**1. Sorting Functions**

1.1 **Selection\_Sort(marks)**

* Implements the selection sort algorithm to sort the **marks** list in ascending order.
* Prints the sorted list after the sorting process.

1.2 **Bubble\_Sort(marks)**

* Implements the bubble sort algorithm to sort the **marks** list in ascending order.
* Prints the sorted list after the sorting process.

1.3 **top\_five\_marks(marks)**

* Prints the top five marks from the sorted **marks** list in descending order.

**2. Input of Marks**

* Takes the number of students (**n**) as input.
* Prompts the user to input the marks for each student.
* Populates the **marks** list with the entered marks.

**3. Menu and Main Program**

3.1 Menu Display

* Prints a menu with options for performing selection sort, bubble sort, or exiting the program.

3.2 Main Loop (**while flag == 1**)

* Repeats until the user chooses to exit (**flag = 0**).

3.3 Menu Options

**3.3.1 Option 1 - Selection Sort**

* Calls **Selection\_Sort** to sort the **marks** list using selection sort.
* Asks the user if they want to display the top marks, then calls **top\_five\_marks** if the answer is 'yes'.

**3.3.2 Option 2 - Bubble Sort**

* Calls **Bubble\_Sort** to sort the **marks** list using bubble sort.
* Asks the user if they want to display the top marks, then calls **top\_five\_marks** if the answer is 'yes'.

**3.3.3 Option 3 - Exit**

* Exits the program.

**3.3.4 Invalid Choice**

* Prints a message for an invalid choice and exits the program.

**Note:**

* The code assumes that the user always inputs valid integers for the number of students and their marks.
* The sorting algorithms are applied directly to the original **marks** list.
* The program offers the option to display the top five marks after sorting.
* The use of the **flag** variable controls the loop, and setting it to 0 exits the program.
* The user is prompted for displaying top marks even after choosing to exit, which might be improved for clarity.

Algorithm:Top of Form

1. Initialize an empty list marks to store the marks of students.

2. Take user input for the number of students (n) and their marks.

3. Define two sorting functions: Selection\_Sort and Bubble\_Sort. Each function takes the list of marks as input and sorts it in ascending order using the respective sorting algorithm.

4. Define a function top\_five\_marks that prints the top marks in descending order.

5. Use a while loop with a menu-driven interface to allow the user to choose between sorting algorithms or exit the program.