Lab 1: Review on C Programming

During the data structure class, you will be given some practice problems (and homework) that need C programming. In this first class, you will practice 4 programming problems by using array and pointer in order to refresh your C programming skills. In addition, you need to know a fundamental file I/O, and how to use command line arguments.

You should finish the first problem (1-1) during the lab session and confirm it with the TA. For the other 3 problems (1-2, 1-3, 1-4), you can submit it to the course website (<https://portal.hanyang.ac.kr)> (Due on 3/6 11:59PM). Your report should include the description of your own implementation. No email submission.

**1-1. Array of characters**

Obtain a user name from the standard input, and put it on the standard output.



* program name: p1\_1.c
* data structure : array of characters
* input : a user name (string)
* output : a user name (string)
* conditions :
  + the length of the user name should be up to 30 characters
  + blank spaces should also be part of the name (e.g. Abraham Lincoln)

**1-2. Pointer for integers**

Obtain 2 integers from the the standard input, and swap two integers as follows. Use the concepts of pointer in your work. Print your result on the standard output.

|  |
| --- |
| >p1\_2  enter 2 integers  2 3  you entered  2, 3  after swapping  3, 2 |

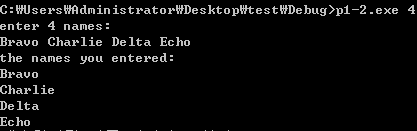
 

* program name: p1\_2.c
* data structure : array of integers
* input : 2 integers separated by space
* output : 2 integers swapped
* condition : use pointers to swap two numbers

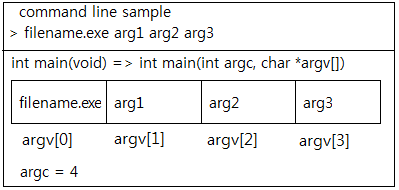
make a function for

**1-3. Dynamic allocation**

Use a command line argument for the total number of students (>2). In the standard input, get *n* (the number you input) names in a single line when the instruction “enter n names:” is given. Use 2D dynamic allocation to store all names in one variable. Print your result in standard output.



* program name: p1\_3.c
* data structure : array of pointers
* input : n names
* output : n names
* condition :
* the number of students should be more than two
* use dynamic allocation to store names (malloc())
* the length of the user name should be up to 30 characters
* no blank space is allowed in the name



**1-4. Function pointer**

Implement a calculator for addition, subtraction, multiplication, and division. Use a command line argument for the operator of your choice, and two numbers. You use the following array to assign operator functions.

int add(int x, int y);

int sub(int x, int y);

int mul(int x, int y);

int div(int x, int y);

int (\*pf[4])(int, int) = { add, sub, mul, div };



* program name: p1\_4.c
* data structure : array of function pointer
* input : the number for the choice of operator (1, 2, 3, or 4),

two integers for the operand

* output : the result of operation