|  |
| --- |
| HW8. Self-referential Structures  (Open Source Programming 2019, Spring English class) **Class number: 060**  부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공  201524582  HeeSeok Jeong(정희석)  Submission date: 2019-06-03 |

# Explanation for your codes (60 Points)

1. **Explain of key variables**
2. This time, Main data is consist of linked-list named Node. Node has Contact object and Node pointer for next node. It’s like a Array but not continuous. This program has three Pointer variable that head is pointer of Start of Node , tail is pointer of end of Node. Phonebook is symbolic variable of linked-list of Contact. Explain this below.

**(2) Describe the main data structure**

- linked list for PhoneBook

A. PhoneBook is a node that consist of pointer of next node and object of Contact object, Node pointer head and tail point each first of linked list(phonebook), end of linked list.

(Head) PhoneBook (Tail)

**[Contact info|Node\* next]->[Contact info|Node\* next]->…->[Contact info|Node\* next]**

**스크린샷이(가) 표시된 사진

자동 생성된 설명**

**(3) Explain how to implement functions.**

- if void pointers are used, provide detailed explanations for how void pointers work => Not Use

**-> I delete codes that not using or unnecessary codes**

**- Phone.h**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

-> for line 23 same as homework 7’s phone.h file. Declared function is driver function.

-> ContactNode is linked-list’s Node.

-> PhoneBook is Symbolic variable of Node, head is first Node of PhoneBook, tail is end Node of PhoneBook.

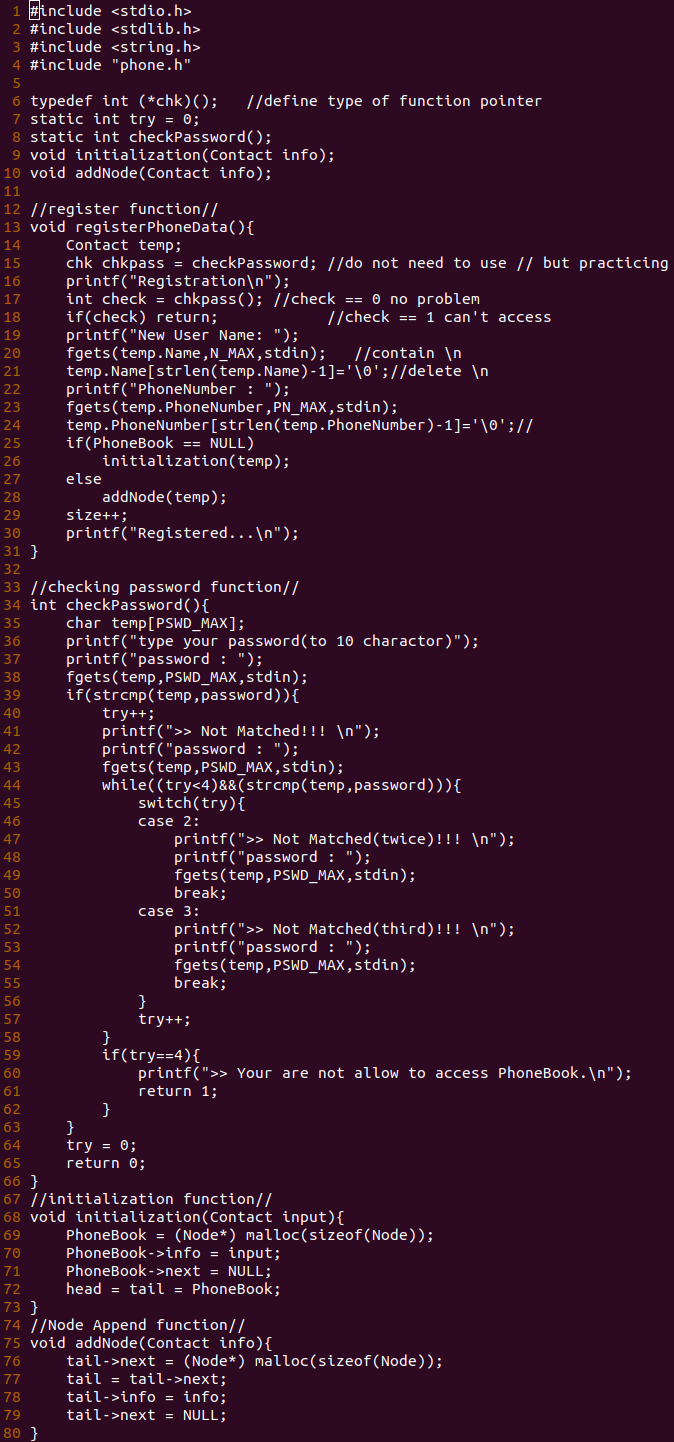
**- hw8Main.c**

**텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명**

* This Code same as Homework 7’s main source code. I’ll skip this code.

