CS 115

Assignment 1 – Due Oct 2nd 7:00pm

A glyph is a 10 × 10 matrix of pixels. Each pixel is a decimal digit (i.e., 0-9). A glyph file is a text file containing 10 rows of integers, each row containing 10 entries. Every entry is an integer in the range of 0-9, indicating the corresponding pixel value. An example is shown below.

```
0 0 0 6 9 9 9 7 0 0
0 0 6 0 0 0 0 0 7 0
0 9 0 0 8 0 8 0 0 9
0 9 0 0 4 0 4 0 0 9
0 9 0 0 0 0 0 0 0 9
0 9 0 0 0 2 0 0 0 9
0 9 0 0 0 0 0 0 0 9
0 7 0 0 0 3 0 0 0 6
0 0 7 0 0 0 0 0 6 0
0 0 0 7 8 8 8 6 0 0
```

It is intended a glyph represents an image (of very low resolution). Each pixel value indicates the "color" of that pixel. Let us name this glyph file example.glyph.

1. [20%] In the following questions, you should put all shared code into a separately compiled .cpp file, complete with an appropriate header file to facilitate reuse.

2. [20%] Implement a C++ filter function **render** that takes a glyph stored in an array, and outputs to the screen the glyph as an ASCII art. Specifically, each glyph should be rendered as 10 lines of 10 characters, with each row represented by a line of 10 characters, and each pixel represented by a single character. The conversion from a pixel value to a character is specified as follows:

| Pixel | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| Character | . | o | O | * | # | + | / | \ | - | \| |

For example, example.glyph should be rendered into the following lines.

```
.../|||\..
../.....\.
.|..-.-..|
.|..#.#..|
.|.......|
.|...O...|
.|.......|
.\...*.../
..\...../.
...\---/..
```

If you look carefully, you can tell that the above glyph represents a clown face (somewhat shifted to the right).

Use formatted I/O to read and output glyphs.

Do NOT use 10 nested if-statements to perform conversion from pixel values to characters. There is a much more elegant way to achieve this. (Hint: Think arrays.)

3. [15%] The transpose of a glyph is obtained by the following steps:

   a. Swap the pixels at the opposite sides of the diagonal that extends from the upper left corner to the lower right corner of the glyph.
   b. Convert every pixel with a value 8 (corresponding to the character '-') to the value 9 (corresponding to the character '|'), and vice versa.

For example, the following glyph is the transpose of example.glyph.

```
0 0 0 0 0 0 0 0 0 0
0 0 8 8 8 8 8 7 0 0
0 6 0 0 0 0 0 0 7 0
6 0 0 0 0 0 0 0 0 7
8 0 9 4 0 0 0 0 0 9
8 0 0 0 0 2 0 3 0 9
8 0 9 4 0 0 0 0 0 9
7 0 0 0 0 0 0 0 0 6
0 7 0 0 0 0 0 0 6 0
0 0 8 8 8 8 8 6 0 0
```

The ASCII-art rendering of the transposed glyph looks like this:

```
..........
..-----\..
./......\.
/........\
-.|#.....|
-....O.*.|
-.|#.....|
\......../
.\....../.
..-----/..
```

Implement a C++ filter function transpose that takes a glyph stored in an array, and generates the transpose of the given glyph stored in the array.

4. [15%] The mirror of a glyph is obtained by the following steps:

   a. Swap the pixels at the opposite sides of the vertical axis between the 5'th and the 6'th columns (i.e., in the middle).
   b. Convert every pixel with a value of 6 (corresponding to the character '/') to the value 7 (corresponding to the character '\'), and vice versa.

For example, the mirror glyph file of example.glyph is shown below.

```
0 0 6 9 9 9 7 0 0 0
0 6 0 0 0 0 0 7 0 0
9 0 0 8 0 8 0 0 9 0
9 0 0 4 0 4 0 0 9 0
9 0 0 0 0 0 0 0 9 0
9 0 0 0 2 0 0 0 9 0
9 0 0 0 0 0 0 0 9 0
7 0 0 0 3 0 0 0 6 0
0 7 0 0 0 0 0 6 0 0
0 0 7 8 8 8 6 0 0 0
```

Here is the corresponding ASCII art.

```
../|||\...
./.....\..
|..-.-..|.
|..#.#..|.
|.......|.
|...O...|.
|.......|.
\...*.../.
.\...../..
..\---/...
```

Implement a C++ filter function mirror that takes a glyph stored in an array, and generates the mirror of the given glyph stored in the array.

5. [15%] To rotate a glyph is to shift all dots up one row, and replace the last row with the first row of the original glyph. For example, rotating example.glyph once gives us the following glyph.

```
0 0 6 0 0 0 0 0 7 0
0 9 0 0 8 0 8 0 0 9
0 9 0 0 4 0 4 0 0 9
0 9 0 0 0 0 0 0 0 9
0 9 0 0 0 2 0 0 0 9
0 9 0 0 0 0 0 0 0 9
0 7 0 0 0 3 0 0 0 6
0 0 7 0 0 0 0 0 6 0
0 0 0 7 8 8 8 6 0 0
0 0 0 6 9 9 9 7 0 0
```

```
../.....\.
.|..-.-..|
.|..#.#..|
.|.......|
.|...O...|
.|.......|
.\...*.../
..\...../.
...\---/..
.../|||\..
```

Rotating the resulting glyph again gives us the following.

```
0 9 0 0 8 0 8 0 0 9
0 9 0 0 4 0 4 0 0 9
0 9 0 0 0 0 0 0 0 9
0 9 0 0 0 2 0 0 0 9
0 9 0 0 0 0 0 0 0 9
0 7 0 0 0 3 0 0 0 6
0 0 7 0 0 0 0 0 6 0
0 0 0 7 8 8 8 6 0 0
0 0 0 6 9 9 9 7 0 0
0 0 6 0 0 0 0 0 7 0
```

```
.|..-.-..|
.|..#.#..|
.|.......|
.|...O...|
.|.......|
.\...*.../
..\...../.
...\---/..
.../|||\..
../.....\.
```

Implement a C++ filter function rotate that takes a glyph stored in an array, and generates the rotation of the given glyph stored in the array.

6. [15%] In your main.cpp file use your functions transpose, mirror, rotate, and render to create a program that takes example.glyph as input, and outputs on the screen the following ASCII art:

```
../-----..
./......\.
/........\
|.....#|.-
|.*.O....-
|.....#|.-
\......../
.\....../.
..\-----..
..........
```

**Submission Requirements**

You will submit your assignment to UR Courses before 7pm on Oct 2nd, 2018.

You will submit 3 files…
glyph.h
glyph.cpp – containing glyph-manipulating filters from points 1-5
main.cpp – outputs ASCII art outlined in point 6