

CSC 104 Lab 4, 2 March 2012

This lab contains a component which comprises 3% of your grade in this course. The graded component must be completed by *half an hour after* the end of the lab time you have signed up for. **You must sign up for a lab time on the course web page;** you are not enrolled automatically. Please sign up well in advance.

This lab can be done individually or in pairs. If working in pairs, please make sure that the student who isn't typing is following everything which is going on, and please switch places around halfway through.

Part A: Basic spreadsheet use

Create a Gnumeric spreadsheet which looks like this (a portion of the Canadian income tax form):

| | A | B | C | D | E | F | G | H | I | J | K | |
|----|---|---|---|-----------------------|-------|--------------------|------|-----------|-----------|-----|----|--|
| 1 | Donations and Gifts | | | | | | | | | | | |
| 2 | | | | | | | | | | | | |
| 3 | Total eligible amount of charitable donations and government gifts: | | | | | | | 2000 | | | | |
| 4 | Enter your net income from line 236: | | | | 20000 | times 75%: | | 15000 | | | | |
| 5 | | | | | | | | | | | | |
| 6 | Allowable charitable donations and government gifts: | | | | | | | | | | | |
| 7 | Enter the amount from line 3 or line 4, whichever is less: | | | | | | | 2000 | | | | |
| 8 | Eligible amount of cultural and ecological gifts: | | | | | | | | | | | |
| 9 | | | | | | Add lines 7 and 8: | | 2000 | | | | |
| 10 | Enter \$200 or the amount from line 9, whichever is less: | | | | | | | 200 | times 15% | | 30 | |
| 11 | | | | Line 9 minus line 10: | | | 1800 | times 29% | | 522 | | |
| 12 | | | | | | Add: | | | | 552 | | |
| 13 | This is your donations and gifts deduction amount. | | | | | | | | | | | |

When the “charitable donations”, “net income”, or “eligible amount of cultural and ecological gifts” amounts are changed, the values in cells H4, G7, G9 through G11, and J10 through J12 must automatically be recalculated. That is to say, cells H3, E4, and G8 are inputs, and the rest are outputs.

(Obviously, you can skip on re-typing the text.)

Part B: Day of the year

1. Copy `/u/ajr/104/lab4/starter.gnumeric`, which looks like this:

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| 2 | 31 | 29 | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |

Call your file “`yearday.gnumeric`”.

2. Add a third row using a recurrence formula to calculate the cumulative number of days before a month, so that your spreadsheet now looks like this:

| | A | B | C | D | E | F | G | H | I | J | K | L |
|---|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec |
| 2 | 31 | 29 | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |
| 3 | 0 | 31 | 60 | 91 | 121 | 152 | 182 | 213 | 244 | 274 | 305 | 335 |

except that row 3 must be *calculated*; for example, if the value “29” in cell B2 were replaced by 28 next year, most of the values would change. Try this and make sure it works. (Actually, better would be to replace cell B2 with a formula which correctly calculates the value 28 or 29 in accordance with whether the year in question is a leap year. That’s not our task for today; but what you write today should be

(continued)

compatible with such a future change.)

3. Further below in the spreadsheet, add cells in which you can specify a date, with the abbreviated month in one cell and the day number of that month in another cell. (E.g. “Mar” in one cell and “2” in the next.)

| | A | B | C | D | E | F | G | H | I | J | K | L | |
|---|-----|------------------|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| 1 | Jan | Feb | Mar | Apr | May | Jun | Jul | Aug | Sep | Oct | Nov | Dec | |
| 2 | | 31 | 29 | 31 | 30 | 31 | 30 | 31 | 31 | 30 | 31 | 30 | 31 |
| 3 | | 0 | 31 | 60 | 91 | 121 | 152 | 182 | 213 | 244 | 274 | 305 | 335 |
| 4 | | | | | | | | | | | | | |
| 5 | | | | | | | | | | | | | |
| 6 | | | Date: | Mar | 2 | | | | | | | | |
| 7 | | Day of the year: | | 62 | | | | | | | | | |

4. Next, we will look up the number of days prior to the given month. (This is a portion of the calculation for cell D7 above.) Suppose the value “Mar” is in cell D6. We want a formula which takes that “Mar” and gives us the number 60 (from cell C3), representing the number of days prior to March. So then March 2, for example, will be the 62nd day of the year: 60+2.

