HW7. Function pointer and void pointer

(Open Source Programming 2019, Spring English class) **Class number: 060**

부산대학교 전기컴퓨터공학부 정보컴퓨터공학전공

201524582

HeeSeok Jeong(정희석)

Submission date: 2019-05-27

1. Explanation for your codes (60 Points)

(1) Explain of key variables: function pointers should be explained.

A. This time, hw7main.c has function pointers array -> this replace switch – case condition, register.c has integer function pointer for expressing checkpassword function, delete.c has void function pointer for expressing deleteNSort function. And basically sort.c has integer function pointers for expressing contactCompare with cmp and contactSwap with swap.

(2) Describe the main data structure

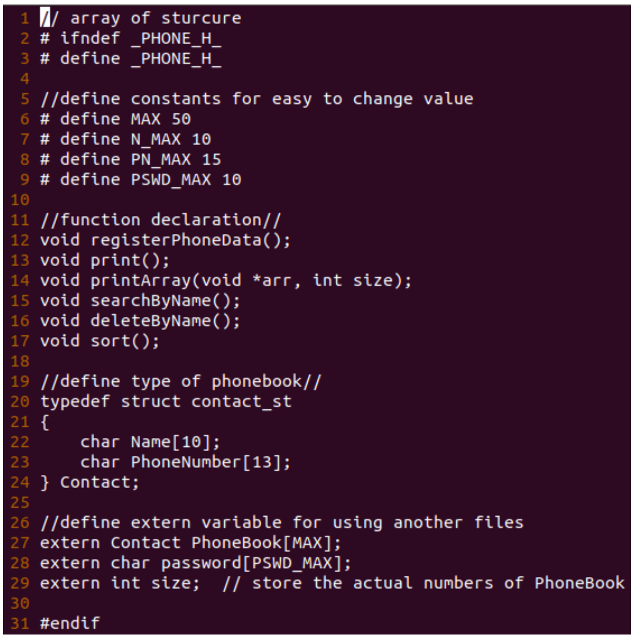
A. I use structure type array structure, that named PhoneBook[MAX=50]; Because This Program reuse code of HW6, I used array in HW6. This array is composed of this structure.

[Name[10byte]**|**PhoneNumber[15byte]**]**|…|**[**Name[10byte]**|**PhoneNumber[15byte]**]**

(3) Explain how to implement functions.

- if void pointers are used, provide detailed explanations for how void pointers work

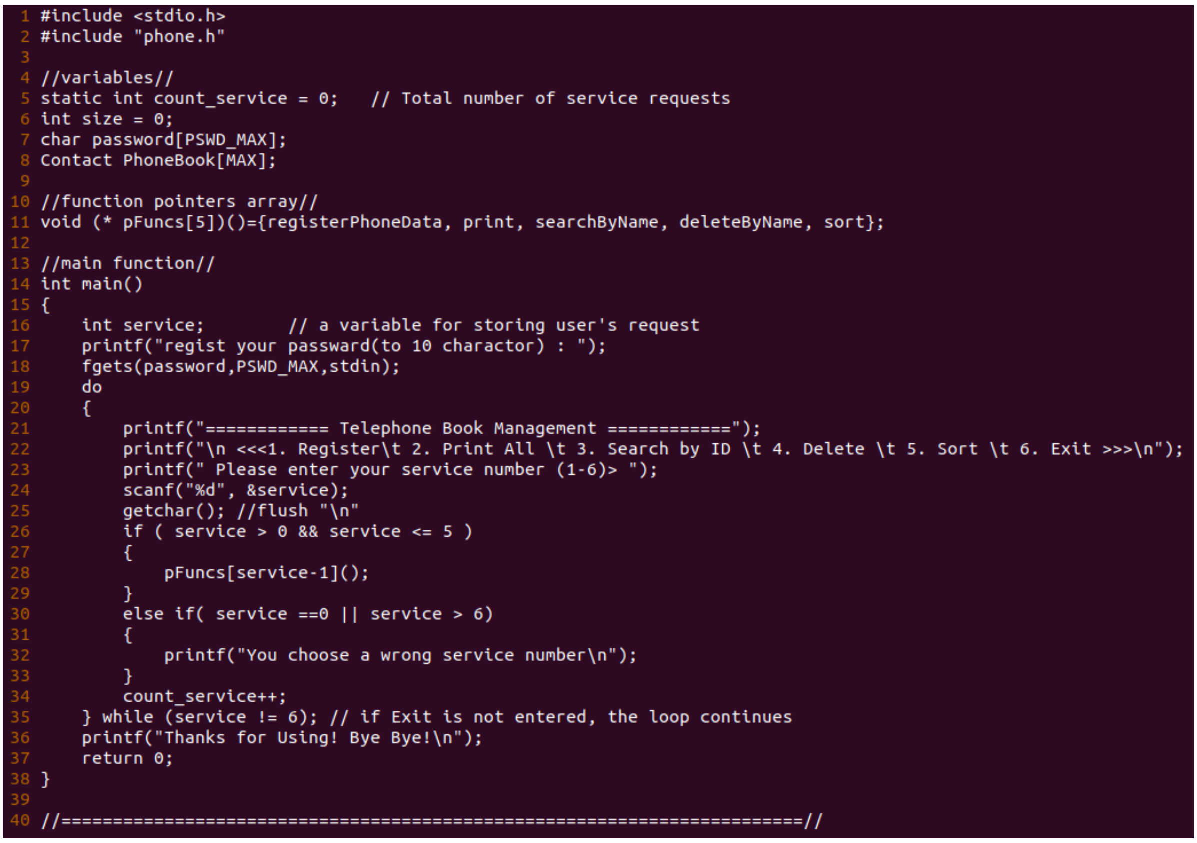
- Phone.h



```
// array of sturcure
# ifndef _PHONE_H_
# define _PHONE_H_

//define constants for easy to change value
# define MAX 50
# define N_MAX 10
# define PN_MAX 15
# define PSWD_MAX 10

//function declaration//
void registerPhoneData();
void print();
void printArray(void *arr, int size);
void searchByName();
void deleteByName();
void sort();

//define type of phonebook//
typedef struct contact_st
{
    char Name[10];
    char PhoneNumber[13];
} Contact;

//define extern variable for using another files
extern Contact PhoneBook[MAX];
extern char password[PSWD_MAX];
extern int size;  // store the actual numbers of PhoneBook

#endif
```

