Homework 6 Report
Author: Ahmet Halil Yamlı (2.04.2024)

Summary

This C program appears to be designed for numerical data analysis. It offers various functionalities, including:

- Finding the size of an array (until a sentinel value of -1 is encountered)
- Duplicating an array into another array
- Counting the number of times a specific number appears in an array
- Determining the maximum value in an array
- Sorting an array in ascending order (using a bubble sort implementation)
- Finding the number of repetitions of numbers within a specified range (inclusive) in an array
- Printing a histogram to visually represent the number of repetitions
- Filling an array with numbers from a specified range (inclusive) that exist in the original array
- Calculating the average (mean) of an array
- Calculating the median of an array (handling both even and odd-sized arrays)
- Identifying the mode(s) of an array within a specified range (the number(s) that appear most frequently)

Detailed Function Descriptions

1. find_size(int arr[]):
    - Iterates through the array until it encounters the sentinel value (-1).
    - Returns the index of the element before -1, representing the array's size (excluding the sentinel).
2. duplicate_array(int main_arr[], int destination_arr[], int size):
    - Copies elements from main_arr to destination_arr up to size elements, excluding any elements with a value of -1.
3. find_repeat_times(int arr[], int number, int size):
    - Counts the number of times the specified number appears in the arr array within the given size.
4. find_max_number(int arr[], int size):
    - Finds the largest element in the arr array within the provided size.
5. sortArray(int arr[], int size):
    - Implements a bubble sort algorithm to arrange the elements of arr in ascending order.
6. find_repeated_numbers(int mainArr[], int arr[], int a, int b, int size):
    - Counts the repetitions of numbers within the range [a, b] (inclusive) in the mainArr array.
    - Stores the counts in the arr array.
    - Fills the remaining elements of arr with -1.
7. print_histogram(int arr[], int size):
    - Finds the maximum repetition count in the arr array.
    - Iterates through arr from the maximum count down to 0.
    - For each count, prints asterisks (*) corresponding to the number of repetitions, followed by spaces for empty bars.
8. fill_interval_numbers(int mainArr[], int destinationArr[], int a, int b, int size):

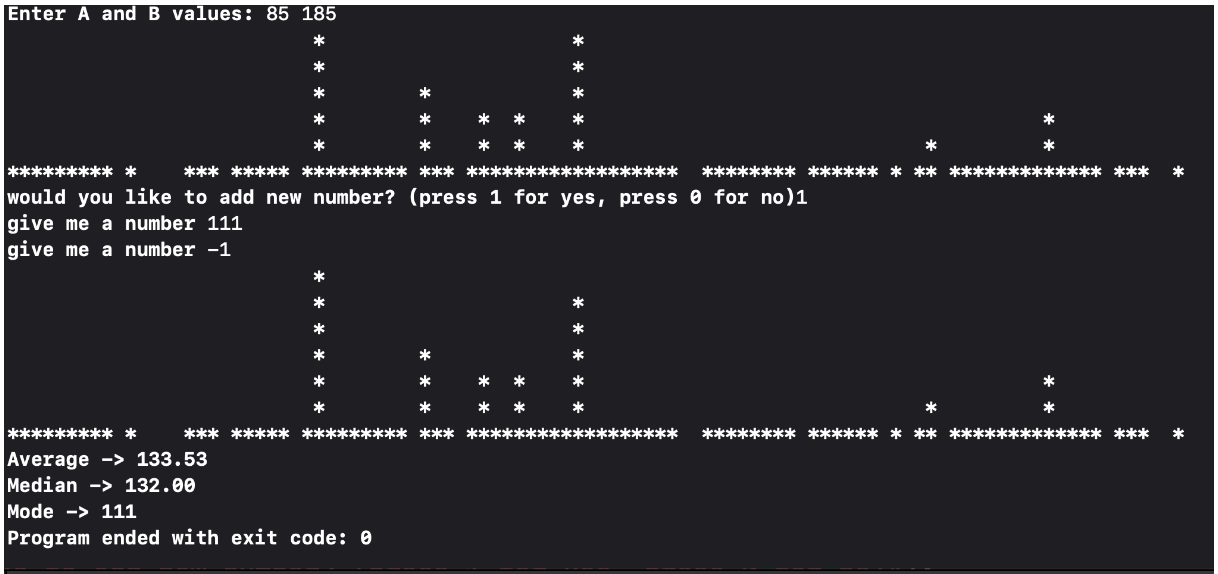
    - Creates an array destinationArr to hold numbers from mainArr that fall within the inclusive range [a, b].
    - Copies elements from mainArr to destinationArr if they are within the range, excluding -1.
    - Fills the remaining elements of destinationArr with -1.
9. calc_average(int arr[], int size):
    - Calculates the average (mean) of the elements in the arr array by summing all elements and dividing by the size.
10. calc_median(int arr[], int size):

- Determines the median of the arr array based on its size:
    - For even-sized arrays, the median is the average of the middle two elements.
    - For odd-sized arrays, the median is the middle element.

