

Assignment 2 — ECEN 449/749 (Fall 2017)

Due: Oct. 11, 2017, midnight

This assignment requires you to use the “qsort” function in C. You can easily find a lot of explanations of this function by googling “C qsort.”

For problems 1, 2, and 5 in this assignment, your C program needs to read the input from a text file “inputX.txt”, and write the output to a text file “outputX.txt,” where X is the problem number. For example, your program for the first problem reads input from “input1.txt” and writes output to “output1.txt.”

1.(25pt)

The input file contains 10 integer numbers, one number in each line. Write a program to sort these 10 numbers in descending order.

Sample input:

```
5
7
2
8
0
6
1
9
4
3
```

Sample output:

```
9
8
7
6
5
4
3
2
1
0
```

2. (25pt)

Design a program that reads a file of student records, which include the ID, number of earned credits, and GPA. Depending on the input, the program either sorts all students in descending order by their number of earned credits, or by their GPA. There are multiple ways to solve this problem. Please explain your solutions in the comments at the top of your code.

The specific file format of the input file is as follows:

1. The first line of the file contains a single integer N . N is the number of student records in the file.
2. Each of the next N lines contains three numbers, separated by space. They are, in this order, student ID (type int), number of earned credits (type int), and GPA (type float).
3. The last line contains a single character. If the character is ‘C,’ that means the program should sort students by their number of earned credits. Otherwise, the program should sort them by their GPA.

The program outputs the sorted student records.

Sample input 1:

```
5
1 60 3.2
2 77 2.8
3 24 3.8
4 51 3.6
5 40 3.3
C
```

Sample output 1:

```
2 77 2.8
1 60 3.2
4 51 3.6
5 40 3.3
3 24 3.8
```

Sample input 2:

```
5
1 60 3.2
2 77 2.8
3 24 3.8
4 51 3.6
5 40 3.3
G
```

Sample output 2:

```
3 24 3.8
4 51 3.6
5 40 3.3
1 60 3.2
2 77 2.8
```

3. (25pt)

Write a C language swap() function with the following function header:

```
void swap(void *p1, void *p2, unsigned int size);
```

The swap function swaps contents of p1 and p2. Pointers p1 and p2 are of the same type. The swap function works for ANY types of pointers, including pointers for user-defined structures. The user needs to specify the size of the type in the last parameter. Your program needs to include the swap() function as well as a main function that tests it.

Example:

```
int main(){
int x = 10; int y = 20;
swap(&x, &y, sizeof(int));
//x = 20, y = 10 after swap
char a = 'F'; char b = 'G';
swap(&a, &b, sizeof(char));
//a = G, b = F after swap
}
```

4.(25pt)

Write a Makefile for the C codes in 449_HW2.zip. The Makefile needs to compile as few files as possible every time a user types “make.” It also needs to support “make clean.”

5. (749 only, 20pt)

Repeat problem 2 with only two modification: First, the input file only contains N lines of student records, but you know nothing about N (For example, you cannot assume that N is no larger than 100000). Second, instead of sorting student records, the program simply outputs all student records in reverse order of the input. There are multiple ways to solve this problem. Please explain your solutions in the comments at the top of your code. (Hint: The easiest solution is to create a “linked list” of student records.)

Sample input:

```
1 60 3.2
2 77 2.8
3 24 3.8
4 51 3.6
5 40 3.3
```

Sample output:

```
5 40 3.3
4 51 3.6
3 24 3.8
2 77 2.8
1 60 3.2
```