CS 1044 Programming Assignment # 5 **Summer Session #1, 2002**

Project Objectives:

This programming assignment uses many of the ideas presented in various sections of the course notes, so you are advised to read those carefully. Read and follow the following program specification carefully.

Consider collating and reporting data on the TV watching habits of a group of people. You are given a list showing time intervals during which members of the group watched various channels, as described in the input section below. You must produce a table summarizing how much each channel was watched and identifying the channel(s) that were watched most.

Due Dates:

The due date for this project is 11:59:59pm, June 26th (Wednesday evening). The drop dead date for this project is 2pm, June 27th (Thursday afternoon). You have a maximum of 5 submissions to the grader for this project.

Input file description and sample:

Your program must read its input from a file named *watch.dat*. The first line of the input file contains the *number of lines of data that your program is to process*. (**Note:** there will be at least that many lines of input, and there may actually be more - process the specified number of lines and ignore any additional lines of input.) The remainder of the input file will consist of one or more lines of data, each containing:

- a channel name (character string, at most 20 characters)
- a day code (either U for Saturday or S for Sunday)
- a starting time (in hh:mm format)
- an ending time (in hh:mm format)

Each data item is separated by a single tab character. You may assume that all the input values will conform to the description above (no decimal numbers or missing colons, for instance). However, *do not assume that the numbers are always in the correct logical range*. For example, there may be an invalid time value like 12:78 or 27:03, invalid day code, or a channel name that's not in the specified list! The list of channels and their corresponding channel numbers is as follows:

Channel Name	Number
WPXR	1
TBS	2
CBS	3
NBC	4
ABC	5
Headline News	6
Comedy Central	7
HBO	8
Showtime	9
ESPN	10
TNN	11
QVC	12
PIN	13
USA	14
FOX	15
HBO2	16
SciFi	17
CSPAN	18
Lifetime	19
HSN	20

An example of the input file is:

```
17
TBS       S 08:30 10:00
WPXR      S 13:00 15:28
TBS       S 21:00 22:30
NBC       S 22:00 23:30
Headline News   S 07:00 07:10
HSN       S 14:30 15:14
WPXR      S 15:15 15:17
Headline News   U 07:00 07:10
HSN       U 16:00 17:02
Headline News   U 07:00 07:10
SciFi        U 18:25 19:00
HSN       U 19:09 20:15
Headline News   S 07:00 07:10
WPXR      S 11:00 11:30
Headline News   U 07:00 07:10
FOX       S 27:45 00:15
Headline News   U 07:00 07:64
SciFi     S 13:25 15:00
```

In this project, the input values are separated by tabs. Thus, for each line of input (from the 2nd line on...) the format is: <station name><tab><day code><tab><start time><tab><end time>

If the start time is later than the end time, the day code is invalid, the channel name is invalid, or one of the times is invalid for any reason, ignore that line of data.

What to Calculate:

You will write a program which will read the data in the input file and:

- calculate the total time each channel was watched on each day, and also the sum for both days
- calculate the total time TV was watched each day
- sort the channel data in descending order by total watch time

Output description and sample:

Your program must write output data to a file named *tv.dat*. The sample output file shown below corresponds to the input data given above: (note that the line starting with Peter is the first line of output (insert your name, not mine!), and the line starting with Total is the last line of output. There is no blank line before the Peter line or after the Total line (the software I use to type this handout adds extra space).

```
Peter DePasquale
Nielsen Ratings

 Channel          Sat         Sun        Total
 ----------------------------------------------
 WPXR             0:00        3:00        3:00
 TBS              0:00        3:00        3:00
 HSN              2:08        0:44        2:52
 NBC              0:00        1:30        1:30
 Headline News    0:30        0:20        0:50
 SciFi            0:35        0:00        0:35
 ----------------------------------------------
 Total            3:13        8:34       11:47
```

Note that in the event of a tie, the order in which you print those channels must be the same as mine (ascending channel numbers). To ensure that, use selection sort and a strict comparison (< instead of <=).

You are not required to use this exact horizontal spacing, but your output must satisfy the following requirements:

- You must use variables of type int for all the numbers used in computations. This rule includes the reading of the time input.
- You must use the specified title and include your name in the first line as shown.
- You must use blank lines exactly as shown.
- You must arrange your output in neatly aligned columns, with a label identifying the contents of each column.
- You must use the same ordering of the columns as shown here.
- You must use exactly the same column, row and status labels shown here.
- If the number of minutes in a time value is less than 10, you must print a leading zero as shown.

