Lab 2: Review on C Programming

In this class and assignment, you will practice a programming problem by using recursive function, pointer and structure.

1. **Recursive function**

As we discussed in the class, we will implement a program that prints all possible permutations when a set of elements is given by using a recursive function. For example, when a list “abc” is given as a standard input, your program should print all possible permutations in the standard output, which is abc, acb, bac, bca, cba, and cab.

Your input list has non-redundant characters. The number of elements ranges from 1 to 50. The elements in the list can be upper-case or lower-case alphabet, number, or special character.

1. An example of command line

>p2\_1 abc

1. The corresponding output

abc

acb

bac

bca

cba

cab

1. **Structure**

Use a command line argument with a designated input file name and your own output file name. Read the first integer from the input file to figure out the number of students. From the second line, the student name and three scores (for literature, math, and science) are provided in each line. Create an array of structures, and put the information in it. Then print the structures in the output file.

1) An example of command line

>p2\_2 input.txt output.txt

1. An example of Input (input.txt)



1. The corresponding output (output.txt)



* program name: p2\_2.c
* data structure : structure

typedef struct studentT{

char \*name;

int literature;

int math;

int science;

}studentT;

* input : the number of students and their information (see input.txt)
* output : all of the score information
* conditions :
* the length of the user name should be up to 30 characters
* no blank space is allowed in the name or major
* student name and scores are separated by space in the input file

1. **Structure that has array of structure**

* An example of command line

>p2\_3 input.txt output.txt

* An example of Input (input.txt)

Hanyang

2

Sarah 96 90 80

Minsu 86 60 90

--

Seoul

3

Minjee 30 50 60

Ye 20 90 60

* The corresponding output (output.txt)

Hanyang 91 75 85

Seoul 25 70 60

* program name: p2\_3.c
* data structure : structure

typedef struct courseT{

student students[MAX\_ENROLL]

int numEnrolled;

}courseT;

* input : the number of students for each class and their student information (see input.txt)
* output : all of the score information
* conditions :
* the length of the user name should be up to 30 characters
* no blank space is allowed in the name or major
* student name and scores are separated by space in the input file