
CS 2263 Assignment 2 Fall 2018

Due Date: Friday, Oct. 5, 5:00 pm

Purpose: Practice with pointers and arrays, and using Git for local version control

1. Debug, Modify, and Examine

- (a) The following function supposedly computes the sum and average of the numbers in the array `a`, which has length `n`. `avg` and `sum` point to variables that the function should modify. Unfortunately, the function contains several errors; find and correct them.

```
void avgSum(double a[], int n, double *avg, double *sum){
    sum = 0.0;
    for(int i=0; i<n; i++)
        sum += a[i];
    avg = sum/n;
}
```

- (b) Once you have fixed the function, rewrite it using pointer arithmetic instead of array indexing (i.e. eliminate `i` and all uses of `[]`).
- (c) Write a C program to test `avgSum` (either corrected from a, or from b). Examine it in the GNU debugger (either `DDD` or `gdb`) just after entering `avgSum`. Examine the values of the parameters `avg` and `sum`, then change stack frames and examine the values of the two variables where the sum and average will be stored, as well as their addresses. Show your terminal session using one or more screenshots.

Submit code listings for (a) and (b), and screenshots for (c).

2. Finding the Best Flight

The following table shows daily flights from one city to another:

Departure time	Arrival time
8:00	10:16
9:43	11:52
11:19	13:31
12:47	15:00
14:00	16:08
15:45	17:55
19:00	21:20
21:45	23:58

Write a C program, `flights.c`, that asks the user to enter a time (expressed in hours and minutes, using the 24-hour clock). The program then displays the departure and arrival times for the flight whose departure time is closest to that entered by the user:

Enter a 24-hour time: 13:15

Closest departure time is 12:47, arriving at 15:00

Your program must include implementation of the following function:

```
void findClosestFlight(int desiredTime, int *departureTime, int *arrivalTime);
```

Which reads a file `flights.dat` of flights, using:

```
FILE *f = fopen("flights.dat", "r");
```

and `fscanf` (which works just like `scanf`, but with an extra first parameter giving the file handle, which here is `f`). This also requires including the header file `stdlib.h`. `findClosestFlight` then finds the flight with the closest departure time, and returns its departure and arrival times via the `departureTime` and `arrivalTime` parameters. You may assume that all times are for the same day.

Test your program using the supplied `flights.dat` file, and with selected desired times. Record a terminal session showing your testing.

Commit at least two version of your program to your local Git repository.

3. 2D Array with Pointers

Modify my solution (attached) to the Bellman-Ford algorithm from lab 2 to use two arrays to represent the adjacency matrix: a 1D array (as before), plus an array of pointers to each row. Then modify each reference to the adjacency matrix to use 2D array indexing: `graph[i][j]`, where `graph` is the array of pointers.

Commit at least two version of your program to your local Git repository.

To pass in the assignment: Create a single pdf document with your code listings and terminal session for question 1 and with your code listing and terminal session for questions 2 and 3, as well as the source files for question 2 and 3, **and the log file for your Git repository** (using the `git log` command). Submit these files to the Desire2Learn dropbox. Name your documents `LastName_FirstName_As2.pdf` and `LastName_FirstName_As2.zip` (`LastName` and `FirstName` are of course substituted with your last and first name).