**1. Attendee Functions**

1.1 **addAttendee(sorted\_list)**

* Takes a sorted list (**sorted\_list**) as input.
* Prompts the user to enter the roll number of the attendee.
* Appends the roll number to the list and sorts the list.

1.2 **display(sorted\_list)**

* Takes a sorted list (**sorted\_list**) as input.
* Prints the sorted list of attendees.

1.3 **binarySearch(sorted\_list, roll)**

* Takes a sorted list (**sorted\_list**) and a roll number (**roll**) to search for.
* Performs binary search on the sorted list to find the index of the given roll number.
* Prints whether the student attended the program or not.

1.4 **fibonacciSearch(sorted\_list, roll)**

* Takes a sorted list (**sorted\_list**) and a roll number (**roll**) to search for.
* Implements Fibonacci search algorithm to find the index of the given roll number.
* Prints whether the student attended the program or not.

**2. Helper Function min\_fibonacci(m)**

* Finds the largest Fibonacci number less than or equal to the given number **m**.

**3. Menu Function menu()**

* Prints a simple menu with options for adding an attendee, displaying the list, performing binary search, performing Fibonacci search, and exiting.

**4. Main Program**

4.1 **sorted\_attendees**

* Initializes an empty list to store the sorted list of attendees.

4.2 Main Loop

* Continuously displays the menu and waits for user input until the user chooses to exit.

4.3 Menu Options

**4.3.1 Option 1 - Add Attendee**

* Calls **addAttendee** to add an attendee to the sorted list.

**4.3.2 Option 2 - Display**

* Calls **display** to print the sorted list of attendees.

**4.3.3 Option 3 - Binary Search**

* Takes a roll number as input and calls **binarySearch** to search for the student in the sorted list using binary search.

**4.3.4 Option 4 - Fibonacci Search**

* Takes a roll number as input and calls **fibonacciSearch** to search for the student in the sorted list using Fibonacci search.

**4.3.5 Option 5 - Exit**

* Exits the program if the user chooses this option.

**4.3.6 Invalid Choice**

* Prints a message for an invalid choice.

**Note:**

* The code assumes that the user always inputs valid integers for choices and roll numbers, which might not always be the case in a real-world scenario.
* The binary search and Fibonacci search algorithms are implemented within their respective functions.
* The code could benefit from additional input validation and error handling.

Top of Form