

Branch: **master** ▼

Find file

Copy path

wsu / csc5050 / ass3 / README.md



borodark links in README

8980f78 3 minutes ago

[1 contributor](#)**Raw** **Blame** **History**

50 lines (26 sloc) 3.74 KB

CS5050 Assignment 3

To run

Developed and tested in Mac OSX

Install `cmake` and `make` , pull this repo change to this directory, run:

```
cmake . to create Makefile
```

```
make to comple
```

Q 1,2,3 4

source code:

- Q1 - [src/q1.cpp](#) - driver and implementation
 - Results: [q1.out](#)
- Q2 - [src/q2.cpp](#) - driver and implementation
 - Results: [q2.out](#)

- Specify and implement an *ADT for fractions*: see [Fraction Header](#) [Fraction Implementation](#). Provide operations that [add](#), [subtract](#), [multiply](#), and [divide](#) these numbers. The results of all arithmetic operations should be in lowest terms, so include a private function [reduceToLowestTerms](#). To simplify the determination of a fraction's sign, you can assume that the denominator of the fraction is positive. Next, Specify and implement an ADT for mixed numbers: see [MixedNumber Header](#), [MixedNumber Implementation](#) , each of which contains an integer portion and a fractional portion in lowest terms. You can utilize the ADT fraction you created in the first part of this problem. Provide operations that [add](#), [subtract](#), [multiply](#), and [divide](#) mixed numbers. The results of all arithmetic operations should have fractional portions that are in the lowest terms: see `void MixedNumber::reduce()`. Also, include an operation that converts a fraction to a mixed number: see constructor that takes *Fraction* `MixedNumber::MixedNumber(const Fraction& f)`.
- Q3 - [src/q3.cpp](#) - driver and implementation
 - Results: [q3.out](#)
 - Modified ArrayBag interface, header, cpp - [include/](#)
 - Modify the ArrayBag class: see [ArrayBag Header](#) [ArrayBag Implementation](#), given in Chapter 3 to meet the following specs:
 - Write a member function `replace` that replaces a given item in a given bag with another given item. The function should return a boolean value to indicate whether the replacement was successful: `bool ArrayBag::replace(const ItemType& oldEntry, const ItemType& newEntry)` . See trace: [q3.out#L6](#) of driver code starting here: [src/q3.cpp](#).
 - Write a recursive array-based implementation of the method `toVector` for the class `ArrayBag`. See helper function: [toVectorR](#)
 - Write a client function that [merges](#) two bags into a new third bag. Do not destroy the original two bags. See trace here: [q3.out#L15](#)
- Q4 - [src/q4.cpp](#) - driver
 - [include/](#) Classes declaration and implementation
 - Results: [q4.out](#)

- Write two *reverse* functions for the linkedBag ADT: [LinkedBag Header](#) [LinkedBag Implementation](#), using both [iterative](#) and [recursive](#) approaches. The function is supposed to reverse the linked list data structure. See trace here: [reverse loop](#), [reverse recursive](#)
- Also, write another function *removeDuplicates* that will locate and remove any duplicates in the linked list (only one occurrence of each linked list entry is allowed): see here [iterative](#). Test your functions using a driver program: [see trace here](#)