➔ this header file contain basic variable of this Program, and function's declaration. Implement PhoneBook's base structure contact_st, Define Contact array PhoneBook with extern variable, can store 50 Contacts. Char array password, int size store current array size and define several Constant with number for easily change number of elements

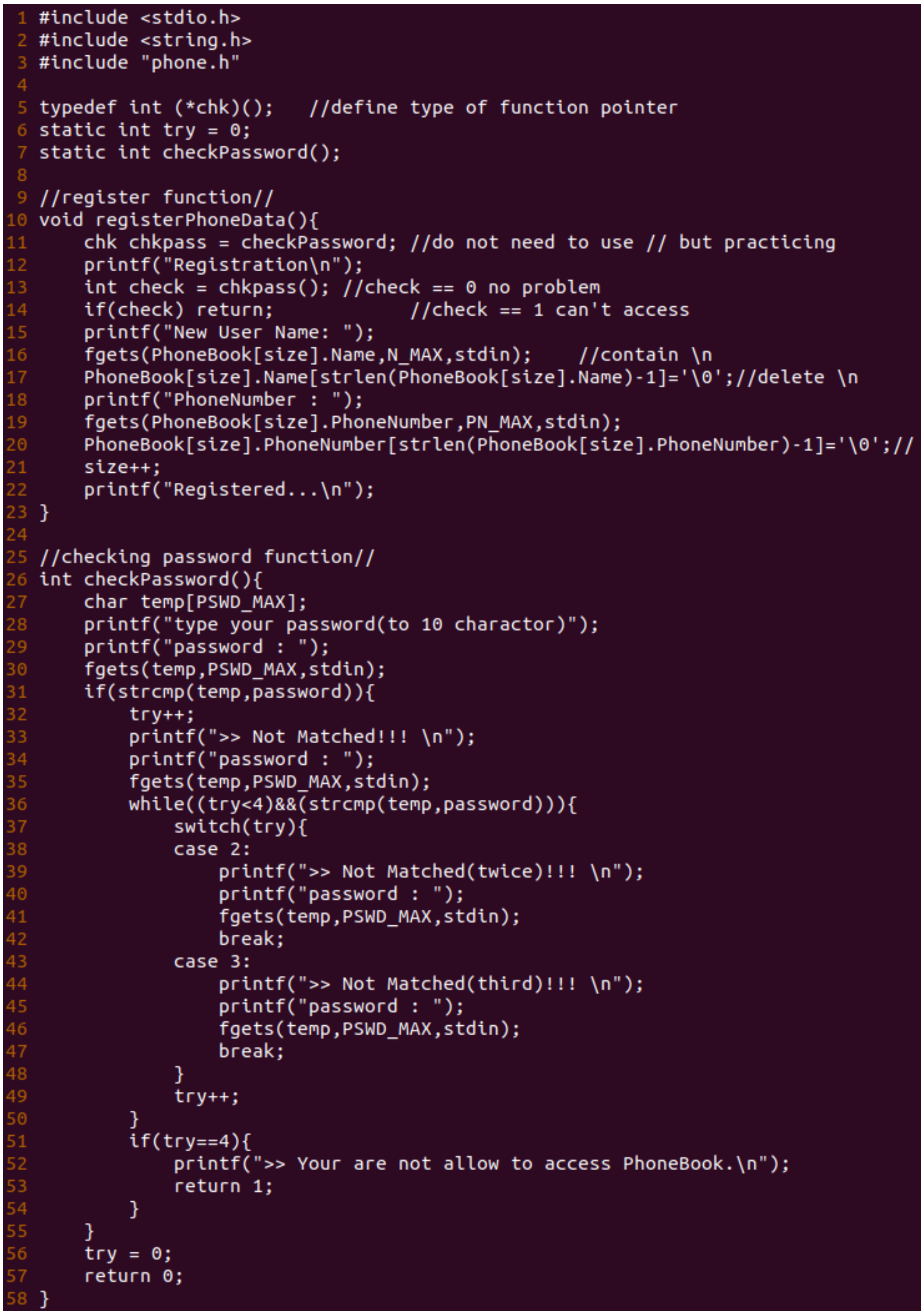
1^{st}. Main



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

//variables//
static int count_service = 0;   // Total number of service requests
int size = 0;
char password[PSWD_MAX];
Contact PhoneBook[MAX];

//function pointers array//
void (* pFuncs[5])()={registerPhoneData, print, searchByName, deleteByName, sort};

