University of Massachusetts Dartmouth
Department of Electrical and Computer Engineering

ECE 161 Name: `gencalp3.cpp`
Generate Calendar Due: `See http://ece161.viall.org`

Overview

In this program you will expand on the other calendar project to produce a full year calendar on one page. The calendar should display three columns with 4 rows in each column. The year should be displayed at the top of the page

Problem statement:

Prompt the user for a year. Print out a calendar for the given year. This should repeat until 0 is entered for the year. An example run is on the back of this handout. There is also a zip file of the executable in the projects directory of the website (gencalp3.zip).

General Hints:

Make use of previously written routines, if it makes sense to do so. You may also need to modify some routines that you have previously written.

As discussed in class, you may create a two dimensional char array which forms the "grid" of a page...for example `char g[50][80];` This provides a virtual page with 50 lines and 80 columns (actually 79, as the last column will be for the `'\0'`). If `g[][]` is initialized to all spaces except for the last column of each row, then the `g[][]` array may be output with the code:

```
for (i=0; i<50; i++)
    printf("%s\n",g[i]);
```

I suggest you make `g` a global variable. If you declare a variable outside of all functions (including outside of `main()`) then that variable is accessible by all functions. Often times the use of global variables is frowned upon, but they do have their uses. For this project, you may not have any other global variables.

The major changes in your routines will be where you print information. Instead of printing it using a printf, now you must write it into specific locations in the `g[][]` array. You might want to consider a function along the lines of:

```
writeat(r,c,s);
```

where `r` and `c` are integers that represent the row and column to write into in the `g[][]` array, and `s` is the string to write.

To write a number into the array, this becomes a two step process. First the value must be converted to a string, and then that string must be written into the `g[][]` array. For example:

```
int r=3, c=4, z=5;
char buf[30];
sprintf(buf,"%3d",z);
writeat(r,c,buf);
```

Changes to previous calendar programs

In previous calendar programs, each day occupied 4 columns, which to say that each week occupied 28 columns. If you were to put three months across the page, this would occupy 84 columns...a bit difficult, as there are only 79 columns that you can print in. The 80 column limit is created by the default size of the cmd prompt window. While this can be changed, this problem specifies 79 columns. Hence it is necessary to change the format of each day to be three columns. Day names (Sun Mon Tue Wed Thu Fri Sat) will need to be changed to (S M T W T F S) instead.

Also by default the cmd prompt window is 25 rows high. I want you to change this so you can see the entire calendar. To change the default size of the window, run the program once. Right-click on the small icon at the extreme left of the title bar. Select properties. In the Window size Height box, change the setting to 55 (something a bit larger than 50). Click OK. If a dialog box appears asking if you want this to apply to all windows, click yes. When your program now runs, you will see a cmd prompt window that is 80 columns by 55 rows.

Where to start months

Each month needs to have a designated startrow and a designated startcol. These two values are passed to your new genmonth() routine, and will ultimately be passed to the writeat() routine. For example the upper left corner of January might be "hardwired" at row 10 column 4; December might be hardwired at row 40 column 54.

Credit

The problem as specified above is worth 100 points.

For another 50 points, instead of printing out the year on one line, print it out in bigger numbers made from little numbers... for example

```
 222    000     1     222
2   2  0   0   11    2   2
   2   0   0    1       2
 2     0   0    1     2
22222   000   11111  22222
```

Sample run:

```
Enter year (0 to stop): 2012

                                2012

      January               February                March
 S  M  T  W  T  F  S   S  M  T  W  T  F  S   S  M  T  W  T  F  S
 1  2  3  4  5  6  7                  1  2  3   4                  1  2  3
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   4  5  6  7  8  9 10
15 16 17 18 19 20 21  12 13 14 15 16 17 18  11 12 13 14 15 16 17
22 23 24 25 26 27 28  19 20 21 22 23 24 25  18 19 20 21 22 23 24
29 30 31              26 27 28 29           25 26 27 28 29 30 31

      April                 May                   June
 S  M  T  W  T  F  S   S  M  T  W  T  F  S   S  M  T  W  T  F  S
 1  2  3  4  5  6  7            1  2  3  4  5                  1  2
 8  9 10 11 12 13 14   6  7  8  9 10 11 12   3  4  5  6  7  8  9
15 16 17 18 19 20 21  13 14 15 16 17 18 19  10 11 12 13 14 15 16
22 23 24 25 26 27 28  20 21 22 23 24 25 26  17 18 19 20 21 22 23
29 30                 27 28 29 30 31        24 25 26 27 28 29 30

      July                  August                September
 S  M  T  W  T  F  S   S  M  T  W  T  F  S   S  M  T  W  T  F  S
 1  2  3  4  5  6  7                  1  2  3  4                  1
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   2  3  4  5  6  7  8
15 16 17 18 19 20 21  12 13 14 15 16 17 18   9 10 11 12 13 14 15
22 23 24 25 26 27 28  19 20 21 22 23 24 25  16 17 18 19 20 21 22
29 30 31              26 27 28 29 30 31     23 24 25 26 27 28 29
                                            30

      October               November              December
 S  M  T  W  T  F  S   S  M  T  W  T  F  S   S  M  T  W  T  F  S
    1  2  3  4  5  6                  1  2  3                  1
 7  8  9 10 11 12 13   4  5  6  7  8  9 10   2  3  4  5  6  7  8
14 15 16 17 18 19 20  11 12 13 14 15 16 17   9 10 11 12 13 14 15
21 22 23 24 25 26 27  18 19 20 21 22 23 24  16 17 18 19 20 21 22
28 29 30 31           25 26 27 28 29 30     23 24 25 26 27 28 29
                                            30 31

Enter year (0 to stop): 0
```