BSSE23058

Approach Overview

Since I couldn't read student data from a file, I stored all records directly in the program's data section. The system:

1. **Storing and accessing student data**
2. **Parsing marks from strings**
3. **Calculating weighted grades**
4. **Printing results and failed students**

1. Data Storage & Access

- **Student Records:** Each student's data is stored as a **null-terminated string** in .data section since I couldn't get it to read from file so I did this.
- Contains: Name, Roll Number, followed by 6 marks (Quiz 1, Quiz 2, Assignments 1 & 2, Midterm, Final).
- **Array of Pointers:** Allows looping through students using pointer arithmetic.
- **Failed Students Storage:** Made an array of pointer to store the failed students.

2. Parsing Marks from Strings

- **Step-by-Step Extraction:**
    - **Loop through each student string** (s5 = current character pointer).
    - **Count spaces** to locate marks (marks start after the 3rd space).
    - **Convert ASCII digits to integers** (e.g., '1' '0' → 10).

3. Weighted Grade Calculation

- **Marks & Weights:**

Mark	Weight	Max Score
Quiz 1	5%	10
Quiz 2	5%	10
Assignment 1	10%	100
Assignment 2	10%	100
Midterm	30%	50
Final	40%	100

- **Calculation Logic:**
    - **Processes each mark sequentially** (Quiz 1 → Final).

- **Multiplies each mark by its weight percentage** (e.g., Quiz 1 is worth 5% of the total grade).
- **Adjusts for maximum possible marks** (e.g., Midterm is out of 50, but weighted as 30%).
- **Sums all weighted contributions** into a **total score** (out of 100).
- **3. Key Steps in Assembly**
- **Extract Marks from String**
- The program **scans the student's record string**, counting spaces to locate where marks begin.
- Converts **ASCII digits** (e.g., '7' '8') into **numeric values** (e.g., 78).

4. Printing Results

- **Student Info (Name + Roll#):**
  - Print characters until the **3rd space** (marks start after).
- **Total & Grade:**
  - Print computed total and letter grade (A, B, C, D, F).
- **Failed Students List:**
  - If grade is F, store pointer in failed_students.
  - After processing all students, print the list.
- **String Parsing:**
  - Used **space counting** to separate names from marks.
- **Integer Conversion:**
  - Multiplied by 10 for each new digit (e.g., '1' '0' → 1*10 + 0 = 10).
- **Weighted Averages:**
  - Applied different formulas for each mark type (Quizzes, Assignments, Exams).