//main function//
int main()
{
    int service;        // a variable for storing user's request
    printf("regist your passward(to 10 charactor) : ");
    fgets(password,PSWD_MAX,stdin);
    do
    {
        printf("============ Telephone Book Management ============");
        printf("\n <<<1. Register\t 2. Print All \t 3. Search by ID \t 4. Delete \t 5. Sort \t 6. Exit >>>\n");
        printf(" Please enter your service number (1-6)> ");
        scanf("%d", &service);
        getchar(); //flush "\n"
        if ( service > 0 && service <= 5 )
        {
            pFuncs[service-1]();
        }
        else if( service ==0 || service > 6)
        {
            printf("You choose a wrong service number\n");
        }
        count_service++;
    } while (service != 6); // if Exit is not entered, the loop continues
    printf("Thanks for Using! Bye Bye!\n");
    return 0;
}

//=========================================================================//
```

- ➔ Using static integer variables for counting service, integer variables for storing phonebook's size, character array for password, and structure Contact array for phonebook with 50 elements.
- ➔ And Using void function pointers array -> replace switch-case condition, can execute function directly. -> pFuncs[0] = register, pFuncs[1] = print, pFuncs[2] = search, pFuncs[3] = delete, pFuncs[4] = sort
- ➔ First main request input password from user for using check process on register process.
- ➔ Using do-while loop, until service input not same with 6, each function continuously execute.

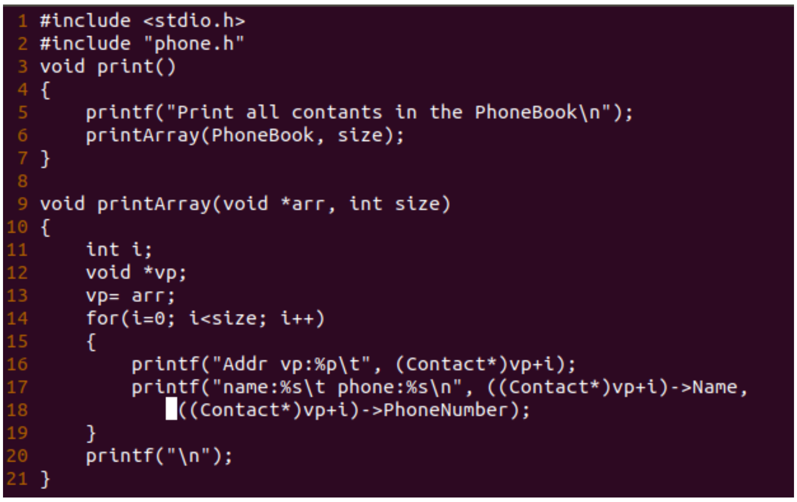
2nd. Register.c -> same with hw6's register.c



```
1  #include <stdio.h>
2  #include <string.h>
3  #include "phone.h"
4
5  typedef int (*chk)();   //define type of function pointer
6  static int try = 0;
7  static int checkPassword();
8
9  //register function//
10 void registerPhoneData(){
11     chk chkpass = checkPassword; //do not need to use // but practicing
12     printf("Registration\n");
13     int check = chkpass(); //check == 0 no problem
14     if(check) return;            //check == 1 can't access
15     printf("New User Name: ");
16     fgets(PhoneBook[size].Name,N_MAX,stdin);    //contain \n
17     PhoneBook[size].Name[strlen(PhoneBook[size].Name)-1]='\0';//delete \n
18     printf("PhoneNumber : ");
19     fgets(PhoneBook[size].PhoneNumber,PN_MAX,stdin);
20     PhoneBook[size].PhoneNumber[strlen(PhoneBook[size].PhoneNumber)-1]='\0';//
21     size++;
22     printf("Registered...\n");
23 }
24
25 //checking password function//
26 int checkPassword(){
27     char temp[PSWD_MAX];
28     printf("type your password(to 10 charactor)");
29     printf("password : ");
30     fgets(temp,PSWD_MAX,stdin);
31     if(strcmp(temp,password)){
32         try++;
33         printf(">> Not Matched!!! \n");
34         printf("password : ");
35         fgets(temp,PSWD_MAX,stdin);
36         while((try<4)&&(strcmp(temp,password))){
37             switch(try){
38             case 2:
39                 printf(">> Not Matched(twice)!!! \n");
40                 printf("password : ");
41                 fgets(temp,PSWD_MAX,stdin);
42                 break;
43             case 3:
44                 printf(">> Not Matched(third)!!! \n");
45                 printf("password : ");
46                 fgets(temp,PSWD_MAX,stdin);
47                 break;
48             }
49             try++;
50         }
51         if(try==4){
52             printf(">> Your are not allow to access PhoneBook.\n");
53             return 1;
54         }
55     }
56     try = 0;
57     return 0;
58 }
```

- ➔ Using integer function pointer chk-> with no argument function pointer -> It will use for CheckPassword() function.
- ➔ this code role register. And assignment require check password process, I append checkPassword function, devide password checking process from registration process
- ➔ I use fgets function instead of gets function because of checking string size. And append Line 17, Line 20 -> prevent print ₩n charactor.
- ➔ Use while loop and if condition, compare password and check trial count

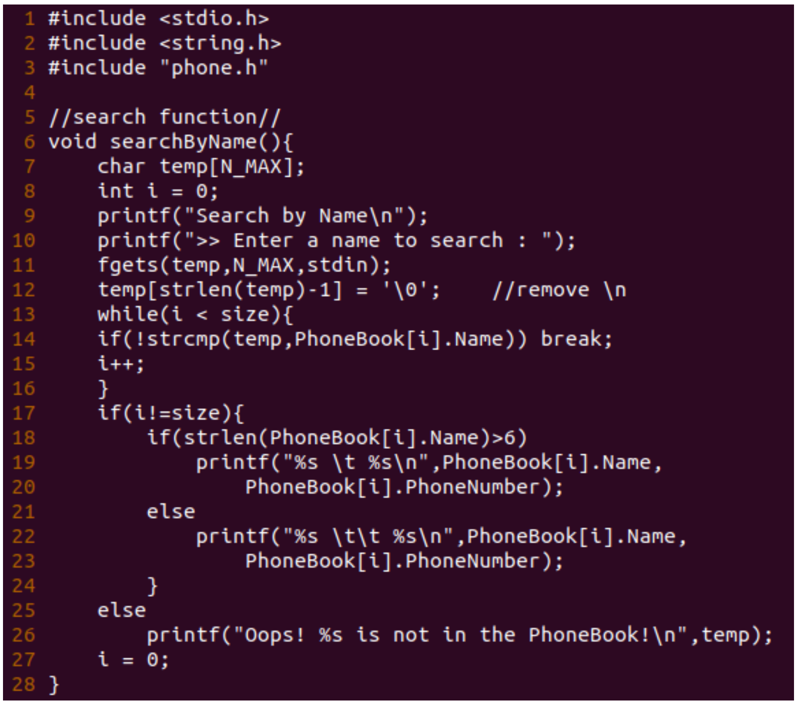
3rd. print.c



