TCP1101 Programming Fundamentals
Trimester 1, Session 2019/2020
Faculty of Computing and Informatics
Multimedia University

ASSIGNMENT Milestone 2

Submission Deadline

25 Sept 2019, Wednesday, 11:59PM, GMT+08 (following MMLS timestamp)

Individual Assignment

This is an individual assignment. STRICTLY NO COPYING from other sources except codes given in this course. If detected all parties involved will get 0 marks.

Assignment - Milestone 2

The users of the display board program have sent you further requests to improve the program. There are two major improvements that need to be incorporated into your program from Milestone 1.

I. External input file commands

To ease the use of the program (…entering input one by one is really tiring! ☹), modify your program to allow reading of commands from an external input file. Users should now be able to write a series of text commands, which are then input into the program to display a specific design of the scrolling banner. For ease of use, the program should ask for the input file name to process, proceed to perform the display, and return back for another round of user input when it ends. A sentinel text can be used to exit program.

The following table summarizes the commands and the order of commands, which are to be used in the text file. Note: For more information of what some of these features do, please refer to Section II ("New features").

Sample text file (`sample.dat`)	Description for each line

`#`
`5`
`Selamat Hari Merdeka`
`6 18`
`20 40`
`lr wr rot90`
	Drawing character (optional, defaults to '#')
	Speed of refresh cycle
	Phrase (max 25 chars), input can be of any case
	Anchor point
	Board size (optional, defaults to 20, 40)
	Direction: `lr`, `rl`, `ud`, `du`, `st`
	Wrap-around: `wr` (optional)
	Rotation: `rot90`, `rot-90`, `mr` (optional)

II. New features

The users have requested for further new features that will enhance the board display. Please note that you *may* have to make substantial changes to your original program in Milestone 1 (this varies from case to case). Among the requested features are the following:

1. **Different characters for drawing**
   Previously, you may have fixed the character used for drawing. Now, users should be allowed to specify their choice of characters for the printing. The first character in the text file indicates this. Default symbol if not specified, is a '#' character.

2. **Variable board size**
   Board sizes can now be specified by the user. The two integers (in correct order) indicate the number of rows, and the number of columns, respectively. Of course, they must still be two positive integers. There should be no limit to the dimensions of the board. Unless specified, the board size will be (20, 40) as defined in Milestone 1.

3. **Scrolling wrap-around effect**
   There are four ways of scrolling the string: left to right (`lr`), right to left (`rl`), up to down (`ud`), down to up (`du`). You are to implement the wrap-around effect while performing the scrolling. When the string scrolls beyond the edges of the board, the part of the string pushed out will enter from the opposite edges, thus the "wrap-around" effect. This feature was postponed from Milestone 1. The user can also choose to have no scrolling effect, making the string stationary (`st`).

4. **Rotated display**
   Allow the user to apply rotation to the string. There are three possible modes: Rotation by 90 degrees (clockwise) `rot90`, Rotation by -90 degrees (anti-clockwise) `rot-90`, and Mirror (vertical reflection, or flipping the object across the x-axis) `mr`.

   Note 1: At each time, *only one* mode is applied.

   Note 2: Rotations are applied to the *entire string* based on the *rotation point*, not to each character. The *rotation point* is defined as the bottom-left corner of the string.

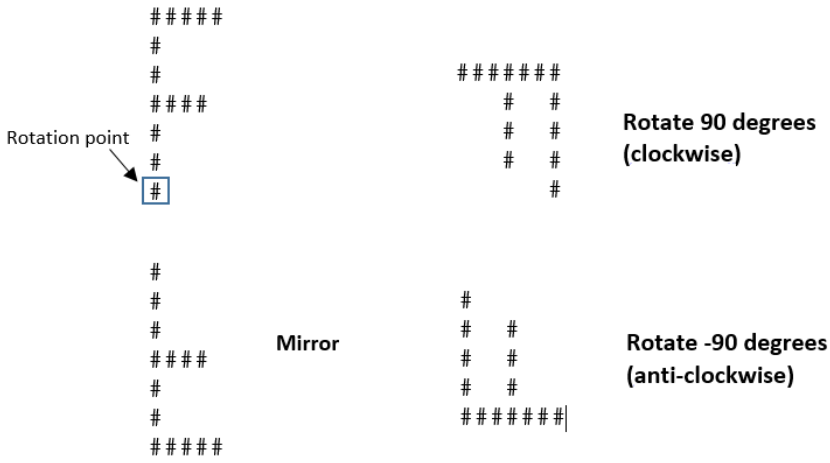

Fig. 1: Three ways to rotate

Other implementation points and caveats to note:

- Show that you are able to use functions appropriately in your code.
- Speed of refresh cycle must be a number from 1-10 (as specified in Milestone 1)
- Characters in the phrase is not case sensitive. However, your board should only display uppercase. The phrase should not be more than 25 characters in length (space included).
- For cases where a rotation command is issued:
    - The anchor point remains the same top-left-most position of the string (see Fig. 2 below).
    - If scrolling is on, the scrolling direction is taken with respect to the board, not the rotated string.
    - Hard-coding the rotated characters in source code is not permitted. No marks will be awarded.
- The last two commands (wrap-around and rotation) can be in any order after the scrolling command, since they are optional.
- Consider carefully, all that necessary input validation that needs to be included into your program to ensure the user is properly informed of any incorrect input(s), or format/value of input(s).

