CSC 111 Assignment 4

NOTE: Your programs must compile and execute using the CLion environment in ECS 242. If you do your work on your own computer, be sure to test it in ECS 242 before you submit it.

Programming instructions

Assignment 4 consists of two parts. Each part requires a separate C program in a separate .c source file. Upload the syntactically and semantically correct C source files to CourseSpaces. Part I involves reading numerical data. Part II introduces you to file output, HTML5 and SVG using `printf()`.

Part I

Write a C program to interactively convert integers from Arabic numerals (e.g. “14”) to Roman numerals (e.g. “XIV”). Your program should prompt the user to enter a number between 1 and 4999 (inclusive), then print the representation of that number in Roman numerals. After each successful conversion, the program should prompt the user for another number, until the user enters and invalid number (less than 1 or greater than 4999) or enters any non-numerical data (such as “exit”).

A sample-run of a model solution is shown below. Text in black is the output of the program and text in blue is user input.

```
Enter a number between 1 and 4999: 4
The value of 4 in Roman numerals is IV

Enter a number between 1 and 4999: 33
The value of 33 in Roman numerals is XXXIII

Enter a number between 1 and 4999: 61
The value of 61 in Roman numerals is LXI

Enter a number between 1 and 4999: 123
The value of 123 in Roman numerals is CXXIII

Enter a number between 1 and 4999: 789
The value of 789 in Roman numerals is DCCLXXXIX

Enter a number between 1 and 4999: 652
The value of 652 in Roman numerals is DCLII

Enter a number between 1 and 4999: 1234
The value of 1234 in Roman numerals is MCCXXXIV

Enter a number between 1 and 4999: -1
Invalid value entered. Program will exit.
```

For the purpose of this assignment, the definition of Roman numerals will be equivalent to the Wikipedia article on Roman numerals (http://en.wikipedia.org/wiki/Roman_numerals). Note that your program is required to use the “subtractive notation” described in the article (so, for example, the number 4 will be represented by “IV”, not “IIII”).

Your program should use the `fscanf()` function (i.e., not the `scanf()` function) to read integers from the console in the `stdio.h` library to read integer values from the user. Note that `fscanf(stdin, …)` is equivalent to `scanf(…)`. On the line before any call to `fscanf()`, you should add the statement “`fflush(stdout);`” to ensure that CLion displays your input prompt before waiting for input.

Part II

In Assignment 5 you will generate beautiful randomized CSC 111 art as depicted in Figure 1 below. The idea is to generate text files in HTML 5 and SVG format. These files can be viewed in your favorite web browser and can be mailed to your friends and family.

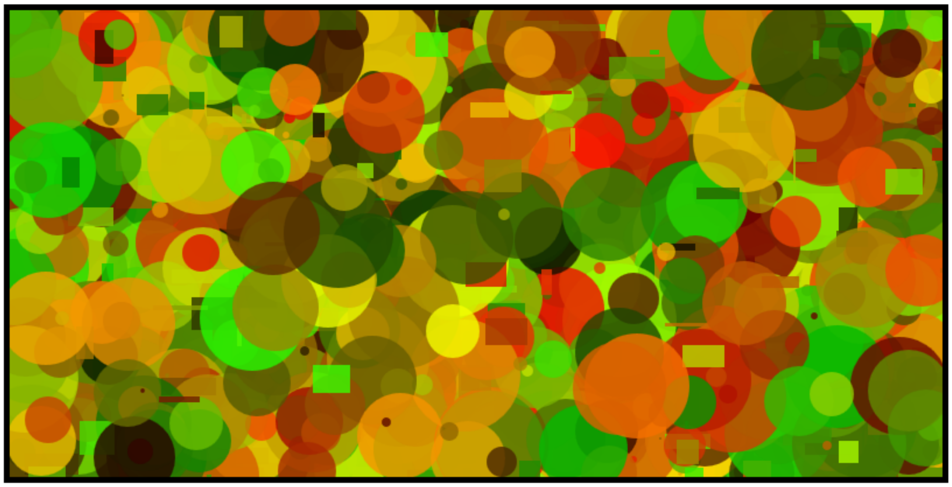

Figure 1: CSC 111 Art

In Part 2 of Assignment 4, you are to generate a simple HTML 5 text file with embedded SVG text. A syntactically correct C program has been provided in the file `A4P2template.c` (posted on CourseSpaces). Study the template file carefully before starting. This template generates the HTML5 file which includes embedded SVG code (cf. Figure 3 below). Figure 2 depicts how this HTML5 file is interpreted by a web browser.