```
#include <stdio.h>
#include "phone.h"
void print()
{
    printf("Print all contants in the PhoneBook\n");
    printArray(PhoneBook, size);
}

void printArray(void *arr, int size)
{
    int i;
    void *vp;
    vp= arr;
    for(i=0; i<size; i++)
    {
        printf("Addr vp:%p\t", (Contact*)vp+i);
        printf("name:%s\t phone:%s\n", ((Contact*)vp+i)->Name,
            ((Contact*)vp+i)->PhoneNumber);
    }
    printf("\n");
}
```

➔ Print array's information. in printArray function, using void pointer *arr -> just store memory address, this pointer don't know type of this address, then we need to type cast for using with (Contact*)vp+i, and this time printArray print void pointer's address value.

4th. Search.c -> same with hw6's search.c

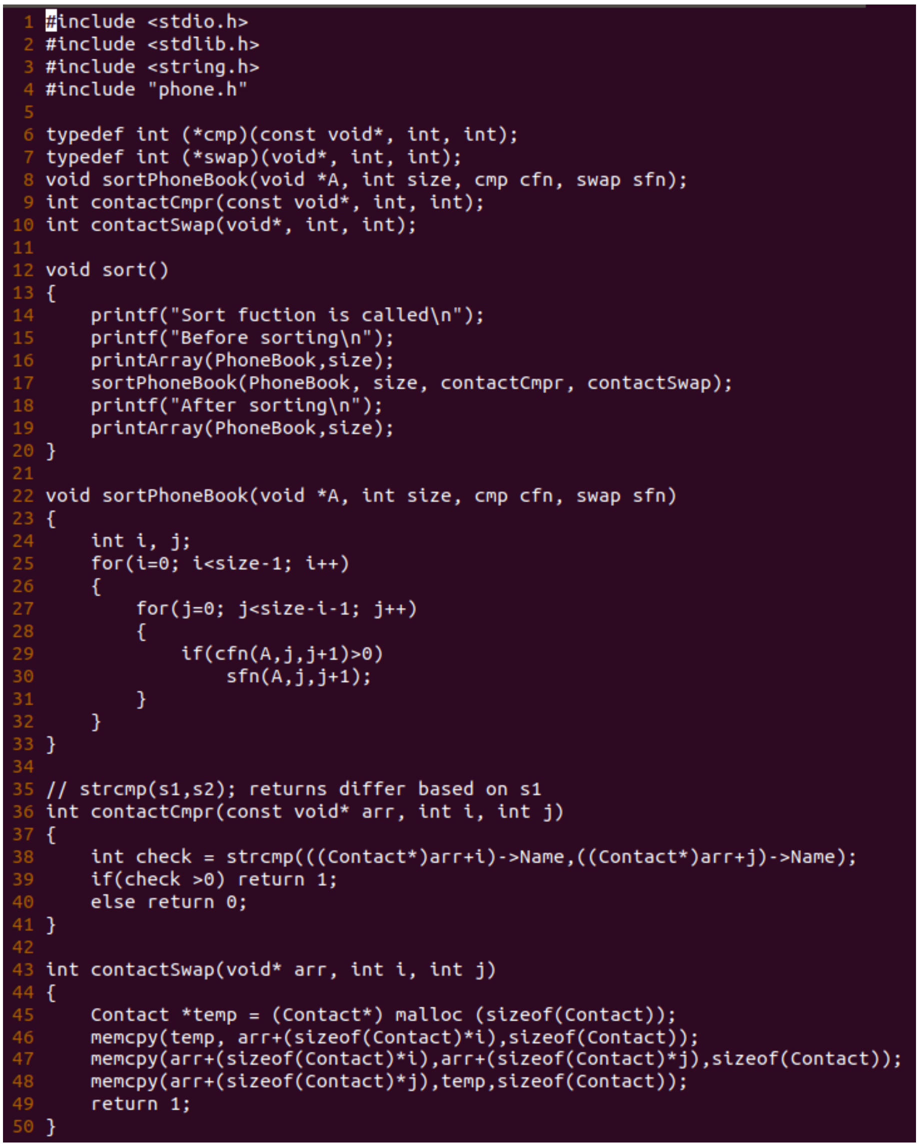


```
#include <stdio.h>
#include <string.h>
#include "phone.h"

