Class assignment (Searching)

Description:

In this programming assignment, you will develop a C++ program to implement the Enhanced Stock Price Analysis Tool. The program will allow users to input historical stock prices, perform sorting, and provide various data analysis features. You are required to use separate functions to enhance code modularity and organization.

Requirements:

1. Create a C++ program that utilizes separate functions for different tasks, enhancing code organization and readability.
2. Implement the Enhanced Stock Price Analysis Tool with the following functionalities:

   a. **inputStockPrices**: A function that allows users to input historical stock prices for a specified number of days. The function should take an array of StockRecord structures and the number of days as parameters.

   b. **bubbleSortByDate**: A function that sorts the stock prices by date in chronological order using the bubble sort algorithm. The function should take an array of StockRecord structures and the number of days as parameters.

   c. **calculateAveragePrice**: A function that calculates and returns the average stock price. The function should take an array of StockRecord structures and the number of days as parameters.

   d. **findHighestAndLowestPrices**: A function that finds and displays the highest and lowest stock prices along with their respective dates. The function should take an array of StockRecord structures and the number of days as parameters.

   e. **searchStockPricesInRange**: A function that allows users to search for stock prices within a specified date range (start date and end date) and display all prices within that range. The function should take an array of StockRecord structures and the number of days as parameters.

   f. **countPricesAboveThreshold**: A function that counts the number of days when the stock price was higher than a user-specified threshold and returns the count. The function should take an array of StockRecord structures, the number of days, and the threshold as parameters.

   g. Implement a menu-driven main program that guides users through the above functionalities and allows them to exit the program.

Instructions:

1. Begin the program by displaying a welcome message.
2. Ask the user to enter the number of days for which they have historical stock prices.
3. Utilize the inputStockPrices function to prompt the user to input the date (in the format "YYYY-MM-DD") and the corresponding stock price (as a floating-point number) for each day.
4. Use the bubbleSortByDate function to sort the stock prices by date in chronological order.
5. Display a menu of additional features within a loop until the user chooses to exit. Each menu option should correspond to one of the functions described above.
6. Ensure that the program handles user inputs and errors gracefully, especially when searching for stock prices on dates that are not in the list and when calculating averages, highest, lowest, and the count of days above a threshold.
7. Allow the user to exit the program when they are finished.

Bonus: Solve this problem and implement all the tasks using pointers and dynamic arrays. Please do not submit the bonus portion on the canvas. You should check the bonus portion with the instructor. Bonus points is only valid during the class hours.