Lab 8

Thursday, November 9, 2023

You will get 1 point for attending the lab, making an effort to work on the problems, and **discussing your work with a lab instructor during the lab**. If you finish during the lab, demonstrate your working code to a lab instructor and they will give you the full 2 points. If you do not finish during the lab, discuss what you've done with a lab instructor and they will give you an opportunity to finish outside of the lab and attend office hours to demonstrate your solution.

Practice Problems. This lab uses the following student structure:

```
struct student {
  int id;           // student ID
  char name[20];    // first name of student
};
```

Complete the following function that changes the name of a given student.

```
// change_name(s, new_name) renames the student s to have the name given by
//   new_name
// requires: s points to a valid student that can be modified
//           new_name points to a valid string of length at most 19
void change_name(struct student * s, char * new_name);
```

In `main`, create a few student records (including your name and ID number) and use `strcmp` and `assert` to test `change_name`. What is the running time of `change_name`?

Second, complete the following function that searches for students with a given name in an array of students `arr`. The function must run in $O(n)$ time when the array has length n.

```
// find_name(name, arr, n, ids) searches for student(s) with given name in arr;
//   returns the number of students found, and the array 'ids' is updated to
//   contain the ID numbers of those students
// requires: arr has length n
//           students in arr have unique ids
//           ids points to enough memory to hold the found student ids
int find_name(char * name, struct student arr[], int n, int ids[]);
```

In `main`, use `assert` to test that `find_name` works correctly on an array of students where at least two students have the same name.

Marked Problem. Complete the following function that searches for a student with a given ID number in an array of students `arr`. The function must run in $O(n)$ time when the array has length n.

```
// find_id(id, arr, n, found_name) searches for a student with given id in arr;
//   returns true if such a student is found and found_name is updated to
//   hold the student's name; otherwise returns false
// requires: arr has length n
//           students in arr have unique ids
//           found_name points to enough memory to hold a name
bool find_id(int id, struct student arr[], int n, char * found_name);
```