//search function//
void searchByName(){
    char temp[N_MAX];
    int i = 0;
    printf("Search by Name\n");
    printf(">> Enter a name to search : ");
    fgets(temp,N_MAX,stdin);
    temp[strlen(temp)-1] = '\0';    //remove \n
    while(i < size){
    if(!strcmp(temp,PhoneBook[i].Name)) break;
    i++;
    }
    if(i!=size){
        if(strlen(PhoneBook[i].Name)>6)
            printf("%s \t %s\n",PhoneBook[i].Name,
                PhoneBook[i].PhoneNumber);
        else
            printf("%s \t\t %s\n",PhoneBook[i].Name,
                PhoneBook[i].PhoneNumber);
        }
    else
        printf("Oops! %s is not in the PhoneBook!\n",temp);
    i = 0;
}
```

➔ - I use variable for store string name for comparing name in PhoneBook, and index, using fgets store name for search, compare with strcmp function in string.h

➔ If find same string in phonebook, break loop and then index not same with size -> matched case! All loop complete –> index same with size -> that case mismatch

5th. sort.c



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

typedef int (*cmp)(const void*, int, int);
typedef int (*swap)(void*, int, int);
void sortPhoneBook(void *A, int size, cmp cfn, swap sfn);
int contactCmpr(const void*, int, int);
int contactSwap(void*, int, int);

void sort()
{
    printf("Sort fuction is called\n");
    printf("Before sorting\n");
    printArray(PhoneBook,size);
    sortPhoneBook(PhoneBook, size, contactCmpr, contactSwap);
    printf("After sorting\n");
    printArray(PhoneBook,size);
}

void sortPhoneBook(void *A, int size, cmp cfn, swap sfn)
{
    int i, j;
    for(i=0; i<size-1; i++)
    {
        for(j=0; j<size-i-1; j++)
        {
            if(cfn(A,j,j+1)>0)
                sfn(A,j,j+1);
        }
    }
}

// strcmp(s1,s2); returns differ based on s1
int contactCmpr(const void* arr, int i, int j)
{
    int check = strcmp(((Contact*)arr+i)->Name,((Contact*)arr+j)->Name);
    if(check >0) return 1;
    else return 0;
}

int contactSwap(void* arr, int i, int j)
{
    Contact *temp = (Contact*) malloc (sizeof(Contact));
    memcpy(temp, arr+(sizeof(Contact)*i),sizeof(Contact));
    memcpy(arr+(sizeof(Contact)*i),arr+(sizeof(Contact)*j),sizeof(Contact));
    memcpy(arr+(sizeof(Contact)*j),temp,sizeof(Contact));
    return 1;
}
```

-> define integer function pointer cmp with arguments that constant void pointer, integer, integer, for contactCmpr function and integer function pointer swap with arguments that void pointer, integer, interger for contactSwap function.

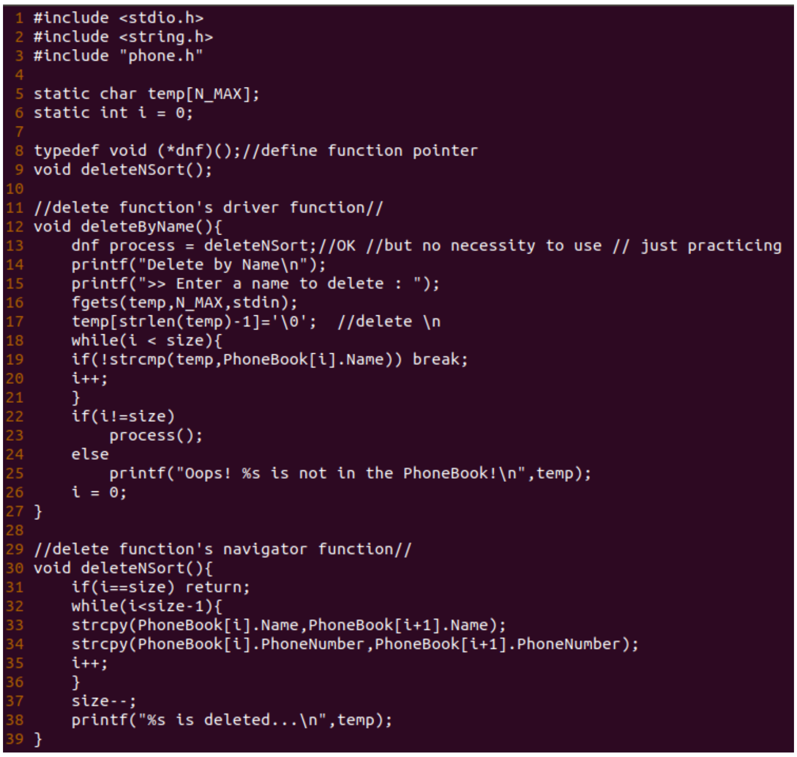
-> sort() function is driver function, sortPhoneBook function is navigator function. -> void pointer just get address value and not know type of address's value. -> I need to use type casting for using.

-> then I use type casting on contactCmpr with strcmp function in string.h, and strcmp return integer that differ between first argument and second argument, and standard is first argument.

-> strcmp >0 : first argument is bigger than(later than) second argument.

-> contactSwap use memcpy for all of contents swap each with using temp pointer with memory allocation.