To find that “60”, we use the function hlookup(). The first parameter to hlookup() is the value we want to look up—in this case “Mar”—so we will write “D6” (or whatever the appropriate cell name is in your spreadsheet). The second parameter identifies the two rows involved: we want to find the value in row 1, and report the corresponding value in row 3. To do this we specify the rectangle bounded by A1, L1, L3, A3. We specify this rectangle in Gnumeric, as in most spreadsheet programs, with the syntax “A1:L3”. (Some other spreadsheet programs use the syntax “A1.L3”.)

Now when it finds the search object, e.g. for “Mar” it will find it in column C, what do we want hlookup to return for its value? We want it to return the contents of C3. This is the third row of the table A1:L3. So that’s the third parameter to hlookup(): 3, meaning the third row.

The fourth parameter to hlookup() can be TRUE or FALSE. FALSE indicates that we want an exact match; TRUE will find the low end of a range when we are searching in a sequence of numbers and the search object falls between two of them. Right now we want the simpler, exact-match search, so say FALSE.

5. Make sure your formula works so far, testing it with different months. For January it should say 0; for February it should say 31; and so on, according to your table—your hlookup formula will say how many days there are in this year *prior to* the supplied month.

6. This now can be added to the day-of-the-month number (e.g. 2 in March 2) to yield the day-of-the-year number (e.g. 62 for March 2). Remember that hlookup() is a function which yields a value, and thus you have a numeric expression, which can be combined with other numeric expressions in the usual ways. Put an appropriate label in another cell so that the user can tell what is going on, similar to the above picture.

7. Submit your yearday.gnumeric for grading using a command like

```
submit -c csc104h -a lab4 yearday.gnumeric
```

If you are working with a partner, also submit a file named “partner” which says who the other student is. The “partner” file should contain the CDF account name of the other student only (and no other text).

(continued)

Part C: Validating user input in Python (not graded)

Note how the file `/u/ajr/104/lab4/monthname.py` uses a list and list indexing to print the name of a month whose number is supplied. But what happens if you type a month number which is zero, more than 12, or negative? Try it. (Answer: If more than 12, we get an error message; if zero or negative, we get weird and undesirable output.)

Use a ‘while’ loop to make the user type a valid number. A session with your program will look like this:

```
Enter month number: 0
Please enter a number between 1 and 12 (inclusive).
Enter month number: 13
Please enter a number between 1 and 12 (inclusive).
Enter month number: 2
The month is Feb
```

Part D: Sharing files with your partner

If working in a pair, after completing the lab please make sure that both of your accounts contain all of the files of interest for your future reference. You can copy files between accounts (whether on the same computer (e.g. CDF) or different computers over the internet) using the “scp” command.

Examples:

To copy my file named “sludge”, in the current directory of my account I’m logged in to somewhere on the net, to the file `/a/b/c` on the CDF computers using the privilege of your account “c2abcdef” I could type:

```
scp sludge c2abcdef@cdf.toronto.edu:/a/b/c
```

After typing this command (or any of the commands below), it will require c2abcdef’s password before doing the copy.

You can copy multiple files in one command (and with only one typing of password), if the target is an existing directory. If `/d/e/f` on the CDF computers is an existing directory to which you (c2abcdef) have write access, to copy my file named “sludge” *and* my file named “detritus”, both in the current directory, to similarly-named files inside the `/d/e/f` directory at CDF I could type:

```
scp sludge detritus c2abcdef@cdf.toronto.edu:/d/e/f
```

The source files can be either relative or absolute path names, like always.

The target of such a command can also be a relative path name, in which case it is relative to the recipient’s home directory. To copy my file named “sludge” into your existing directory “glop” under your home directory I could type:

```
scp sludge c2abcdef@cdf.toronto.edu:glop
```

To copy my file named “sludge” into your home directory I could type:

```
scp sludge c2abcdef@cdf.toronto.edu:
```

Note the colon at the end.

General note about scp: If none of the arguments has a colon in it, it will do a simple “cp”. So if it doesn’t prompt for a password, see if you’ve omitted the colon and perhaps created a file with the odd name “c2abcdef@cdf.toronto.edu”. (Which you could delete and then try the scp command again.)