Programming Assignment 3  
LRU Buffer Pool

Implement a disk-based buffer pool class based on the LRU buffer pool replacement strategy.

This assignment comes from Project 8.3 on page 308 of your text. Using the supplied C++ files to implement an LRU Buffer Pool. I have made the following changes to Project 8.3:

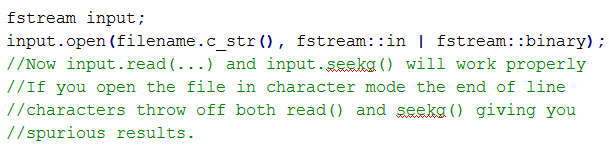
* Implement a BufferBlock class using the supplied BufferBlockADT.h
  + Your Buffer Block must inherit BufferBlockADT or you will not get credit for your work.
  + All BufferBlockADT virtual functions must be implemented in BufferBlock
  + Block Size: 4096
  + The book says you have to use the first 4 bytes of the buffer block to store the block ID. I do not require that. You may add an instance variable to your buffer block implementation to store the block id; i.e., “int blockID.”
* Implement a Buffer Pool by inheriting BufferPoolADT (BufferPoolADT.h) – implement all of BufferPoolADT’s functions (if you do not inherit BufferPoolADT you will not get credit for your work).
* Your buffer pool should consist of 5 buffer blocks
* Your buffer pool should manage the buffers using the LRU strategy
* Your buffer pool should be named LRUBufferPool and the file containing the LRUBufferPool class should be named LRUBufferPool.h
* Use the provided main.cpp and the included test file mydatafile.txt to test your program.

DO NOT MAKE ANY CHANGES TO THE ADT FILES! The ADT files you submit with your assignment must match EXACTLY the files I have provided, or **you will not get any credit for the assignment**.

## File Provided:

* BufferBlockADT.h
* BufferPoolADT.h
* constants.h (contains both constants and test function definitions)
* main.cpp (test program driver)
* mydatafile.txt (input file for the test run) – DO NOT COPY the contents of this file into a new file. You must move the original file into your project.
* Buffer Pool Output.txt – if your program is correct the output for the first three calls to getBytes() should look like this.

## Note: You must open the input file in binary mode for the read() and seekg() methods to work properly. See example code below.



## Testing

Use main.cpp as the test driver and mydatafile.txt as the input file.

## Assignment Submission:

Put your Visual Studios 2022 project folder into a zip file and submit your assignment to the assignment link in Canvas along with the following:

* Approach document describing your approach to solving (day-by-day plan for that day, future tense, and at least 500 words in length) this problem and any approvals I’ve given to change files I’ve provided.
* Screen shots or an output text file with the results of your test run. Make it clear which file this is in your zip file submission.

## Rubrics:

* **Program must run in order to get any points**

|  |  |
| --- | --- |
| **Requirement** | **Value** |
| Approach | 5% |
| BufferBlock |  |
| getData | 30% |
| LRUBufferPool |  |
| getBytes |  |
| Read data in buffer | 10% |
| LRU Read from disk | 35% |
| Last Read Block to front | 10% |
| printBufferBlockOrder | 5% |
| getLRUBufferBlockID | 5% |
| **Total** | **100%** |

[Course Learning Outcomes Alignment](https://www.dropbox.com/s/cg01sxds3tgyamq/5.%20Sorting%20%26%20File%20Processing.docx?dl=0)