**- register.c**

-> I append two function that initalization, and addNode.

-> basically registerPhoneData function and checkPassword function is same as homework7.

-> initialization (Contact input) function initialize Phonebook Node as first node. With Contact input object.

And first initialize process there’s no any initilize, append head, tail’s pointer point to PhoneBook Node.

-> addNode(Contact info) function append Node on tail with Contact info object, first. Memory allocate for next Node, second move to next Node(setting tail pointer), third add Contact object, last initialize next pointer to NULL..

**- print.c**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

-> this is simple. first print Node inp, second move to next, loop this two process until Node arrive to last Node.

**- search.c**

**텍스트이(가) 표시된 사진

자동 생성된 설명**

-> searchByName similar to homework 7’s function, append searchNode(char\* name)

-> searchNode compare name with input, Node search start on Phonebook, strcmp compare name to Node’s name, if name correct, print that Node’s Contact information, if not correct go to next Node until arrive to Node’s end, if move to Node’s end, return 0, this case print incorrect information to user on searchByName function.

**- delete.c**

텍스트이(가) 표시된 사진

자동 생성된 설명-> using function pointer type \*dnf with char\* argument for deleteNode function.

-> deleteNode get name for deleted. And search Node from head (PhoneBook) to end of Node.

-> compare name to Node’s name, if incorrect, move to next Node, if correct, check Node’s position, if head position or tail position, relocation this Node, and previous Node’s next pointer connect to removed Node’s next Node. after this process, free memory of this Node.

**텍스트이(가) 표시된 사진

자동 생성된 설명- sort.c**

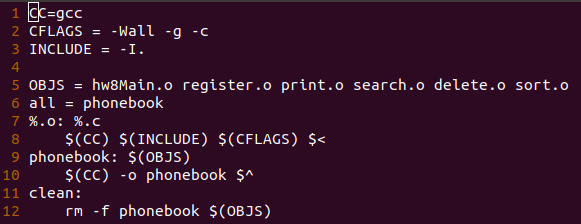
-> I use Bubble sort because Homework 7 use this sort algorithm. And It is easy for implement.

-> using three Pointer of Node, and two while loop, while start move to end Node, check two Node’s name with strcmp

-> if Next Node’s name’s Alphabet(based on ASCII) smaller than Current Node’s Alphabet, swap two Node’s Contact object to ascending order.

-> There’s No change of Node’s pointer. And Node pointer Move to end of Node. repeat this process.

**(4) Describe makefile**



* same with hw8 and change hw7Main.o to hw8Main.o
* using macro variable, and special macro variable
* each source code with .c file compile to object file first, and link with -o option and create executable file phonebook.
* SIMILAR TO HOMEWORK 7

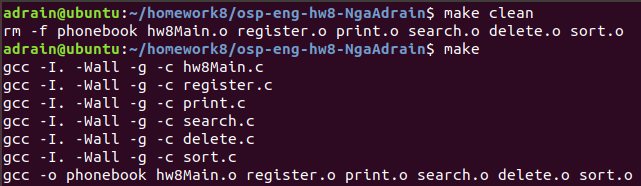
# Program Execution (10 points)

1. **Types of operating systems and compilers used**
2. OS : Ubuntu 18.04.2 LTS based on debian linux

GCC : gcc (Ubuntu 7.3.0-27ubuntu1~18.04) 7.3.0

=> I can check Ubuntu version with ‘cat /etc/\*release’ and gcc version with ‘gcc –version’ command

1. **How to compile and execute the program**
2. I can compile with gcc and make makefile & use make command -> make executable file (named phonebook) -> execute ./phonebook



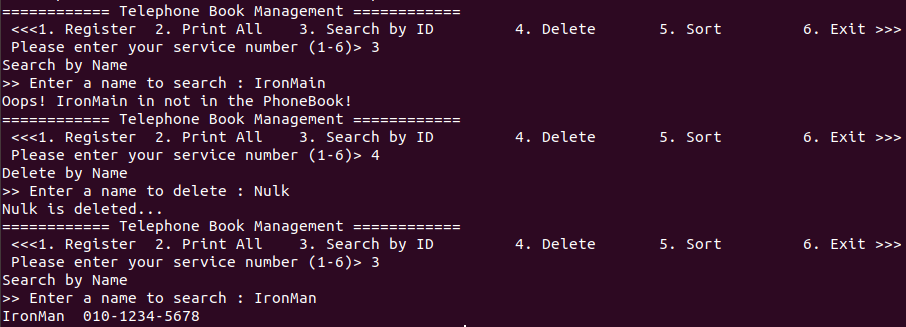
1. **Include a screen capture for illustrating how the program works**
2. Register and Print

스크린샷이(가) 표시된 사진

자동 생성된 설명

* I can execute register process, and checking password, print with pointer’s address, name, phone number.