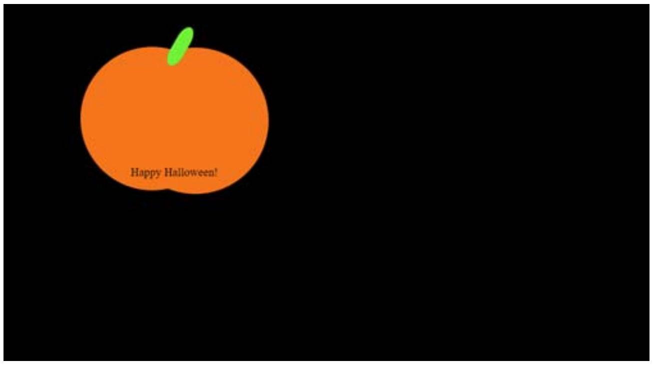

Figure 2: CSC 111 Pumpkin

```
<!DOCTYPE html>
<html>
    <head>
        <title>My CSC 111 Web Page</title>
    </head>
    <body>
    <svg width="1100" height="350">
        <rect x="0" y="0" width="1100" height="350"
              style="fill:rgb(0,0,0)"></rect>

        <g transform="translate(41,7) scale(0.7)">
            <circle cx="150" cy="150" r="100"
                    style="fill:rgb(255,118,25)"></circle>
            <circle cx="210" cy="153" r="102"
                    style="fill:rgb(255,118,25)"></circle>
            <rect x="160" y="-50" height="60" width="20"
                  rx="20" ry="20" style="fill:rgb(25,255,50)"
                  transform="rotate(30 50 50)"></rect>
            <text x="120" y="230" >Happy Halloween!</text>
        </g>
    </svg>
    </body>
</html>
```

Figure 3: CSC 111 Pumpkin HTML5 file generated by A4P2template.c

Modify `A4P2template.c` so that it generates 20 pumpkins of different colours, positions, and sizes. Call the function `writePumpkin()`20 times to render 20 pumpkins. Note that the pumpkin can be moved (i.e., translated) and scaled using the `transform` attribute in the `<g>` tag. Note that these parameters should be in ranges. Use `rand()` to generate integers and floating point numbers for these parameters in ranges. You do not have to produce the exact pumpkin arrangement depicted in Figure 4. Over 15 pumpkins should be at least partially visible.

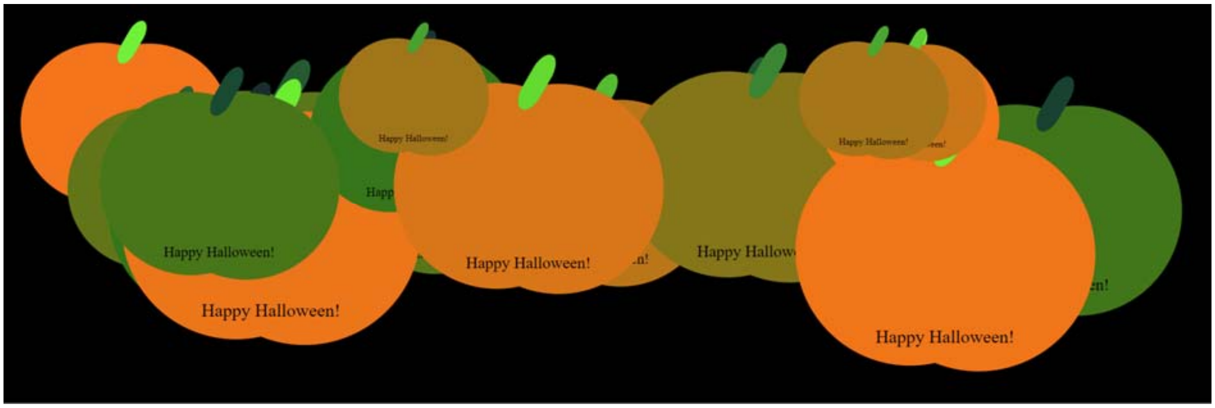

Figure 4: Sample solution for A4P2

Assignment submission instructions

CSC 111 assignments will only be accepted electronically through the assignment page on the CSC 111 CourseSpaces site. Your submission will consist of ***two C source files*** named by the following convention: If your student ID is ***V00123456,*** your C source files for parts 1 and 2 (respectively) must be named **V00123456A4P1.c and V00123456A4P2.c**. In addition, your ***full name, student ID,*** and ***Assignment name*** (e.g., Assignment 4) must appear in a ***comment section at the beginning of each C program***.

For example (in Part 1):

`Name: Polar Bear` **(Replace this with your name)**
`UVicID: V00123456` **(Replace this with your student number)**
`Date: 2017/10/13` **(Replace this with the date you wrote the program)**
`Assignment: A4`
`File name: V00123456A4P1.c` **(Replace V00123456 with your student number)**
`Description: This program reads integers from the user and converts each value to Roman numerals.`

Please submit only the source file and not the executable file. To verify that you have submitted the correct file, you are strongly encouraged to download your submissions from the site and test that they work correctly in your CLion environment. Submissions that do not follow the guidelines above will receive a mark of zero.

Since this assignment only requires three source files, CourseSpaces will only allow you to submit three files. However, until the assignment due date, you may change your submission by deleting and resubmitting your source files multiple times. After the due date, no submissions will be accepted. When grading is complete, your assignment mark and comments will appear in the ***Gradebook*** section of CourseSpaces.