table giving the column field format and contents.

columns	*contents*	*comments*
1-20	category title	Text description of the category.
21-24	integer	The size of the category, i.e., the number of contained reviews.
25-80	file name	The file name of the Movie file with extension (.mov). May contain relative or absolute file pathname.

The format for a Movie file requires lines to be organized as follows.

columns	*contents*	*comments*
1-29	title	The title of the movie.
30-34	real	Star Rating: [1.0, 5.0]
35-80	file name	The file name of the Review file with extension (.rev). May contain relative or absolute pathname of the file.

The star rating, in columns 30-34, has a further limitation of the decimal part being either .0 or .5. No other fractional portions are legal.

The format for a Review file requires lines organized as follows. Extra lines, after the last specified full movie review line, will be ignored and should not generate an error message.

line	*contents*	*comments*
1	title	The title of the movie.
2	director	Movie director.
3	year	Release year.
4	stars	Names of the stars.
5	brief review	One line review.
6	positive integer	Number of lines of full review.
7-EOF	text	Full movie review.

The program need not check the contents of each line of a movie review file. It need only determine the existence of each line.

Output

The output report must be echoed to both the screen and the output text file "xxx.rpt", where xxx is replaced by the Category file name. Each file validated must have entries similar to the following:

```
category.vrs
===============
File is correctly formatted.

horror.mov
===============
Line 1: Invalid star rating.
Line 3: Invalid movie file extension.

teachers.rev
===============
Line 6: Invalid full review size.
```

Assumptions

It may be assumed, for ease of initial implementation, that a maximum of no more than **100** total files, (Category, Movie and Review), will be checked and validated. (This maximum file limitation will be removed later.) Lines in all files should be no more than 80 characters in length. Characters beyond the 80th column can be ignored. Data files will be provided. Downloading instructions will be available from the course Web site: (`http://ei.cs.vt.edu/~cs1704`).

Grading

The due date for this program is Wed., Sept. 24th. An array of struct, (records), must be used to store complete movie information, excluding the full review. Separate compilation is also required for this program. Failure to produce a separately compiled program will result in grade penalties. Turn in hard copies of the following: source code, output files, initial and final structure charts. Submit a diskette (system labeled: 95 or NT), with files containing: ASCII source code, executable image, I/O files, and a brief ASCII *readme* file with execution instructions. The executable image must be called *reviewer.exe*. The files on the disk should not be compressed and file protection should be set to allow anyone access. All deliverables must be placed in an envelope folder, neatly labeled. Only 3 1/2 inch DOS (FAT 1.4 MB) disks are to be submitted. In addition, the GTAs may require you to demonstrate your program. To receive partial credit for programs that are non-working, or are not fully functional, a brief one or two paragraph description of the problem(s) must be included in the assignment folder. The location, routine minimum, must also be specified along with possible corrections that need to be made.