6th delete.c -> same with hw6's delete.c

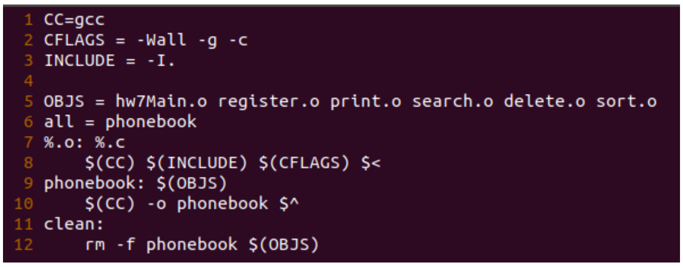


```c
#include <stdio.h>
#include <string.h>
#include "phone.h"

static char temp[N_MAX];
static int i = 0;

typedef void (*dnf)();//define function pointer
void deleteNSort();

//delete function's driver function//
void deleteByName(){
    dnf process = deleteNSort;//OK //but no necessity to use // just practicing
    printf("Delete by Name\n");
    printf(">> Enter a name to delete : ");
    fgets(temp,N_MAX,stdin);
    temp[strlen(temp)-1]='\0';  //delete \n
    while(i < size){
    if(!strcmp(temp,PhoneBook[i].Name)) break;
    i++;
    }
    if(i!=size)
        process();
    else
        printf("Oops! %s is not in the PhoneBook!\n",temp);
    i = 0;
}

//delete function's navigator function//
void deleteNSort(){
    if(i==size) return;
    while(i<size-1){
    strcpy(PhoneBook[i].Name,PhoneBook[i+1].Name);
    strcpy(PhoneBook[i].PhoneNumber,PhoneBook[i+1].PhoneNumber);
    i++;
    }
    size--;
    printf("%s is deleted...\n",temp);
}
```

<similar to search.c>

- ➔ I use static variable for store string name for comparing name in PhoneBook, and index (originally I define static int index but string.h has char index -> crash!). -> I devide delete and sort section from search and delete
- ➔ Define void function pointer type *dnf for deleteNSort function.
- ➔ Pointer process express deleteNSort() function.
- ➔ Deletebyname -> search name same with input name
- ➔ If find name on phonebook -> execute deletensort , Else -> exit.
- ➔ deleteNSort() use strcpy's copy feature and remove previous data, search matched index on static variable i, from that index to end of array, strcpy from next element to previous element and finally decrease size.

(4) Describe makefile



```makefile
CC=gcc
CFLAGS = -Wall -g -c
INCLUDE = -I.

OBJS = hw7Main.o register.o print.o search.o delete.o sort.o
all = phonebook
%.o: %.c
	$(CC) $(INCLUDE) $(CFLAGS) $<
phonebook: $(OBJS)
	$(CC) -o phonebook $^
clean:
	rm -f phonebook $(OBJS)
```

- ➔ same with hw6 and change phoneBookMain.o to hw7Main.o, and append sort.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.

2. Program Execution (10 points)

(1) Types of operating systems and compilers used

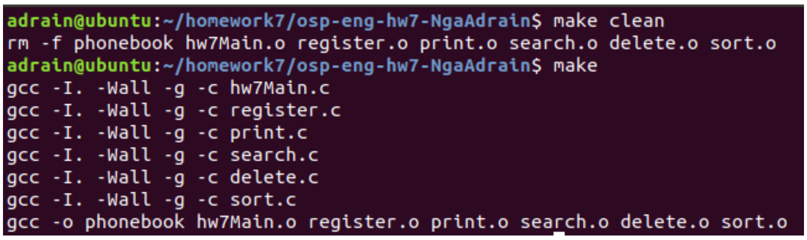
A. OS : Ubuntu 18.04.2 LTS based on debian linux

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

(2) How to compile and execute the program

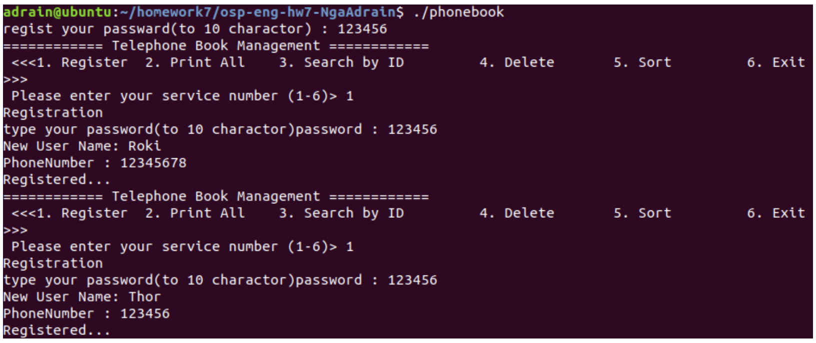
A. I can compile with gcc and make makefile & use make command -> make executable file(named phonebook) -> execute ./phonebook



```
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ make clean
rm -f phonebook hw7Main.o register.o print.o search.o delete.o sort.o
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ make
gcc -I. -Wall -g -c hw7Main.c
gcc -I. -Wall -g -c register.c
gcc -I. -Wall -g -c print.c
gcc -I. -Wall -g -c search.c
gcc -I. -Wall -g -c delete.c
gcc -I. -Wall -g -c sort.c
gcc -o phonebook hw7Main.o register.o print.o search.o delete.o sort.o
```

(3) Include a screen capture for illustrating how the program works

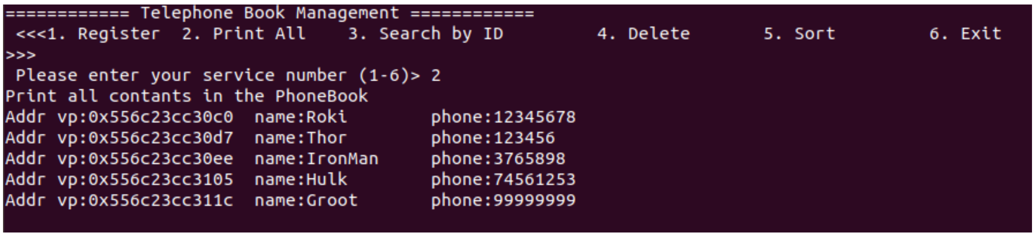
1. register



```
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ ./phonebook
regist your passward(to 10 charactor) : 123456
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID         4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 1
Registration
type your password(to 10 charactor)password : 123456
New User Name: Roki
PhoneNumber : 12345678
Registered...
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID         4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 1
Registration
type your password(to 10 charactor)password : 123456
New User Name: Thor
PhoneNumber : 123456
Registered...
```

-> first execute program, register password, and check password. And input name, and phonenumber.

