HW7: Function pointer and void pointer	
	Class number: 060
	Department Name: 전기컴퓨터공학부
	201800234
	후판칭
	Submission date: 2019-05-27
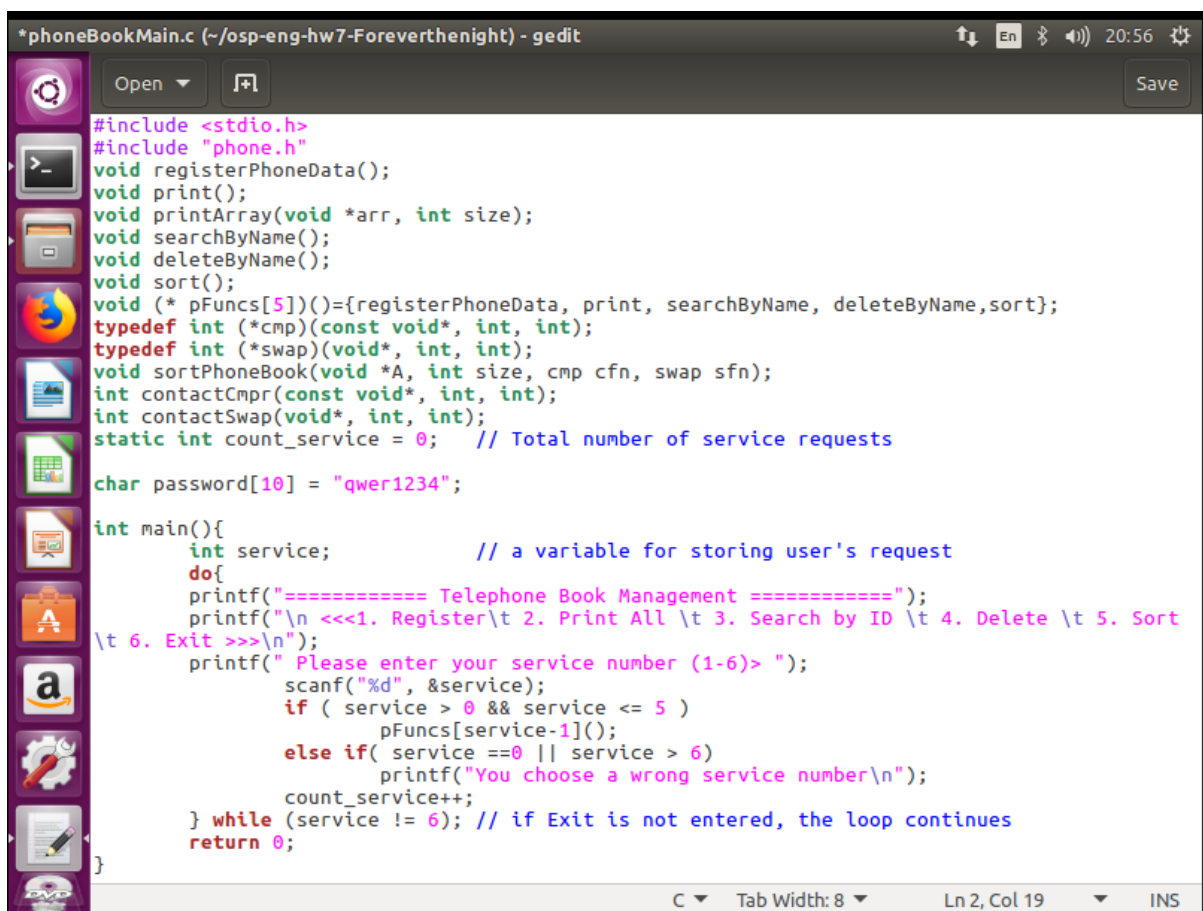
1. Explanation for your codes (60 Points)

phoneBookMain.c runs mostly by invoking functions of other .c files, so it contains extern to use registerPhoneData(), printAll(), searchByName(), deleteByName() of other files.

