EECE.2160: ECE Application Programming

Summer 2018

Lecture 11: Key Questions
June 20, 2018

QUESTIONS:

1. Review the basics of using structures.
2. Explain how one structure can be nested inside another.
3. Explain the use of the fopen() function.
4. Explain the use of the fclose() function.
5. Explain how fscanf() and fprintf() are used for formatted file I/O.
6. Explain how `fread()` and `fwrite()` are used for unformatted I/O.
7. Describe the standard input and output streams.
8. Describe how to test that an operation has reached the end of a file or caused an error.
9. Describe the functions used for character I/O.
10. Describe the functions used for line I/O.

EXAMPLES:

For today's exercise, you will complete the following functions that work with the structures Name and SINew. The structure definitions are listed below:

```
typedef struct {
    char first[50];
    char middle;
    char last[50];
} Name;

typedef struct {
    Name sname;
    unsigned int ID;
    double GPA;
} SINew;
```

The function descriptions are as follows:

For the Name structure:

- **void printName(Name *n)**: Print the name pointed to by n, using format <first> <middle>. <last>
- **void readName(Name *n)**: Prompt for and read a first, middle, and last name, and store them in the structure pointed to by n

For the StudentInfo structure:

- **void printStudent(SINew *s)**: Print information about the student pointed to by s
- **void readStudent(SINew *s):** Prompt for and read information into the student pointed to by s
- **void printList(SINew list[], int n):** Print the contents of an array list that contains n SINew structures
- **int findByLName(SINew list[], int n, char lname[]):** Search for the student with last name lname in the array list. Return the index of the structure containing that last name, or -1 if not found
- **int findByID(SINew list[], int n, unsigned int sID)**: Search for the student with ID # sID in the array list. Return the index of the structure containing that last name, or -1 if not found

From Name.c:

```
// Print contents of Name struct
void printName(Name *n) {





}
```

```
// Read information into existing Name
void readName(Name *n) {






}
```

From SINew.c:

```
// Print information about student
void printStudent(SINew *s) {






}
```

```
// Reads student information into existing structure
void readStudent(SINew *s) {













}
```

From SINew.c (continued):

```
// Print list of students
void printList(SINew list[], int n) {

}

// Find student in list, based on last name
// Returns index if student found, -1 otherwise
int findByLName(SINew list[], int n, char lname[]) {

}
```

<u>From SINew.c (continued):</u>

```
// Find student in list, based on ID #
// Returns index if student found, -1 otherwise
int findByID(SINew list[], int n, unsigned int sID) {

}
```

1. Write a program to:
   - Read three integers from file `myinput.txt`
   - Determine the sum and average of those values
   - Write the original values, sum, and average to file `myoutput.txt`.

2. Show the output of each of the following short program.

a. Input: **`Test Input    1    23 4 5`**

```
void main() {
   char c;
   char buffer[50];
   int i, n;
   i = 0;
   while ((c = fgetc(stdin)) != '\n') {
      if (c != ' ') {
          buffer[i++] = c;
      }
   }
   buffer[i] = '\0';
   fputs(buffer, stdout);
}
```

b. Input:

```
Test1
Test 2
abcdefghijklmnopqrstuvwxyz
This is a test of the fgets() function
```

```
void main() {
   char str[25];
   int i;
   for (i = 0; i < 5; i++) {
      fgets(str, 24, stdin);
      strcat(str, "\n");
      fputs(str, stdout);
   }
}
```

c. Input:

1024Some other stuff

```
void main() {
   char c;
   char buffer[50];
   int n = 0;

   // isdigit in <ctype.h>
   while (isdigit(c = getchar())) {
      n = n * 10 + (c - 48);   // Hint: '0' = 48   }
      // (ASCII value)
   ungetc(c, stdin);
   fgets(buffer, 50, stdin);

   printf("n = %d, n * 2 = %d\n", n, n * 2);
   printf("buffer = %s\n", buffer);
}
```