1. Search and delete

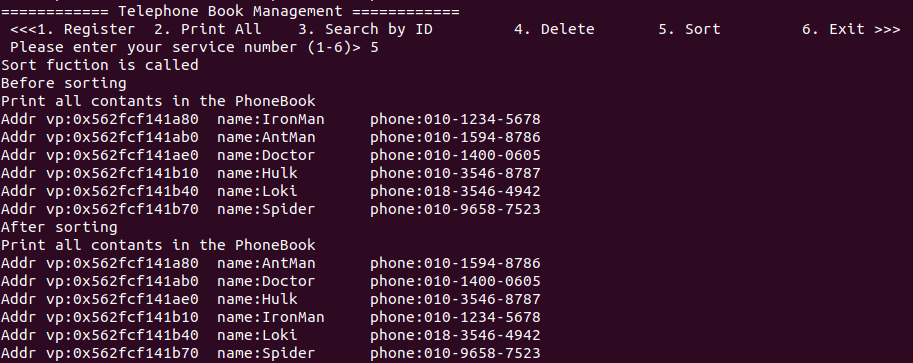


* I can search with name, delete with name, if name not exist on Contact, print error message

※ For check delete correction on try to delete Nulk Object, this exist on upper screenshot

and not exist on below screen shot. (Because Nulk is mistyping of Hulk, I try to delete)

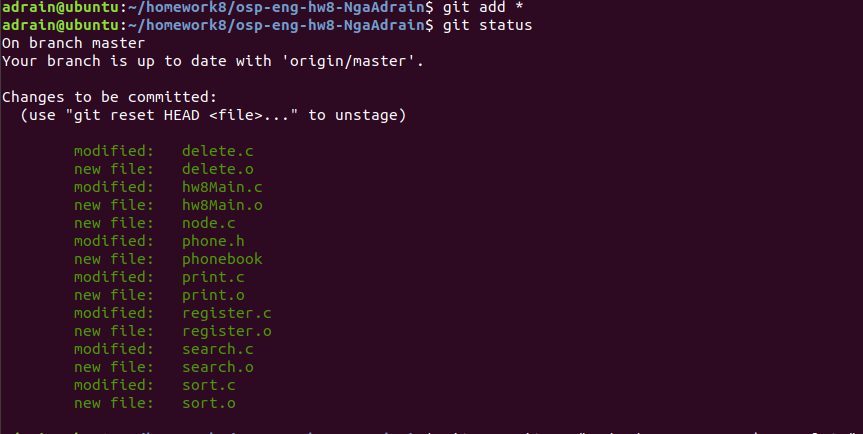
1. Sort



* After delete Nulk, I append AntMan, Doctor, Hulk, Loki, Spider, and execute sorting process, Phonebook successfully sorted to ascending order of ASCII rule.

# Github repository (20 points)

1. **include github commands for cloning, adding, committing and pushing**

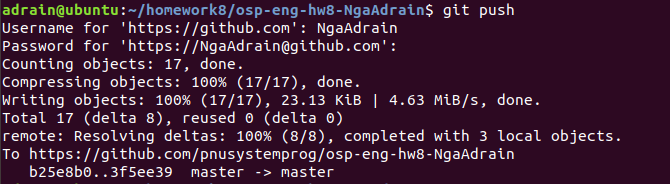


* First, add all changed, created file on tract with command ‘git add \*’(wild card), check status of git with ‘git status’

스크린샷이(가) 표시된 사진

자동 생성된 설명

* Second, Commit all file on track to local repository with command git commit, and give -m option, add comment.



* Finally, push all file from local repository to remote repository, with login github.

1. **After pushing the source code and the makefile to your github directory, include the screen capture image of the Github repository**

스크린샷이(가) 표시된 사진

자동 생성된 설명

* This is result of commit to github.

# Discussion (10 points)

**- What you learned while doing your homework (contents other than class hours),**

A. This homework can exercise making linked list replace array, using linked list that make, add, delete, search, sort. And I think this homework unnecessary to use void pointer, because of using linked list, that need to use pointer, and each pointer has pointer type. That is why I not use void pointer this time. Last I can exercise too memory allocation and deallocation with malloc, free function.

**- Describe difficulties during homework**

A. This is not a big problem, there’s no template code, when accept this homework first, I feel confuse what should I do first? And first, copy my homework7’s code to this project. Second, modify makefile. Third create node.c file for node operation. But I think node operation’s declaration, and implement can be implemented on each source code(print.c, register.c, … etc). and remove node.c and after finish function implementation, remove code that about homework6, 7 and not using code.

**Thanks for reading my report and sorry about my poor English.**