

CSC 104 Assignment 3, Winter 2012

Due by the end of Wednesday March 21, 2012; no late assignments without written explanation.

Unrelated note: Lab 5 will require understanding the Bresenham line algorithm as covered in lecture on March 27.

Related note: Be sure to comply with the course rules about doing your own work, as outlined at the end of the course information sheet.

Part 1: Date calculation

Create a spreadsheet in *Gnumeric* called “datediff.gnumeric” into which the user can type two dates (in four cells; the first three letters of the month name go into one cell and the day number of the month goes into another cell, like in lab 4). They will just type the month and day number; the dates are assumed to be in the year 2012. Calculate the number of days between the dates by converting each first to a day number of the year as in lab 4. For example, there are 47 days between March 14 and April 30, since March 14 is the 74th day of the year and April 30th is the 121st day of the year, and $121 - 74 = 47$. (Use the `abs()` function so that it doesn’t matter which way you subtract or which date is later.)

Make sure your output cell is labelled so that the user can tell what it’s supposed to be. Submit your spreadsheet with sample dates filled in.

Part 2: A simple model

You are designing a house. You need to include a 20 square metre tiled area. The tiles are 40 cm (0.4 metres) in each dimension.

The area will be rectangular.

If you made the area 4 metres wide by 5 metres long, then the width would be ten tiles wide ($=4/0.4$) and the length would be thirteen tiles long (because $5/0.4$ is twelve and a half, and the additional half-tiles are wasted). Thus altogether you would use 130 tiles (10×13). (Assume that cutting this kind of tile ruins the remaining portion; you can’t cut one tile into two useable half-tiles.)

Alternatively, if you made the area one metre wide by 20 metres long, you would need three tiles to cover the width and 50 tiles to cover the length, thus using 150 tiles altogether.

What is the shape which yields the smallest number of tiles? Do not state an answer to this question, but create a spreadsheet to determine it, or to allow someone else to determine the answer for other parameters.

First, produce a little table at the top with the two parameters: The required tiled area size and the size of each tile. Then, produce a chart with columns:

- width — this will go from 0.5 metres to 4.5 metres (inclusive) in steps of a half metre. I.e. 0.5, 1, 1.5, 2, etc.
- length — this is the entire required area (20 in this example) divided by the width
- width-tiles — the number of tiles required to stretch over this width
- length-tiles — the number of tiles required to stretch over this length
- total tiles — the number of tiles required, which is simply the product of the entries in the previous two columns.

Your spreadsheet must be a “Gnumeric” spreadsheet and should be named “tiles.gnumeric”.

Note that references to the little table at the top need to be absolute.

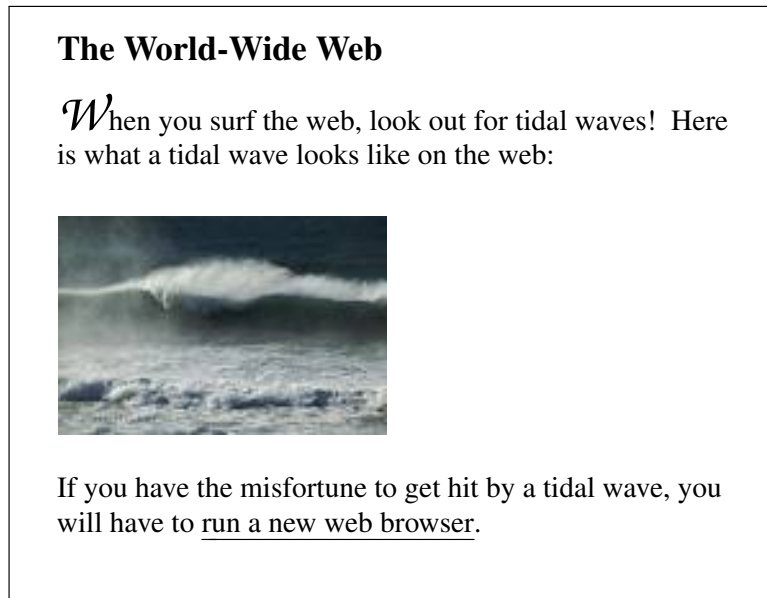
You will want to use the `CEILING` built-in function. This converts a real number to an integer by “rounding up”. For example, `CEILING(3.3)` is 4, and `CEILING(3)` is 3.