In the main function of phoneBookMain.c, I is used of storing user's request. Here is a do-while circulation to make sure it running until users make it stop.

```c
#include <stdio.h>
#include <string.h>
#include "phone.h"

extern struct Contact PhoneBook[MAX];
extern char password[10];

void registerPhoneData(){
        char inputPassword[100];
        int wrongCount = 0;
        do{
                printf("Password: ");
                scanf("%s", inputPassword);

                if (strcmp(password, inputPassword) != 0){
                        wrongCount++;
                        switch(wrongCount){
                                case 1: printf(">>Not Matched !!!\n"); break;
                                case 2: printf(">>Not Matched (twice) !!!\n"); break;
                                case 3: printf(">>Not Matched (3 times) !!!\n"); break;
                        }
                }

                if (wrongCount == 3){
                        printf(">>You are not allowed to access PhoneBook.\n");
                        return ;
                }
        } while (strcmp(password, inputPassword) != 0);

        printf("New User Name: ");
        scanf("%s", PhoneBook[size].Name);
        printf("PhoneNumber: ");
        scanf("%s", PhoneBook[size].PhoneNumber);
        printf("Registered...\n");
        size++;

        return ;
}
```

There is two parts in regidter.c, firstly determine if input password is true, and then register user information if password is true.

Inputpassword[100] stores the password inputted by user, wrongCount stores times that password wrong.

```c
#include <stdio.h>
#include "phone.h"

extern struct Contact PhoneBook[MAX];

void print(){
        printf("Print all contants in the PhoneBook\n");
        printArray(PhoneBook,count_service);
}
void printArray(void *arr, int size){
        int i;
        void *p;
        p= arr;
        for(i=0; i<size; i++){
                printf("Addr vp:%p\t", (Contact*)p+i);
                printf("name:%s\t phone:%s\n", ((Contact*)p+i)->Name, ((Contact*)p+i)->PhoneNumber);
        }
        printf("\n");
}
```
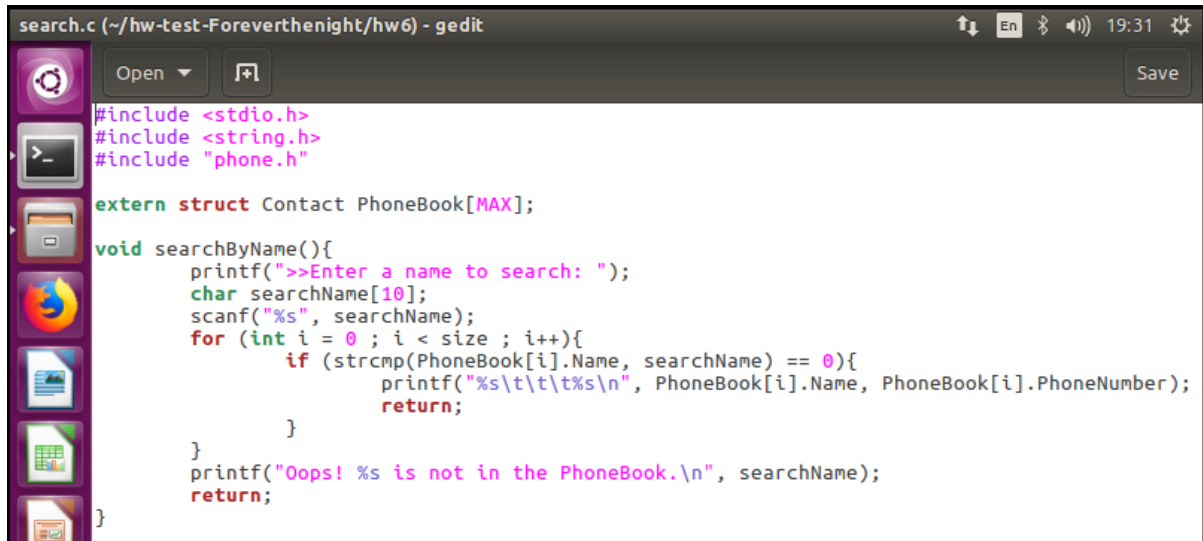
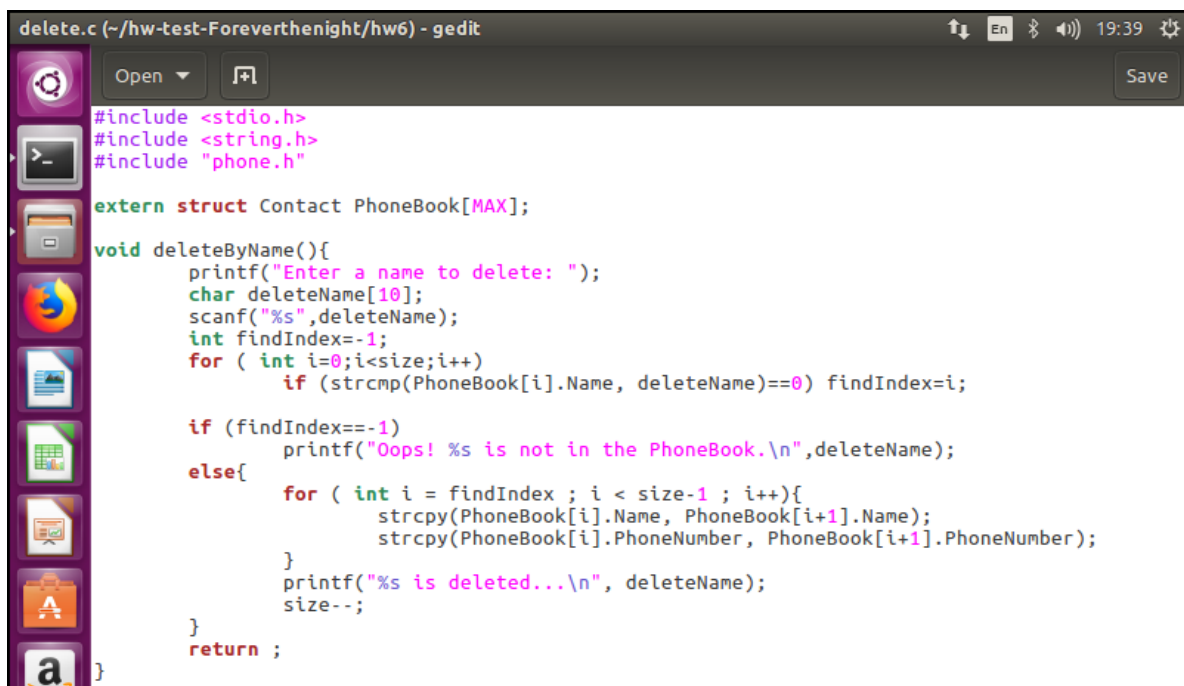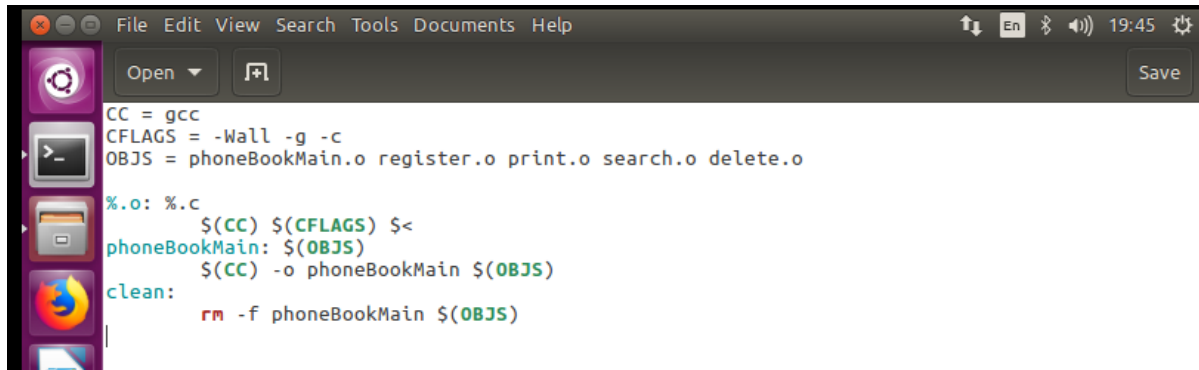Calling printArray to print all rhe contacts.

```c
search.c (~/hw-test-Foreverthenight/hw6) - gedit

#include <stdio.h>
#include <string.h>
#include "phone.h"

extern struct Contact PhoneBook[MAX];

void searchByName(){
        printf(">>Enter a name to search: ");
        char searchName[10];
        scanf("%s", searchName);
        for (int i = 0 ; i < size ; i++){
                if (strcmp(PhoneBook[i].Name, searchName) == 0){
                        printf("%s\t\t\t%s\n", PhoneBook[i].Name, PhoneBook[i].PhoneNumber);
                        return;
                }
        }
        printf("Oops! %s is not in the PhoneBook.\n", searchName);
        return;
}
```

