National University of Computer and Emerging Sciences, Islamabad

FAST School of Computing Fall 2022

Due Date: 6-Nov-2022

Data Structure

Assignment No. 2

The purpose of the assignment is to put your knowledge of stack and queue structures to the test. This assignment will take more time to complete than the previous. Begin your assignment right away! You will require time to understand the assignment and answer the numerous questions that may emerge as you read the description and done with coding.

A compiler convert source code into an object program. The parser is required by most compilers to read the input file. Your first objective will be to parse a file and then complete the tasks specified in the file. The assignment is divided into 4 different parts:

1. **Parsing the file**

   The file will be in txt format, and you are restricted to use the stack to read the file. There will be different tags in file, each tag have specific meaning like Start, head, paragraph, pre_exp, sol_exp, post_exp, function, and read_image.

   Every tag has a syntax for open and closing like start tag will be start with |start| and end on |\start|. Same syntax will be followed for the other tags.

   Tag explanation:

   a. Start→ Indicates the starting and ending of document.
   b. Head→ Indicates the section header, you have to print it on the top.
   c. Tab.→ insert tab.

```
|head|
                |tab| |\tab| |tab| |\tab| |tab| |\tab|First File.
        |\head|
```

Output:

First File

d. Priority→ This tag contains the document priority.

```
|priorty|
        21
|\priorty|
```

e. Paragraph→ Contain the text that you have to print, but there can be multiple tags inside the paragraph.

|paragraph|

*The file getting processed and the converted postfix notation will |post_exp| a + b * c |\post_exp|.*

*One more statement to test your coding skills to handle the character arrays without using any built-in function, so show this expression in to prefix form |pre_exp| a * b / (c + d) ^ f - g |\pre_exp|.*

The skill test doesn't stop there, you have to evaluate on more statement to prove your skills on basic stack working.

Evaluation result of provided string is |sol_exp| abc+|\sol_exp|.*

|\paragraph|

Output:

The file getting processed and the converted postfix notation will abc+.*

*One more statement to test your coding skills to handle the character arrays without using any built-in function, so show this expression in to prefix form -/*ab ^+cdfg.*

The skill test doesn't stop here, you have to evaluate one more statement to prove your skills on basic stack working.

*Evaluation result of provided string is a+b*c.*

f. Pre_exp→ There will be infix expression inside the tag and you have to convert this statement into prefix form and print on the place of infix expression.
g. Post_exp→ There will be infix expression inside the tag and you have to convert this statement into postfix expression and print on the place of infix expression.
h. Sol_exp→ There will be an expression inside the tag and you have to evaluate this statement print on the place of infix expression.
i. Function→ This tag contains the function. You have to compile the function and show the output.
i. Read_image→ This tag is based on your previous assignment. This tag will contain the image source path and you have to read that image. Compile image and find the objects in region, also save the new images generated by your algorithms. Explain in task 4.

2. Error Detection

There can be missing tags or invalid expressions, so you have to detect the parsing errors. There will be many tags other than those mentioned above. So, just consider these tags as the starting and ending boundaries of these tags, not more than that. While solving the expression, if you find any invalid expression, print "Invalid Expression" instead of the expression. If you find any missing or extra tags in the file, print Syntax Error.

> *|paragraph|*
> *So, you said you enjoyed the previous assignment and learned much and more :). Now, tell me about the number of galaxies in the picture. There are |src| \path |\src| galaxies in the image. |\paragraph|*

Output:

So, you said you enjoyed the previous assignment and learned much and more :). Now, tell me about the number of galaxies in the picture. There are 4 galaxies in the image.

3. Least recent unit (LRU):

You have to maintain the queue with a fixed size of 4. In which you will store the file name on the base of their priority. When the queue gets full and you need to delete the item from the queue, you will delete the least recently used item in the queue.
When you start reading the file, you will check the file in the queue. If a file exists in the queue, print File Hit; else, File Fault. At the end of the program, you have to print the total number of hits and faults.

4. Extracting the galaxies (Extend part of Assignment 1)

In previous assignment you have learn the following things:

1. Read the image.
2. Altering the image.
3. Detecting the number of objects in images.

Now in part you have to reuse the previous function and have to generate separate image for each object. (No Screen Shot, save the image properly using openCV).