(continued)

Part 3: Write some web pages in simple HTML

For this part of the assignment you need to write two web pages, using only structural tags.

One page, with file name `surfing.html`, should look like this:



You may use any (inoffensive) picture you like for the picture of a tidal wave. Perhaps find something interesting (but small) on the web.

The fancy *W* can be made to appear in your text by using the image at

`http://www.dgp.toronto.edu/~ajr/gen/fancyW.gif`

with an `` tag. Be sure to use suitable 'alt' text.

The text “run a new web browser” is a link to another page in a file named “`browser.html`”. The `browser.html` file should contain instructions about how to run a web browser at CDF (probably just one sentence), and also a link back to the `surfing.html` page.

In your HTML files, use only structural tags, not tags which directly specify the desired appearance. Structural tags include `<html>`, `<head>`, `<body>`, `<title>`, `<h1>`, `<p>`, ``, ``, `<sup>`, etc. Appearance tags which you should not use include ``, ``, `
`.

(An exception, not relevant here, is that you could use `<i>...</i>` to indicate the *English-language* use of italics, as traditionally used in prose, such as to indicate stress, or to indicate terminology or non-English words. For typographic emphasis, you would use `...`; for citations you would use `<cite>...</cite>`; etc.)

All applicable tags should be properly paired. Most tags we are using should be closed, except for `<p>`, ``, and ``.

All of the above means that you may *not* use a “web authoring” program. You must compose the HTML yourself and you will be graded on the actual HTML file, i.e. how it looks when viewed as plain text (e.g. with “cat”).

Please see `http://www.dgp.toronto.edu/~ajr/104/a3/htmlerrors.html` for a list of common errors to avoid.

(continued)

Part 4: Publish your web pages through your CDF account

To publish these two files on the web through the CDF web server, make a subdirectory of your home directory called “public_html” and put them in that subdirectory. This allows you to view the file using the URL `http://www.cdf.toronto.edu/~yourname/filename`.

However, the web server will not be able to read these files yet, because your home directory is set up by default to be unreadable by other users. The web server runs with user permissions, under the special user name “www”.

So part 4 of this assignment is to modify your file permissions as needed to publish your web pages.

First, examine the output of the `ls` command when you give it the option “-l” (the letter ell, for “long” format output). The various fields will be described on a CSC 104 web page, and changing them with the `chmod` command will be described on a web page and/or in lecture or tutorial.

If you specify a directory on the `ls` command line, it shows you the files *inside* that directory. If you want the information about the directory instead, use the “-d” option in addition to the “-l” option. For example, you can see the “ls -l” information for the public_html directory by typing “ls -l -d public_html”.

To make your home directory accessible to the web server, you will want to give it global ‘x’ permission by typing “`chmod a+x .`” when you are in your home directory. The public_html directory will similarly need to be accessible to the web server; type “`chmod a+x public_html`”. For plain files (the HTML files inside that directory), to make them readable by others you should use “`chmod a+r`”.

(If you use the URL `http://www.cdf.toronto.edu/~yourname/`, the web server will reply with the contents of a file called “index.html”, if one exists, or a permission error, if it doesn’t. You don’t have to make an index.html file for this assignment.)

Submission

Parts 1 and 2 must be answered by submitting Gnumeric spreadsheets with the ‘submit’ command. The file names must be datediff.gnumeric and tiles.gnumeric. You must use Gnumeric to produce your files.

The submission commands are similar to those for previous assignments and labs, except that you will use “-a a3” where you used “-a lab4” and “-a a2” and similar. Also remember the “-f” option for resubmission, if needed.

For part 3, you should make the HTML files work at the following URLs (as described in part 4), replacing c2abcdef with your CDF logname:

```
http://www.cdf.toronto.edu/~c2abcdef/surfing.html
http://www.cdf.toronto.edu/~c2abcdef/browser.html
```

You may also, if you like, submit these two HTML files using the ‘submit’ command. I recommend this only if you are having trouble making them visible on the web. **Important note:** If you submit the files using the ‘submit’ command, then those versions will take precedence over the web versions, and the web versions will only be consulted for visibility. So you have to be sure to re-submit them after making any changes. It’s easiest just to make them directly available over the web and not to ‘submit’ them at all.

Hints and other questions and answers

There will be a “questions & answers” web page, which will be updated as more questions come in. Also note a list of common past HTML errors in CSC 104 (see the web page).