In the searchByName(), for circulation let the name compared one by one, and then print the matched name.

```c
delete.c (~/hw-test-Foreverthenight/hw6) - gedit

#include <stdio.h>
#include <string.h>
#include "phone.h"

extern struct Contact PhoneBook[MAX];

void deleteByName(){
        printf("Enter a name to delete: ");
        char deleteName[10];
        scanf("%s",deleteName);
        int findIndex=-1;
        for ( int i=0;i<size;i++)
                if (strcmp(PhoneBook[i].Name, deleteName)==0) findIndex=i;

        if (findIndex==-1)
                printf("Oops! %s is not in the PhoneBook.\n",deleteName);
        else{
                for ( int i = findIndex ; i < size-1 ; i++){
                        strcpy(PhoneBook[i].Name, PhoneBook[i+1].Name);
                        strcpy(PhoneBook[i].PhoneNumber, PhoneBook[i+1].PhoneNumber);
                }
                printf("%s is deleted...\n", deleteName);
                size--;
        }
        return ;
}
```

First of all, search if there exist a matched name, and delete the matched information, print o ut error message if not.
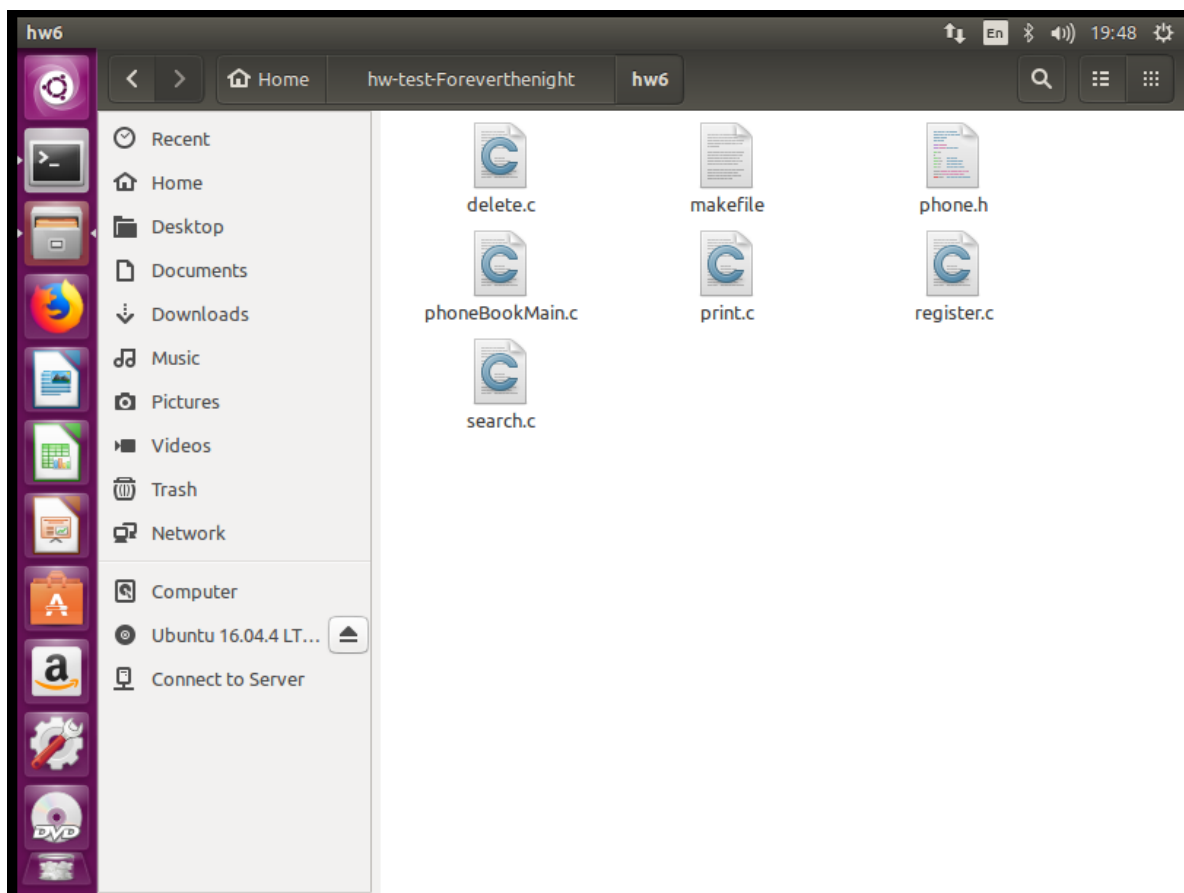
```
CC = gcc
CFLAGS = -Wall -g -c
OBJS = phoneBookMain.o register.o print.o search.o delete.o

%.o: %.c
        $(CC) $(CFLAGS) $<
phoneBookMain: $(OBJS)
        $(CC) -o phoneBookMain $(OBJS)
clean:
        rm -f phoneBookMain $(OBJS)
```

In the first two lines, using macro the replays some long options.

%.o:%.c means all the .c files is prerequisite of the corresponding .o files.

2. Program Execution (10 points)

3. Github repository(20 points)

4. Discussion (10 points)

Reason:

CRLF : linebreak of windows

LF : linebreak of linux

Solution:

Using command `git config core.autocrlf false` to make it work.