NOVEMBER 12, 2022

DATA STRUCTURES

ASSIGNMENT MODULE 1

IBRAHIM AKHTAR
21I-1679
DS-N

Introduction

Welcome reader, to the introduction of my first assignment for the course of Data Structures. Through this PDF you will experience the thought process of an average programmer during development of any software.

It will be brief as well as in-depth view of what problems I faced and how I managed to overcome them. Furthermore, it will also include a brief peek into my brain and how I tackle problems, be it programming or any other task at hand.

In this file you should find the following files.

Source.cpp

Array.cpp

Array.h

Queue.h

Queue.cpp

Stack.h

Stack.cpp

Log (may not be that functional)

Readme.txt

Without further delay let us officially start the Project Report from the page that follows.

Problems and Solutions

Problem 1: Laziness

The most common problem that I think most of us went through in this assignment was laziness. Given that we were all swamped with other assignments as well as the pressure of sessional exams, the weight of all that was too much at the start and I failed to get an early start in this assignments solution.

<image proof in the Problem 6 problem description>

Solution:

The solution was simple, I simply had to bore myself from everything else that the only useful thing I had left was completing this assignment.

In addition, learning that this assignment would ultimately become my semester project, motivation was flowing in me like water at Niagara Falls. Fear did play a small role but once that helped me start, hours went by like minutes because of how much fun I had while doing this assignment.

Problem 2: Array or Linked List?

Up next was the problem of deciding the data structure that was best suited to my needs. This problem took the least amount of time to solve.

Solution:

The solution was quite simple, I just had to read the question completely. Once read, it wasn’t even my choice as we had to go with Arrays.

Problem 3: 1-D or 2-D

Once I figured out that we had to go with Arrays, the next question was do I go for a 2-D implementation or a 1-D implementation?

Solution:

1-D and 2-D both seemed simple at the time so I went with a 1-D implementation which seemed quite tedious going forward. I ultimately ended up going with a 2-D implementation because the access of array using an ‘i’ and ‘j’ value seemed way too good and way too easy to pass up on.

The code of this implementation can be found in the constructor definition of the Picture Class in Array.h and Array.cpp.

Problem 4: Templates

The issue of templates was way too persistent and ended up taking a lot of time.

Solution:

I ended up hardcoding the implementation for an integer type array because I simply gave up. I did fix the problem for a lot of functions but the function that would return the class type object or reference did not want to exist in the template implementation which was the final nudge for me to abandon the template implementation.

Problem 5: T.I.M.E A.K.A “The Log File”

A future note to myself, Visual Studio does not like a lot of functions like localtime() and locatime_r(). This was the most annoying problem I have ever faced in all the programming I have done over the years.

The time functions refused to work and they still don’t. I am making a log file but either my processes are too quick or my time is set 0 permanently.

A> Localtime() is not a function that VS likes because it is unsafe according to it.
B> Time is a very tricky thing to deal with in cpp. Python takes the cake when it comes to anything to do with time.

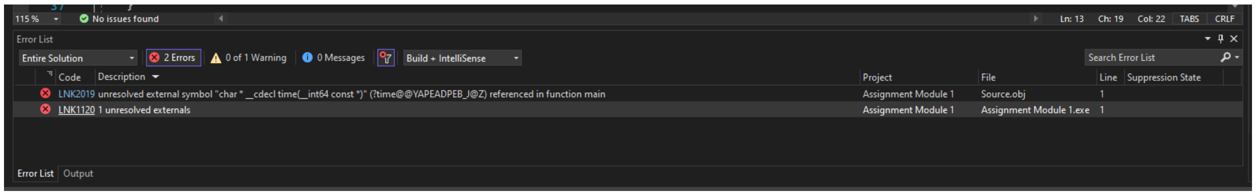
Solution:

I resorted to the Microsoft version of the function that annoyingly switches the two parameters of the localtime() and the localtime_r() functions.

This Microsoft function is localtime_s() that takes the first argument as the tm structure and the second argument as the time_t variable.

All in all, this was very annoying because it took my 4 hours to read up the entire ctime library to figure out what exactly im doing wrong.

