NAME: AHMED IRTAZA

SAP: 53759

Report algorithm analysis project

Analysis & pseudocode

A. Requirement summary

- Add a list of stocks (symbol, name, price)
- Display all stocks
- Sort stocks by price, symbol, or name using merge sort
- Exit

B. Module analysis

- **Stock**: simple strict holding symbol, name, price. Has print method.
- **Input**: reads n stocks from user input.
- **Display**: prints stock info for all stocks.
- **Sorting**: merge sort by selected field.
- **Main loop**: menu, delegates to above modules.

C. Pseudocode

Plaintext

```
Stock market system
--------------------
Main menu loop:
   1. Add stocks          --> call input stocks (stocks)
   2. Display stocks       --> call display stocks (stocks)
   3. Sort stocks         --> get sort criterion (price, symbol, and name)
                     --> call merge sort (stocks, 0, stocks. Size () - 1, sort by)
   4. Exit

Input stocks (stocks)
   Prompt user for 'n' (how many stocks)
   Repeat n times:
      Prompt for symbol, name, and price
      Push stock to stocks list

Display stocks (stocks)
   For each stock in stocks:
```

```
    print stock (symbol, name, and price)

Merge (stocks, left, mid, right, sort by)
  copy halves into temp arrays l, r
  while l and r not empty:
    compare on field selected by sort by
    take lesser (left) value into stocks
  copy any leftovers from l or r

Merge sort (stocks, left, right, sort by)
  if left < right:
    mid = (left + right) / 2
    recursively sort left half
    recursively sort right half
    merge the halves
```

D. Data flow

- User input goes into stocks vector
- Menu operates on stocks vector
- Sorting operates in-place on stocks vector
- Output always from stocks vector

Code test cases

Manual (interactive) test cases:

1. **Add and display**

   Code

```
 input:
- number of stocks: 2
- symbol: msft, name: microsoft, price: 384.65
- symbol: aapl, name: apple, price: 171.89
Display:
- symbol: msft, name: microsoft, price: $384.65
- symbol: aapl, name: apple, price: $171.89
```

sort by price

Code

```
 add as above.
Sort by: price
Output (display should show stocks ascending by price.)
- symbol: aapl, name: apple, price: $171.89
```

- symbol: msft, name: microsoft, price: $384.65

Sort by symbol

Code
(add several stocks. Sort by: symbol.)
Output should show stocks sorted alphabetically by symbol.

Sort by name

Code
Add: amzn, amazon, 146.97
Output sorted by name:
- amzn, amazon, 146.97
- aapl, apple, 171.89
- msft, microsoft, 384.65

Invalid input handling

Code
Input: -1 for number of stocks, or invalid choice on menu
Output: program should prompt for valid input; should not crash.

Sorting with ties

Code

6. Add multiple stocks with same price, symbol, or name.
7. Check order is stable (merge sort is stable).
8.

3. Benchmarking and complexity

- **Time complexity of merge sort:** o(n log n)
- **Space complexity:** o(n) due to temp arrays
- **Benchmarks:**
    - With 1, 10, 100, 1000 stocks, sort time should scale logarithmically
    - Could use a loop to auto-add random stocks for stress testing

Handling conflicting or ill-defined requirements

Potential conflicts:

- If user tries to sort with no stocks, code correctly handles (no stocks to sort!)
- If symbol/name has edge cases (contains spaces, special chars) – reading uses getline, so ok.

Recommendations:

- Add validation for price (non-negative)
- Clearer error messages
- Possible extension: allow deleting or updating stocks

Report drafting & presentation readiness

- Structure report as:
    - Introduction: system purpose
    - Architecture: modules (see pseudocode)
    - Analysis: complexity, correctness
    - Test cases: manual scenarios
    - Improvements: error handling, input validation, performance

Summary

Operation	Scenario	Expected result
Add stocks	Valid input	Stocks retained in list
Display stocks	Any list size	All stocks printed in order
Sort by price/symbol/name	Any list	List sorted as requested
Sort empty list	No data	"no stocks to sort!" message
Invalid menu input	Choice out of [1-4]	"invalid choice. Please try again."
Add duplicate symbol	Same symbol twice	Both entries appear; no de-duplication
Tied values (price etc.)	Stocks with identical fields	Sort is stable (checks with existing order)