Quiz Section for Program Design (II)

Exercise #5

In this exercise, you will need to write a program to implement a Linked list. You will need to write the six functions listed in the table below.
Here are some special rules of the Linked list:

1. Every node inserted into the linked list has a **unique name and id**.
2. The name of the first node is **"`FOODPAPA`"** whose **id is 0** and the pointer **`HEAD` points to this node**. It has been already written in the source file, **you don't need to do it by yourself**.
3. The first node "`FOODPAPA`" will **not be deleted and swapped**.

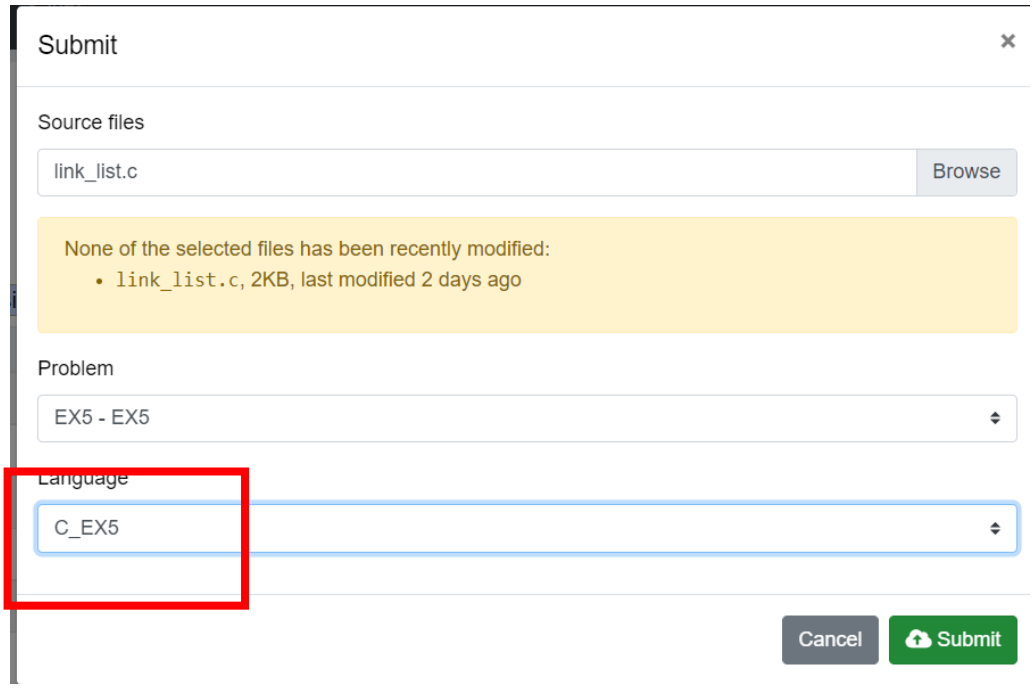
The table below described the corresponding terminal command for each operation (**Operation**), the data type of each parameter of each operation (**Parameter**), and the expected outcomes of each operation (**Task**).

Operation	**Parameter**	**Task**
`INSERT` *ID Name*	*ID* : `int` *Name* : `string`	Insert a new node stored by *Name* after the node whose id is *ID*. If the *ID* doesn't exist in the linked list, print "`INVALID OPERATION`".
`DELETE` *Name*	*Name* : `string`	Delete the node which stores by *Name*. If the node doesn't exit such a node, print "`INVALID OPERATION`".
`SWAP` *Name_1 Name_2*	*Name_1* : `string` *Name_2* : `string`	Swap the nodes named *Name_1* and *Name_2*. If either of the nodes doesn't exist in the linked list, print "`INVALID OPERATION`". **Notice: Don't just swap the value store in the nodes. You need to swap the entire node. You need to break the link and recombine it.**
`PRINT_ID` *Name*	*Name* : `string`	Print the ID of the node by providing the node's name. If the *Name* doesn't exist in linked list, print "`INVALID OPERATION`"
`PRINT_NAME` *ID*	*ID* : `int`	Print the Name of the node given the node's ID. If the *ID* doesn't exist in linked list, print "`INVALID OPERATION`"

`RESULT`	None	Print out the information of all nodes in the linked list in the format "ID NAME" starting from the HEAD pointer.

In this program exercise, you only need to implement the functions. The Input/Output (IO) is already implemented in the source files. We will provide all source files of this program on eCourse2 and you can explore **the structure of the linked list in *myDS.h***. Your task is to implement the source file, *linked_list.c*. **Do not modify any file we provided, excepting** ***link_list.c*.**

You can compile this program using the terminal command "`gcc main.o myIO.o link_list.c`" and entering "`./a.out`" to execute the program. **When you submit your code to DOMjudge, please upload *link_list.c* alone and choose the language "C_EX5" as shown in the figure below.**



The table below shows the example input and output. The integer of the first input (e.g., `2` in the first example) represents the numbers of operations.

Input	Output

`2`
`INSERT 0 LUNE`
`RESULT`
	`0 FOODPAPA`
	`1 LUNE`

`4`
`INSERT 0 LUNE`
`RESULT`
`DELETE LUNE`
	`0 FOODPAPA`
	`1 LUNE`
	`0 FOODPAPA`

RESULT	
4 PRINT_ID FOODPAPA PRINT_NAME 0 PRINT_ID Jay PRINT_NAME 3	0 FOODPAPA INVALID OPERATION INVALID OPERATION
6 INSERT 0 LUNE INSERT 1 AMY INSERT 0 LYON RESULT SWAP AMY LYON RESULT	0 FOODPAPA 3 LYON 1 LUNE 2 AMY 0 FOODPAPA 2 AMY 1 LUNE 3 LYON
11 INSERT 0 Jack INSERT 1 GIANNIS INSERT 2 Middleton RESULT SWAP GIANNIS DURANT SWAP GIANNIS Middleton RESULT DELETE GIANNIS SWAP Jack Middleton RESULT PRINT_ID GIANNIS	0 FOODPAPA 1 Jack 2 GIANNIS 3 Middleton INVALID OPERATION 0 FOODPAPA 1 Jack 3 Middleton 2 GIANNIS 0 FOODPAPA 3 Middleton 1 Jack INVALID OPERATION