Student management System

1. **Requirement**

- Ranking Student based on marks
- 2 sorting methods
- Edit, add, search, delete function
- Stack build from scratch

2. Design the ADT

```java
public interface Student {
    int getId();      no usages
    String getName();   no usages
    int getMarks();    no usages
    String getRank();   no usages
    String toString();
}
```

Student ADT

```java
public interface StudentStack {
    void push(StudentADT student);   no u
    StudentADT pop();   no usages
    StudentADT peek();   no usages
    boolean isEmpty();   no usages
    int size();   no usages
    void displayStudents();   no usages
}
```

Studentstack ADT

3. Design the Algorithm

```java
public interface StudentManagement {  no usages
    void addStudent(Student student);  no usages
    void editStudent(int id, String name, int marks);  no usages
    void deleteStudent(int id);  no usages
    void sortStudents();  no usages
    Student searchStudent(int id);  no usages
    void displayStudentsWithRanks();  no usages
}
```

Student management Algorithm

4. Result

```
Inserted: Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Added student: Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Inserted: Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Added student: Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Inserted: Student{id=3, name='Charlie', marks=67.0, status='Excellent'}
Added student: Student{id=3, name='Charlie', marks=67.0, status='Excellent'}
Inserted: Student{id=4, name='Diana', marks=54.0, status='Excellent'}
Added student: Student{id=4, name='Diana', marks=54.0, status='Excellent'}
Inserted: Student{id=5, name='Eve', marks=45.0, status='Excellent'}
Added student: Student{id=5, name='Eve', marks=45.0, status='Excellent'}
```

Add and display student

```
Students with Ranks:
Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Student{id=3, name='Charlie', marks=67.0, status='Excellent'}
Student{id=4, name='Diana', marks=54.0, status='Excellent'}
Student{id=5, name='Eve', marks=45.0, status='Excellent'}
```

```
Student edited: 3
Students with Ranks:
Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Student{id=3, name='Charlie Updated', marks=75.0, status='Excellent'}
Student{id=4, name='Diana', marks=54.0, status='Excellent'}
Student{id=5, name='Eve', marks=45.0, status='Excellent'}
```

Edit Student infomation

Delete student from the list using ID

```
Student deleted with ID: 5
Students with Ranks:
Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Student{id=3, name='Charlie Updated', marks=75.0, status='Excellent'}
Student{id=4, name='Diana', marks=54.0, status='Excellent'}
```

```
Sorted Students by name:
Students with Ranks:
Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Student{id=3, name='Charlie Updated', marks=75.0, status='Excellent'}
Student{id=4, name='Diana', marks=54.0, status='Excellent'}
```

Sort Student by name using mergesort

```
Sorted Students by marks:
Students with Ranks:
Student{id=1, name='Alice', marks=95.0, status='Excellent'}
Student{id=2, name='Bob', marks=82.0, status='Excellent'}
Student{id=3, name='Charlie Updated', marks=75.0, status='Excellent'}
Student{id=4, name='Diana', marks=54.0, status='Excellent'}
```

Sort Student by mark using Bubblesort

```
Found Student: Student{id=2, name='Bob', marks=82.0, status='Excellent'}
```

Search Student using ID