Omer Ahmer

Assignment 4

/\*

Description: Takes an array of inputs, sorts the array, searches for a number that the user chooses, and takes the mean of all of the numbers in the array

Author: Omer Ahmer

COMSC 165-5065

Date: 9/17/22

Status: Complete

\*/

#include <iostream>

**using** **namespace** std;

**void** selectionSort(**int** y[], **int** number);

**int** binarySearch(**int** y[], **int** number, **int** count);

**double** computeMean(**int** y[], **int** count);

**int** main() {

**int** inputs;

**const** **int** MAX\_SIZE = 50;

cout << "Enter number of inputs:" << endl;

cin >> inputs;

**int** arr[MAX\_SIZE];

**for** (**int** i = 0; i < inputs; i++) {

cout << "Enter numbers:" << endl;

cin >> arr[i];

}

**for** (**int** j = 0; j < inputs; j++) {

cout << arr[j] << " ";

}

cout << endl;

**int** search\_number;

cout << "Enter number to be searched:" << endl;

cin >> search\_number;

selectionSort(arr, search\_number);

**int** found = binarySearch(arr, search\_number, inputs);

**if** (found < 0) {

cout << "The number " << search\_number << " was found." << endl;

}

**else** {

cout << "The number " << search\_number << " was not found." << endl;

}

computeMean(arr, inputs);

cout << "After sorting, the array is:" << endl;

**for** (**int** k = 0; k < inputs; k++) {

cout << arr[k] << " ";

}

cout << endl;

}

**void** selectionSort(**int** y[], **int** number) {

**for** (**int** i = 0; i < number; i++) {

**for** (**int** j = 0; j < number-1; j++) {

**if** (y[i]>y[j]) {

**int** temp = y[i];

y[i] = y[j];

y[j] = temp;

}

}

}

}

**int** binarySearch(**int** y[], **int** number, **int** count) {

**int** min = 0, max = count, mid;

**while** (min <= max) {

mid = (min+max)/2;

**if** (y[mid] < number) {

min = mid + 1;

}

**else** {

**if** (y[mid] > number) {

max = mid - 1;

}

**else** {

**return** mid;

}

}

}

**return** -1;

}

**double** computeMean(**int** y[], **int** count) {

**int** sum = 0;

**for** (**int** i = 0; i < count; i++) {

sum += y[i];

}

**return** (**double**)sum/count;

}

**Enter number of inputs:**

6

**Enter numbers:**

38

**Enter numbers:**

13

**Enter numbers:**

45

**Enter numbers:**

87

**Enter numbers:**

11

**Enter numbers:**

-5

**38 13 45 87 11 -5**

**Enter number to be searched:**

25

**The number 25 was not found.**

**After sorting, the array is:**

**87 45 38 13 11 -5**

**Program ended with exit code: 0**