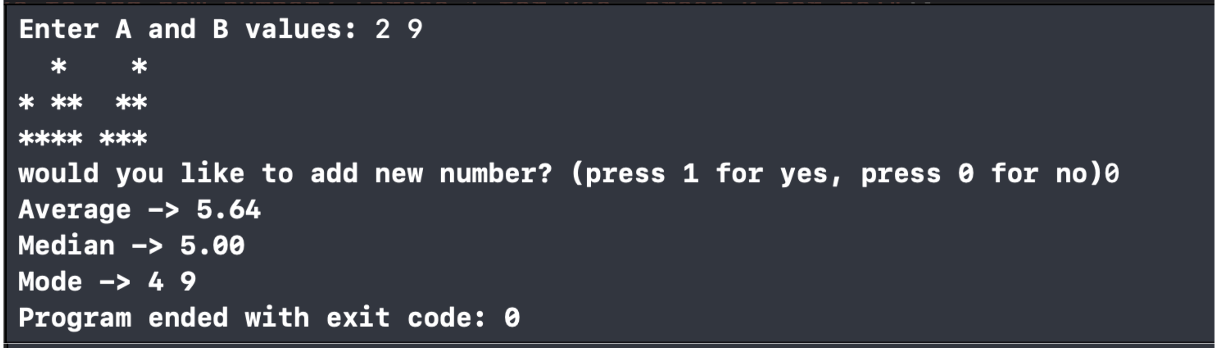
Main Function (main)

1. Declares an array number_array containing integer data.
2. Prompts the user to enter two integer values A and B to define a range.
3. Calculates the size of the array to be used for storing repetitions (valueB - valueA + 2).
4. Declares arrays for storing repetitions (repeated_number) and filtered numbers within the range (interval_numbers).
5. Calls functions to:
    - Count repetitions of numbers in the range (find_repeated_numbers).
    - Fill the interval_numbers array with numbers from the range (fill_interval_numbers).
    - Print the histogram (print_histogram).
    - Sort the interval_numbers array (sortArray).
6. Prompts the user to enter 1 if they want to add new numbers, or 0 if they do not.
7. **If the user wants to add new numbers:**
    - Asks the user to enter the new numbers.
    - Creates a new array to contain the new numbers.
    - Calls the functions again starting from step 5 to process the new array.
    - Calculates the mean (calc_average) and median (calc_median) of the numbers in the range.
    - Finds the mode(s) of the numbers in the range (can be calculated using find_repeated_numbers, but the code currently does not have a section to calculate the mode).
    - Prints the mean, median, and mode to the user.
8. **If the user does not want to add new numbers:**
    - Calculates the mean (calc_average) and median (calc_median) of the numbers in the range.
    - Finds the mode(s) of the numbers in the range (can be calculated using find_repeated_numbers, but the code currently does not have a section to calculate the mode).
    - Prints the mean, median, and mode to the user.
9. Terminates the program.

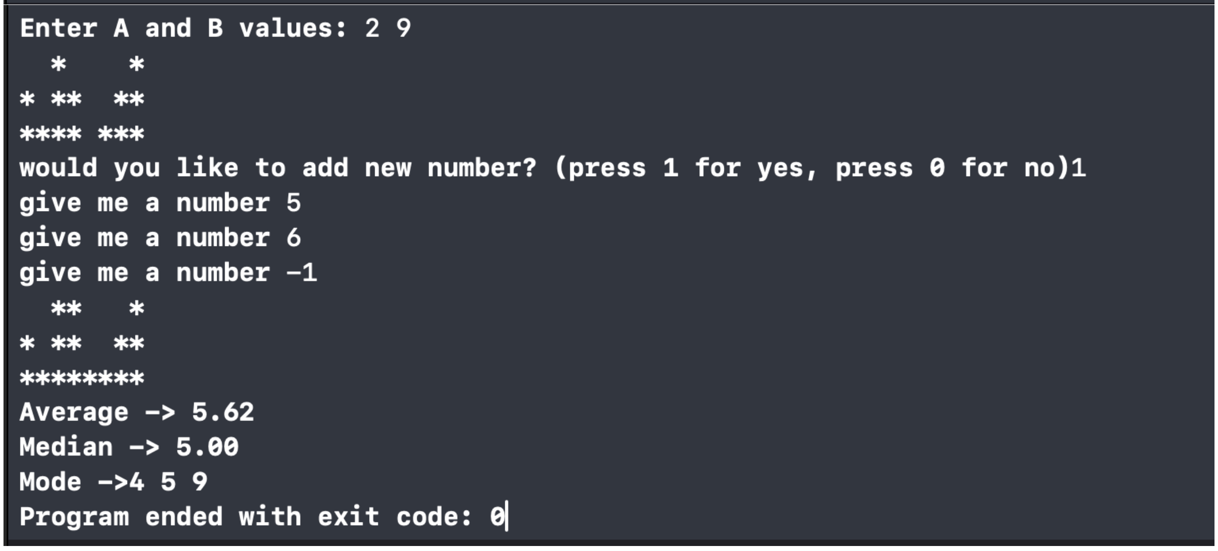
below are some sample code outputs:



```
Enter A and B values: 85 185
                          *                     *
                          *                     *
                          *        *            *
                          *        *    *  *    *                                       *
                          *        *    *  *    *                             *         *
********* *    *** ***** ********* *** ******************  ******** ****** * ** ************* ***  *
would you like to add new number? (press 1 for yes, press 0 for no)1
give me a number 111
give me a number -1
                          *
                          *                     *
                          *                     *
                          *        *            *
                          *        *    *  *    *                                       *
                          *        *    *  *    *                             *         *
********* *    *** ***** ********* *** ******************  ******** ****** * ** ************* ***  *
Average -> 133.53
Median -> 132.00
Mode -> 111
Program ended with exit code: 0
```



```
Enter A and B values: 2 9
  *    *
* **  **
**** ***
would you like to add new number? (press 1 for yes, press 0 for no)0
Average -> 5.64
Median -> 5.00
Mode -> 4 9
Program ended with exit code: 0
```



```
Enter A and B values: 2 9
  *    *
* **  **
**** ***
would you like to add new number? (press 1 for yes, press 0 for no)1
give me a number 5
give me a number 6
give me a number -1
  **   *
* **  **
********
Average -> 5.62
Median -> 5.00
Mode ->4 5 9
Program ended with exit code: 0
```