You are given an image of size 512x512. The image contains clearly distinct objects. The background is BLACK represented by pixel value 0 and objects are displayed in white represented by 255. The goal is to separate each object from the rest of the image. The object can be separated by rest of the image by traversing the image either ROW wise or Column wise. The separated object should be moved to a blank new image. Where all the regions will be black except the area of separated object.

Algorithm:

Find the starting point of the object. Starting point will the first pixel that is 255 and from this point the whole object originates. Each pixel has 8 neighbors. Top, bottom, left, right and 4 diagonal pixels. Not all pixels have same value. The focus of algorithm will be on those pixels which are 255 to find the whole object.

Method 1:

In row wise traversal, you will use queue to process the pixels. You will create a blank image of the same size as input image. You will start from starting pixel and add the location to the QUEUE. If the queue is not empty, you will remove a pixel location and put 255 in place of that pixel location in new blank image. As soon as you remove a pixel location from queue, you will check all its neighbors that are 255. All those locations will be added to the QUEUE if they already haven't been processed from the queue. If the pixel has already been processed by the queue, it will not be added to queue again. When the queue is empty the new blank image will have the object separated from rest of the objects in the input image.

Method 2:

In column wise traversal, you will use stack to process the pixels. You will create a blank image of the same size as input image. You will start from starting pixel and add the location to the stack. As long as the stack is not empty, you will remove a pixel location and put 255 in place of that pixel location in new blank image. As soon as you remove a pixel location from stack, you will check all its neighbours that are 255. All those locations will be added to the stack if they already haven't been processed from the stack. If the pixel has already been processed by the stack, it will not be added to stack again. When the stack is empty the new blank image will have the object separated from rest of the objects in the input image.

NOTE: wherever the term image is referred, it means array as focus of data structure is on array. And term pixel location refers to [i][j] indexes of array.

Honor Policy

This assignment is a learning opportunity that will be evaluated based on your ability to think in a group setting, work through a problem in a logical manner. You may however discuss verbally or via email the assignment with your classmates or the course instructor, but you are to write the actual code for this assignment without copying or plagiarizing the work of others. You may use the Internet to do your research, but the written code should be your own. **Plagiarized reports or code will get a zero**. If in doubt, ask the course instructor.

How to start

You are provided with one .cpp file and three folders named: **Test_files**, **Output_files** and **Images**. Test file folder contain the test files. When you run the code, the output should be saved in the .out file, which will be saved in the output files folder. There is one images folder that contains images. . Each folder includes various example files for further understanding. Start your code by getting the name of all files stored in Test_files and process each file in loop.

Note: Only one output file will be generated after final execution of program, mean output of all test files will be combined into single file and in the end of file you will show the total **hits** and **faults**.
Your output will be look like this.

First File.

The file getting processed and the converted postfix notations are abc*+ and Invalid Expression.
One more statement to test your coding skills to handle the character arrya without using any builtin funciton, so show this expression in to prefix form -/*ab ^+cdfg.
The skill test doesn't stop there, you have to evaluate on more statement to prove your skills on basic stack working.
Evalaution result of provided strings are a+b*c.

So,you said you enjoyed the previous assignment and learned much and more :). Now, tell me the about the number of galxies in the picture.
Total galxies in picture are 4.
this is just check your stack handling skills. so i hope you will pass this test also
so keep going. So, now I am |\tick| tried of this thing. I think that's enough.

Second File

.
.
.
.
.
.

Third File

.
.
.
.

LRU Results: Total Hits:?, Total Faults:?.

Submission

Assignments are to be done individually. No late assignments will be accepted. **Submissions that do not comply with the specifications given in this document will not be marked and a zero grade will be assigned.** Write your name and e-mail id in a comment line in on top of each source file. You are required to submit a single zip file containing an archive of your code files on Google Classroom. You should name your zip as i21-XXXX.zip where i21-XXXX represents your student id.

Distribution of Marks

The assignment will be evaluated on the basis of your performance and errors. The general distribution will be as follows:

1. Tag Parsing (25)
2. Notations (20)
3. LRU (20)
4. Image object detection (10)
5. Saving Extracted Images (25)

Best of Luck