```
# gradebook.py
# Author: [Anshul tagore]
# Title: GradeBook Analyzer - Student Marks Analysis CLI
import csv
import statistics

def welcome():
    print("\nWelcome to GradeBook Analyzer!")
    print("Choose input method:")
    print("1. Manual Entry")
    print("2. Load from CSV")

def manual_entry():
    marks = {}
    n = int(input("Enter number of students: "))
    for _ in range(n):
        name = input("Student Name: ")
        score = float(input("Marks: "))
        marks[name] = score
    return marks

def load_csv(filename):
    marks = {}
    try:
        with open(filename, newline='') as file:
            reader = csv.reader(file)
```

```
            next(reader)  # Skip header
            for row in reader:
                name, score = row
                marks[name] = float(score)
    except Exception as e:
        print("Error loading file:", e)
    return marks

def calculate_average(marks_dict):
    return round(statistics.mean(marks_dict.values()), 2)

def calculate_median(marks_dict):
    return round(statistics.median(marks_dict.values()), 2)

def find_max_score(marks_dict):
    return max(marks_dict.items(), key=lambda x: x[1])

def find_min_score(marks_dict):
    return min(marks_dict.items(), key=lambda x: x[1])

def assign_grades(marks_dict):
    grades = {}
    for name, score in marks_dict.items():
        if score >= 90:
            grades[name] = 'A'
        elif score >= 80:
```

```
            grades[name] = 'B'
        elif score >= 70:
            grades[name] = 'C'
        elif score >= 60:
            grades[name] = 'D'
        else:
            grades[name] = 'F'
    return grades

def grade_distribution(grades):
    dist = {'A': 0, 'B': 0, 'C': 0, 'D': 0, 'F': 0}
    for grade in grades.values():
        dist[grade] += 1
    return dist

def pass_fail(marks_dict):
    passed = [name for name, score in marks_dict.items() if score >= 40]
    failed = [name for name, score in marks_dict.items() if score < 40]
    return passed, failed

def print_table(marks_dict, grades):
    print("\nResults Table:")
    print(f"{'Name':<10}{'Marks':<10}{'Grade':<5}")
    print("-" * 25)
    for name in marks_dict:
        print(f"{name:<10}{marks_dict[name]:<10}{grades[name]:<5}")
```

```
def main():
    while True:
        welcome()
        choice = input("Enter choice (1 or 2): ")
        if choice == '1':
            marks = manual_entry()
        elif choice == '2':
            filename = input("Enter CSV filename: ")
            marks = load_csv(filename)
        else:
            print("Invalid choice. Try again.")
            continue

        print("\n--- Statistical Summary ---")
        print("Average:", calculate_average(marks))
        print("Median:", calculate_median(marks))
        print("Highest:", find_max_score(marks))
        print("Lowest:", find_min_score(marks))

        grades = assign_grades(marks)
        dist = grade_distribution(grades)
        print("\n--- Grade Distribution ---")
        for grade, count in dist.items():
            print(f"{grade}: {count} students")
```

```
        passed, failed = pass_fail(marks)
        print(f"\nPassed ({len(passed)}): {passed}")
        print(f"Failed ({len(failed)}): {failed}")

        print_table(marks, grades)

        again = input("\nRun again? (y/n): ")
        if again.lower() != 'y':
            break

if __name__ == "__main__":
    main()
```