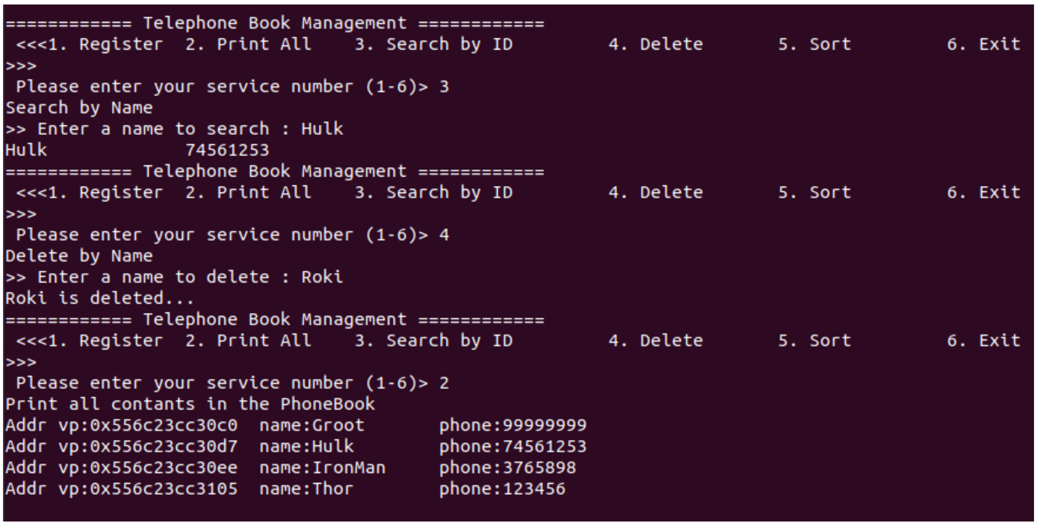
2. print



```
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID         4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 2
Print all contants in the PhoneBook
Addr vp:0x556c23cc30c0  name:Roki       phone:12345678
Addr vp:0x556c23cc30d7  name:Thor       phone:123456
Addr vp:0x556c23cc30ee  name:IronMan    phone:3765898
Addr vp:0x556c23cc3105  name:Hulk       phone:74561253
Addr vp:0x556c23cc311c  name:Groot      phone:99999999
```

-> all input printed with memory address, name, phonenumber.

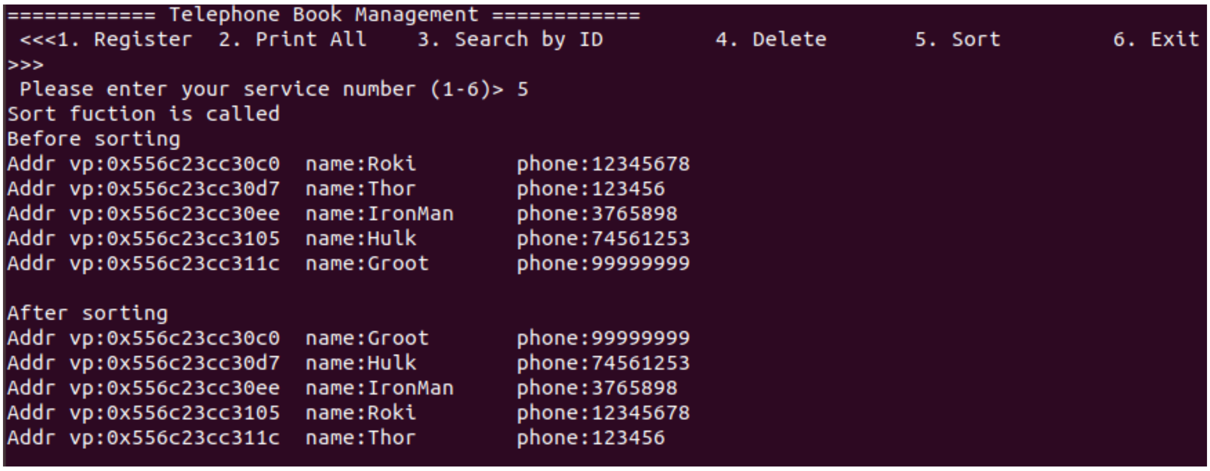
3. search and delete



```
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID        4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 3
Search by Name
>> Enter a name to search : Hulk
Hulk             74561253
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID        4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 4
Delete by Name
>> Enter a name to delete : Roki
Roki is deleted...
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID        4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 2
Print all contants in the PhoneBook
Addr vp:0x556c23cc30c0  name:Groot       phone:99999999
Addr vp:0x556c23cc30d7  name:Hulk        phone:74561253
Addr vp:0x556c23cc30ee  name:IronMan     phone:3765898
Addr vp:0x556c23cc3105  name:Thor        phone:123456
```

-> search with name, delete with name -> successfully executed.

