

CS23021 Computer Science I

Spring 2008, Project #1

Committed by 11:59pm on Friday, October 3, 2008

Preliminaries

Write a program that processes integer number input values and outputs them based upon their sign (+/-) using different HTML font tags embedded in HTML table tags. The HTML table will have four columns where the first three columns contain the numbers read in as input and the fourth column will contain the the sum of the three input values in the same row. Any input values (or summation values) that are negative numbers should be output in red color. Your program should stop reading in input values when an input value of 0 (zero) is encountered. The zero value should not be output as a value in the HTML table. Empty table cells should be output to complete the entries in the last row of input values in the table, if necessary. Following the last row of input values, one additional row will need to be output. This very last row must contain the sum of all numbers in each respective column and the total of all input numbers in the last cell (bottom right side cell of the table).

The best practice for attacking this project (or any project for that matter) is to sketch out a design for the different parts of the control logic before you begin writing the actual c++ code for this program. In other words you need to have a clear understanding of the problem, including the form of the input and the desired output. Keep in mind that the output of you program is an ordered group of HTML tags that encode an HTML table that can contain dynamic input (i.e. the values read-in by your program). You will use unix redirect operators to redirect the input and output streams so that the input and out to you program can be read-in from the command line (or input-file) and written out to an output file that you will be able to view in a HTML browser. More information on the use of these redirect operators is provided below. Designing your solution before you start writing c++ code will save you a tremendous amount of time and help you to figure out what the specific lines of code will need to be in order to complete this project.

HTML (HyperText Markup Language) is the language that is used in writing web-pages. The language consists of formatting elements called HTML tags. A HTML table starts with a `<table>` tag and ends with a `</table>` tag. The information in the table is organized by rows. Every table row starts with a `<tr>` tag and ends with a `</tr>` tag. The rows are divided into table cells. Every cell starts with a `<td>` tag and ends with a `</td>` tag. Information entered in the table cell is formatted and displayed by a web browser. The color with which content is output is controlled by the `` and `` tag pair. For example, ` hello, world ` displays “hello, world” in the color red in a browser. More on HTML can be found at various locations on the internet, such as: http://www.w3schools.com/html/html_intro.asp. Your program will output, using `cout` statements, HTML table tags and the cell values (the values entered as input) so that the table will be viewable in a browser.

An example of an HTML table formatted as described in this project is available in file the `example.html`, which you can view and download from the course web-site. You can view this file in a browser by clicking on it. While working on your own tables you can always view them in a browser by downloading (sftp) them to your PC and opening them in a browser (e.g. FireFox, I.E., or some other browser) by selecting “File”→“Open” from the pull-down menu of your browser.

Your program will read-in the input using the standard input stream (`cin`), and send the output to standard output stream (`cout`). You can use the unix/linux redirect operators (`<` and `>`) to allow your program to get the values from a file and send the output to another file. For example, the command:

```
prompt% table < inputfile > table.html
```

allows the program `table` to receive input from a file named `inputfile` (you may use any filename you like – but the values in the file must conform to the specification of this project) and then all the output of the program `table` is sent to a file named `table.html`. The file `table.html` can be viewed in a browser. Once again, you

may use any file you like for the output file but it should have the extension `.html` if you want to easily view it in a browser). Note, that if you are using `gdb` to debug your program you can use the debugger's `run` command to redirect your input/output:

```
(gdb) run < inputfile > table.html
```

Assignment

Obtain copies of the project files from the course website at: <http://www.cs.kent.edu/~volkert/cs23021/S08/>.

You will need to create a `Project2` directory in your `cs23021_student` directory. You will then need to use the `svn add` and `svn commit` commands to put this new directory under subversion control. The file `exampleinput` contains an example input file that the program can use as input. Note, that your program has to work on **ANY** line of numbers, not only on the numbers specified in the example file `exampleinput`, and must be able to process more or less lines of input than are in the example file provided. Test your program with different input integer sequences (you can for instance create a variety of input files to use for testing, or you can simply enter values in from the keyboard). To recap, for this assignment you will write a C++ program `table.cpp` such that it outputs the tags representing an HTML table with four cells in each row. The first three cells of each row will contain values read-in to the program. The fourth cell of each row will contain the sum of the previous three cells. All negative numbers should be output so that they display in the color red. You are **only** allowed to use branching and looping constructs as the control structures in your code. You are **not** allowed to use functions or arrays (material that we have not studied yet).

Please note that your `table.html` file will not show up as a rendered webpage in the `svn` repository unless you issue a subversion command to set the mime-type of the file. You can do this using the `svn propset` command as illustrated below

```
svn propset svn:mime-type text/html table.html
```

After you have issued the `svn propset` command you will need to commit the changes using the standard `svn commit` command.

Make sure that your program contains ample comments and otherwise adheres to good programming style. A link to a programming style guide is available on the course webpage. Your grade will be lowered if you do not follow the programming style guidelines outlined there.

Have fun with this assignment! If you're looking to be more creative feel free to experiment with other HTML formatting tags to make your output more colorful.