Programming Standards:

All the standards specified for Project 4 are in force for this project as well. The additional requirement is that you must use a single array of records (struct-type variables) to store the channel statistics in this program. Additionally, you must:

- define and call no less than 7 user-defined functions (excluding the main function)
- pass an input file stream to one user-defined function, and pass an output file stream to one user-defined function (not the same function as the input file stream)
- pass the array of structures to at least 3 functions, using the proper mechanism (pass by reference or constant reference)
- not use any file-scoped variables (although file-scoped constants and type definitions are acceptable)

Your implementation must make use of **at least seven** user-defined functions, besides main(). At least one of these functions must explicitly return (using a return statement, not a pass by reference) a structure type (not an array of structures).

Hints:

You should think carefully about the design of this program before implementing it (writing code). I would suggest the following general strategy.

- Begin with your solution to Project 4 and modify it to use an array of struct variables to hold all the channel data. (You'll still want the constant array of channel names, but probably just to initialize your struct variables.) This will require making some modifications to your input code, your calculation code and your output code.
- Modify your input code to read in the channel name (instead of channel number). Now, you'll have to search your array of channel records to find the right one to update.
- Add a selection sort function to sort the array of channel data.

Still don't know where to start? Try attempting a structure chart and then come see me or a teaching assistant (be sure to bring the chart with you!)

Testing:

At minimum, you should be certain that your program produces the output given above when you use the given input file. However, verifying that your program produces correct results on a single test case does not constitute a satisfactory testing regimen. You should make up and try additional input files as well; of course, you'll have to determine what the correct output would be.

Honor Pledge:

The following pledge is required to be present in the beginning of your source code file (near the top of the file) along with the project header. Obviously, you should replace the appropriate line with your name.

```
////////////////////////////////////////////////////////////////////////////////
// Honor Pledge
//   On my honor:
//
//   - I have not discussed the C/C++ language code in my program with anyone
//     other than my instructor or the teaching assistant assigned to this
//     course.
//
//   - I have not used C/C++ language code obtained from another student, or
//     any other unauthorized source, either modified or unmodified.
//
//   - If any C/C++ language code or documentation used in my program was
//     obtained from another sources, such as a text book or course notes, that
//     has been clearly noted with a proper citation in the comments of my
//     program.
//
//   - I have not designed this program in such a way as to defeat or interfere
//     with the normal operation of the Automated Grader.
//
//  YOUR NAME GOES HERE!
////////////////////////////////////////////////////////////////////////////////
```

Hand Grading:

We will be conducting hand grading on your highest graded submission. If two or more submissions result in the highest grades, the first submission will be used for the hand grading process. Your source code will be evaluated for the following:

- Presence of honor pledge (failure results in a 5 point deduction)
- Presence of project/program header; statement of purpose (failure results in a 4 point deduction)
- Comments within the body of the program (failure results in a 3 point deduction)
- Comments for all variables and constants (failure results in a 3 point deduction)
- Use of descriptive variable names for ALL variables (failure results in a 3 point deduction)
- Presence of name, platform information, date, SSN in header (failure results in a 2 point deduction)
- Use of all ints for numeric values (failure results in a 5 point deduction)
- Presence of at least 7 user-defined functions (failure results in a 10 point deduction)
- Passing input and output streams to two separate functions (failure results in 10 point deduction)
- Use of a single array of structs to organize and store TV data. (failure results in a 25 point deduction)
- Presence of a function that returns a single structure. (failure results in a 10 point deduction)

Points deducted as the result of hand grading will be deducted from the highest graded submission to the Curator and will be the recorded grade for this project.

THE USE OF GLOBAL VARIABLES IS STRICTLY FORBIDDEN IN CS 1044 AND MAY RESULT IN AN ADDITIONAL 25 POINT DEDUCTION DURING HAND GRADING. GLOBAL CONSTANTS AND GLOBAL TYPE/STRUCTURE DEFINITIONS ARE PERMISSIBLE.

As stated in the honor code, mentioned in class, and addressed in the syllabus... projects are to be worked on individually. Instances of collaboration with other students, outside help, etc. will be prosecuted under the VT Honor Policy. No exceptions.