CSCI 121 Project 07

Introduction

Purpose

The purpose of this project is to help the students to reinforce the knowledge from Chapter 7 of the textbook.

Objectives

1. Review top-down design
2. Understand and apply the syntax of array
3. Understand and apply the linear search, select-sort, insert-sort, and bubble-sort algorithms

Problem Description

Finish the implementation of the following program. Keep two things in mind:

1. Study the code carefully.
2. Add menu function accordingly

Students should implement and test the following functions:

```
void fill_array(int arr[], int size);
// pre-condition: The arr has actual size that is greater than or equal to
size
// post-condition: arr[0], ..., arr[size-1] is filled from keyboard

void print_array(int arr[], int size);
// pre-condition: The arr has actual size that is greater than or equal to
size
// post-condition: arr[0], ..., arr[size-1] is printed to screen with 5
elements per line

int linear_search(int arr[], int size, int key);
// pre-condition: arr has given size
// post-condition: The index of first occurrence of key in arr is returned. If
the key cannot be found in arr, -1 is returned

void select_sort(int arr[], int size);
// pre-condition: arr has given size
```

```
// post-condition: the elements in arr are rearranged from least to largest

void insert_sort(int arr[], int size);
// pre-condition: arr has given size
// post-condition: the elements in arr are rearranged from least to largest

void bubble_sort(int arr[], int size);
// pre-condition: arr has given size
// post-condition: the elements in arr are rearranged from least to largest
```

Of course, a menu function is needed. The `main` function will look like following:

```
int main() {
    int choice;
    int a[9];
    do{
        menu();
        cout << "Enter your choice: ";
        cin >> choice;
        switch(choice)
        {
            case 1:
            {
                fill_array(a, 9);
                cout << "Enter the key you want to search: ";
                int key;
                cin >> key;
                int index = linear_search(a, 9, key);
                if(index == -1)
                    cout << "The key " << key << " is not in array\n";
                else
                    cout << "The key " << key << " is #" << (index + 1) << "
element in array\n";
                break;
            }
            case 2:
            {
                fill_array(a, 9);
                select_sort(a, 9);
                cout << "After sort, the array is:\n";
                print_array(a, 9);
                break;
            }
            case 3:
```

```
            {
                fill_array(a, 9);
                insert_sort(a, 9);
                cout << "After sort, the array is:\n";
                print_array(a, 9);
                break;
            }
            case 4:
            {
                fill_array(a, 9);
                bubble_sort(a, 9);
                cout << "After sort, the array is:\n";
                print_array(a, 9);
                break;
            }
            case 5:
            {
                cout << "Thank you for using the array functions\n";
                break;
            }
            default:
            {
                cout << "Wrong choice. Please choose from menu: ";
                break;
            }
        }
    }while(choice != 5);

    return 0;
}
```

This is a good example to show how to write the test program to test functions.