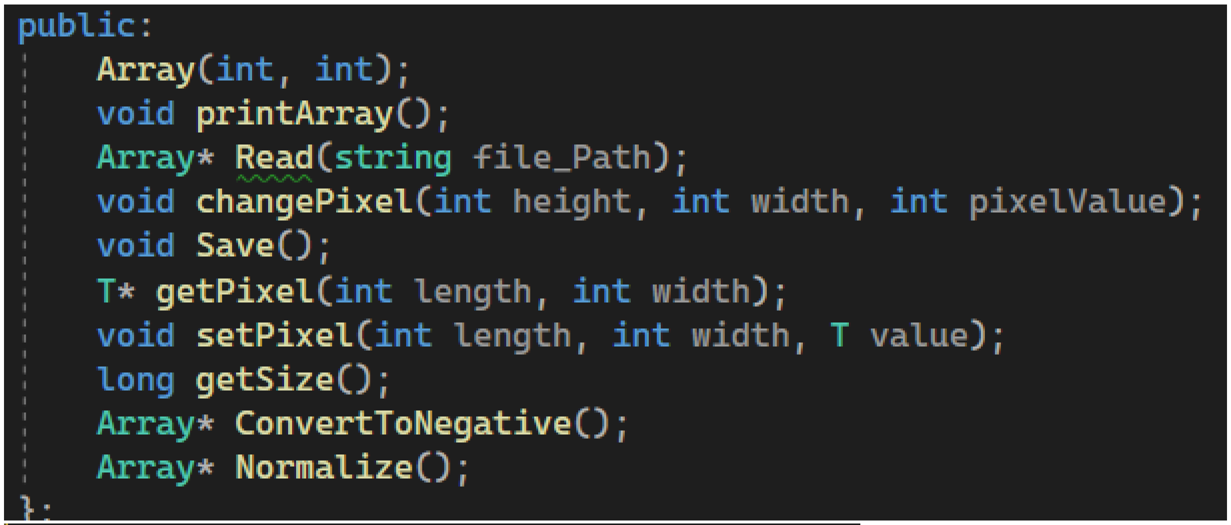
I was too frustrated to take any screenshots other than this one:



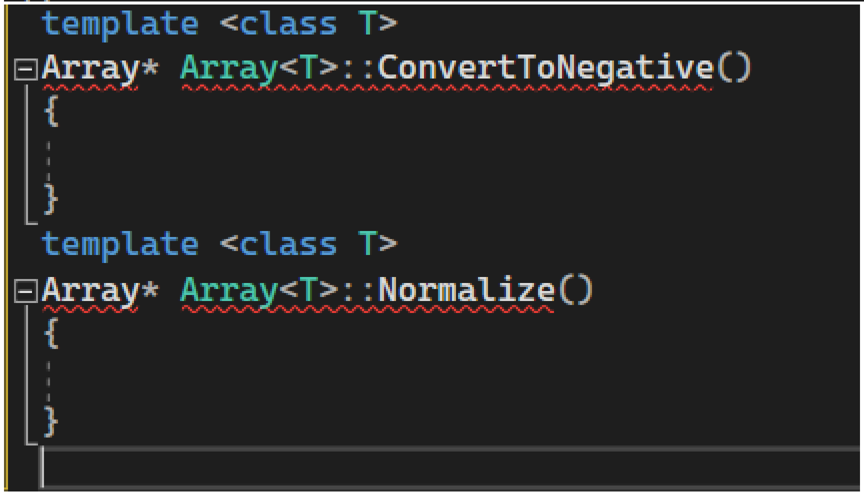
```
115 %   No issues found                                          Ln: 13  Ch: 19  Col: 22  TABS  CRLF
Error List
Entire Solution   2 Errors   0 of 1 Warning   0 Messages   Build + IntelliSense        Search Error List
   Code     Description                                                                                                Project               File                      Line  Suppression State
   LNK2019  unresolved external symbol "char * __cdecl time(__int64 const *)" (?time@@YAPEADPEB_J@Z) referenced in function main   Assignment Module 1   Source.obj                1
   LNK1120  1 unresolved externals                                                                                     Assignment Module 1   Assignment Module 1.exe   1
Error List  Output
```

Problem 6: A Minor STL overlap

I forgot that Array is an actual class that C++ has and accidentally made the entire class called Array* which is why the Picture implementation is in the Array.h and Array.cpp files.



```
public:
    Array(int, int);
    void printArray();
    Array* Read(string file_Path);
    void changePixel(int height, int width, int pixelValue);
    void Save();
    T* getPixel(int length, int width);
    void setPixel(int length, int width, T value);
    long getSize();
    Array* ConvertToNegative();
    Array* Normalize();
};
```



```
template <class T>
Array* Array<T>::ConvertToNegative()
{

}
template <class T>
Array* Array<T>::Normalize()
{

}
```

Solution:

The solution was pretty simple. I changed every Array to Picture and every Array* to Picture* but did not change the file names as a reminder to never make that mistake again.

Problem 7: Question 2 and Question 3

I have no idea what to do in the last questions when it comes to actually coding them so ill leave them as is for now.

Solution:

Not-Found.

Conclusion

Ultimately I had fun even though I did not understand anything from Question 2 and Question 3. I do know how to do it theory but could not understand at all what to do in the code terms. I did store the entire images in the Queue and Stack but failed to do much more.

One thing I would like to ask of the reader is that test and try to break the interface of the program because I built it to be extremely user friendly at every stage.