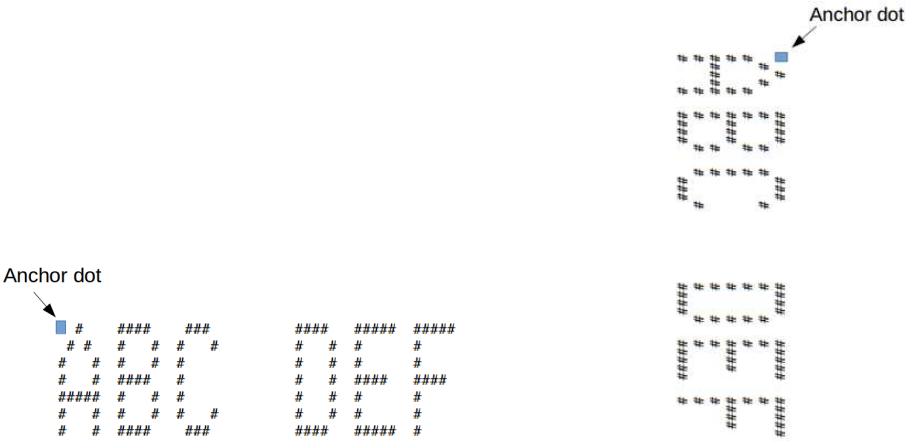

Fig. 2: The anchor dot of a string in (left) normal, and (right) rotated (90°) orientation

III. Bonus features (**Optional**, but bonus marks can be awarded)
To implement these new features, do define additional commands to the input instructions to enable these to work.

- **"Inception" Effect**
  Each character in the string is drawn using its own character. You may replace the first command (drawing character) to something else.
- **Flash Show**
  A single (25-char) phrase is not enough! What if we have an entire message to show? Allow the user to add multiple sets of commands, consecutively to show a long message.
  Note: Each scrolling animation (for each phrase) should end before showing the next. A delay can be specified between each phrase.

Some similar rules as with Milestone 1

- Use only platform-independent standard C++ libraries to implement this milestone. However, if necessary, the use of `system("clear")` or `system("cls")`is permitted while including `windows.h` is not permitted.

- To ease plagiarism checking, put your source code in only one file and use the following file name format:
  `yourLabSection.StudentID.NameInTheMMUNameList.cpp`

  For example, name your source file as `TT01.1171777777.Tony.Gaddis.cpp`.
  0.5 mark will be deducted from a submission which does not follow this format.

Handy advice

- If you are unable to do a certain feature does not mean that the other features cannot be implemented. Most features are independent of one another. Try to work first on the features that you can easily implement.

- Be careful with the parsing of the input file. Some commands are optional, some are mandatory (not stated as optional).

- Your program could potentially be checked by the evaluator using an input file that has not been revealed to you. So, do test your program thoroughly.

Deliverables

a) Source code in one file (For example: `TC01.1171777777.Tony.Gaddis.cpp`).

b) Provide 2-3 sample input text that you have designed to showcase what your program can do.
c) Optionally, if necessary, provide a text file containing any additional bonus features/special commands that you have created beyond the features described in this Assignment.

Additional Info on Deliverables

a) Source codes have to be properly formatted and documented with comments and operation contracts. Executable files need not be submitted.

b) For ALL your files, insert the following information at the top of the file:

```
/**********|**********|**********|
Program: YOUR_FILENAME.cpp
Course: Programming Fundamentals (TCP1101)
Year: 2019/20 Trimester 1
Name: Frank Carrano
ID: 1071001234
Lecture Section: TC101
Tutorial Section: TC201
Email: abc123@yourmail.com
Phone: 018-1234567
**********|**********|**********/
```

Soft-copy submission instruction

a) Create a folder in the following format:

TUTORIALSECTION_ASSIGNMENT_FULLNAME

For example, if your name is Frank Carrano, you come from TC201 tutorial section, and you are submitting Milestone 2, then your folder name should be "TC201_M2_FRANK_CARRANO" without the double quotes.

b) Place all files for submission into the folder.

c) Zip your folder to create a zip archive (e.g. TC201_M2_FRANK_CARRANO.zip).

d) Submit your assignment zip file through MMLS before the given deadline.

e) You can still submit after the given deadline, you will be deducted a late penalty of *20% (of Milestone 2 total)* for every additional day late. No submissions will be accepted after the 5th day late

Evaluation Criteria

Mark Sheet

Criteria	Max	Actual Marks
Milestone II		
User interface	1	
Old features: Speed, Anchor point, String printed in uppercase	2	
Feature: Characters for drawing	1	
Feature: Variable board size	1	
Feature: Scrolling + Wrap-around effect	2	
Feature: Rotations*	3	
Input validation	2	
Appropriate usage: Functions	1	
Appropriate usage: File reading	1	
Error-free compilation & runtime	1	
Good coding style	0.5	
Sample input files provided	0.5	
Total	15	
Bonus Features (max. 2)		
Grand Total		

* Rotated letters that are hard-coded in code are not permitted (0 marks)

Each feature will be evaluated based on fulfilment of requirements, correctness, compilation without warnings and errors, error free during runtime, basic error handling, quality of comments and operation contracts, user friendliness, and good coding format and style.