**CPSC 319, Winter 2025**  
**PA-2: Searching & Sorting - Efficient Anagram Extraction & Grouping (12% OF Final Grade)**  
**Programming Marking Rubric**

**Student Information:**

* **Full Name:** Thomas Boyko
* **UCID:** 30191728
* **Tutorial Section:** T06

**Programming Evaluation Rubric**

**1. Main Program (PA2.java) - 10 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Program executes correctly, processing input and output as required. | 5 |  |
| Follows the problem constraints; implements logic correctly. | 3 |  |
| Code is well-organized, readable, and properly structured. | 2 |  |
| Total | **10** |  |

**2. Merge Sort Implementation - 10 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Correct implementation of Merge Sort for word sorting. | 5 |  |
| Maintains O(n log n) efficiency without unnecessary overhead. | 3 |  |
| Code is clear, well-documented, and modular. | 2 |  |
| Total | **10** |  |

**3. Anagram Grouping Implementation - 10 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Correctly groups words into anagrams using an appropriate data structure. | 5 |  |
| Optimized approach ensuring minimal redundant computation. | 3 |  |
| Code is modular, readable, and logically structured. | 2 |  |
| Total | **10** |  |

**4. File Handling and Input Processing - 5 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Handles file I/O correctly, including error handling. | 3 |  |
| Manages invalid inputs and edge cases properly. | 2 |  |
| Total | **5** |  |

**5. Linked List Data Structure - 5 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Implements a functional singly linked list with correct insert and traversal operations. | 3 |  |
| Code is structured, clear, and uses encapsulation properly. | 2 |  |
| Total | **5** |  |

**6. Code Documentation and Style - 5 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Functions, variables, and logic are clearly documented. | 3 |  |
| Code follows Java best practices and is consistently formatted. | 2 |  |
| Total | **5** |  |

**7. AI/External Tool Usage Disclosure - 2 Points**

|  |  |  |
| --- | --- | --- |
| Criteria | Max Points | Score |
| Clearly states whether AI/tools were used and to what extent. | 2 |  |
| Total | **2** |  |

**Final Score Summary**

|  |  |  |
| --- | --- | --- |
| Section | Max Points | Score |
| Main Program | 10 |  |
| Merge Sort Implementation | 10 |  |
| Anagram Grouping Implementation | 10 |  |
| File Handling and Input Processing | 5 |  |
| Linked List Data Structure | 5 |  |
| Code Documentation and Style | 5 |  |
| AI Usage Disclosure | 2 |  |
| Total Score | **47** |  |

**Rubric Details and Grading Scale**

**1. Correctness (5 Points Scale)**

* **5 Points:** Fully correct; all test cases pass.
* **4 Points:** Minor errors or edge cases fail.
* **3 Points:** Some test cases fail, but core logic is mostly correct.
* **2 Points:** Compiles but fails most test cases.
* **1 Point:** Attempt made but incorrect logic.
* **0 Points:** No effort or irrelevant submission.

**2. Efficiency (5 Points Scale)**

* **5 Points:** Algorithm adheres to expected complexity with no redundant operations.
* **4 Points:** Minor inefficiencies but still acceptable.
* **3 Points:** Noticeable inefficiencies or unnecessary computations.
* **2 Points:** Highly inefficient or deviates significantly from expected complexity.
* **1 Point:** Attempt made but poor implementation.
* **0 Points:** No effort or irrelevant submission.

**3. Readability & Documentation (5 Points Scale)**

* **5 Points:** Code is structured, uses clear naming, and has thorough comments.
* **4 Points:** Mostly readable but missing some comments or structure.
* **3 Points:** Readable but lacks organization or proper commenting.
* **2 Points:** Poorly written, hard to follow.
* **1 Point:** Minimal effort; disorganized and unclear.
* **0 Points:** No effort or irrelevant submission.

**4. AI/External Tool Usage (2 Points Scale)**

* **2 Points:** Clearly states AI/tool usage (or not) in code comments and report.
* **1 Point:** Incomplete acknowledgment or vague statement.
* **0 Points:** No disclosure of AI/tool use or violation of guidelines.

**Final Notes:**

* Code must be efficient, well-documented, and follow the project constraints.
* All submissions must adhere to academic integrity guidelines.
* Partial marks will be awarded for reasonable attempts that show effort and understanding.