4. sort

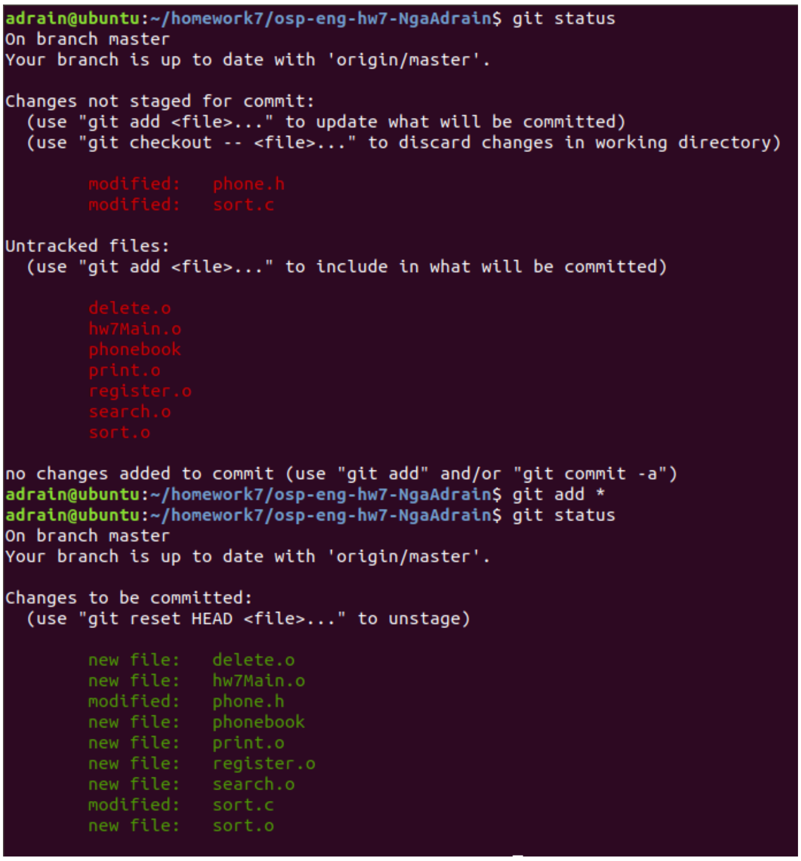


```
============ Telephone Book Management ============
 <<<1. Register  2. Print All    3. Search by ID        4. Delete       5. Sort         6. Exit
>>>
 Please enter your service number (1-6)> 5
Sort fuction is called
Before sorting
Addr vp:0x556c23cc30c0  name:Roki        phone:12345678
Addr vp:0x556c23cc30d7  name:Thor        phone:123456
Addr vp:0x556c23cc30ee  name:IronMan     phone:3765898
Addr vp:0x556c23cc3105  name:Hulk        phone:74561253
Addr vp:0x556c23cc311c  name:Groot       phone:99999999

After sorting
Addr vp:0x556c23cc30c0  name:Groot       phone:99999999
Addr vp:0x556c23cc30d7  name:Hulk        phone:74561253
Addr vp:0x556c23cc30ee  name:IronMan     phone:3765898
Addr vp:0x556c23cc3105  name:Roki        phone:12345678
Addr vp:0x556c23cc311c  name:Thor        phone:123456
```

- ➔ I can sort with name’s character value lowercase.
- ➔ Address of array’s elements are not changed that mean each value of array’s element swap deep in array not swallow(array’s elements are changed each others)

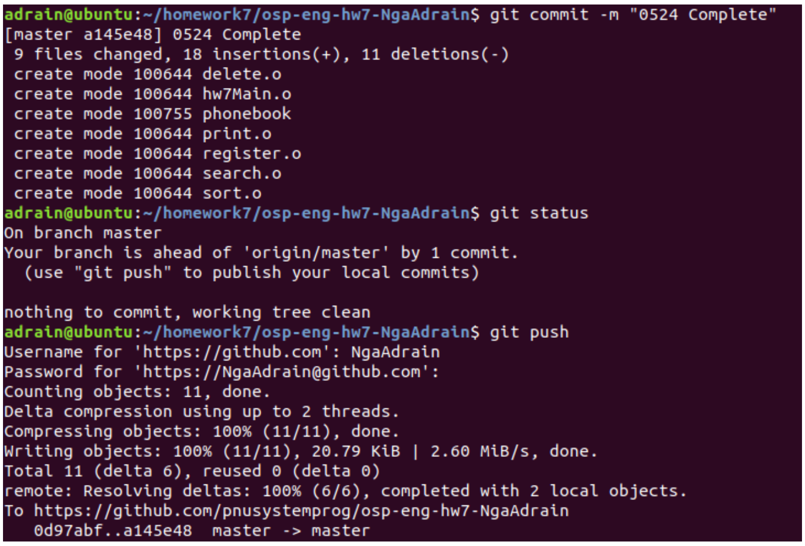
3. Github repository (20 points)

(1) include github commands for cloning, adding, committing and pushing



```
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   phone.h
        modified:   sort.c

Untracked files:
  (use "git add <file>..." to include in what will be committed)

        delete.o
        hw7Main.o
        phonebook
        print.o
        register.o
        search.o
        sort.o

no changes added to commit (use "git add" and/or "git commit -a")
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git add *
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git status
On branch master
Your branch is up to date with 'origin/master'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        new file:   delete.o
        new file:   hw7Main.o
        modified:   phone.h
        new file:   phonebook
        new file:   print.o
        new file:   register.o
        new file:   search.o
        modified:   sort.c
        new file:   sort.o
```

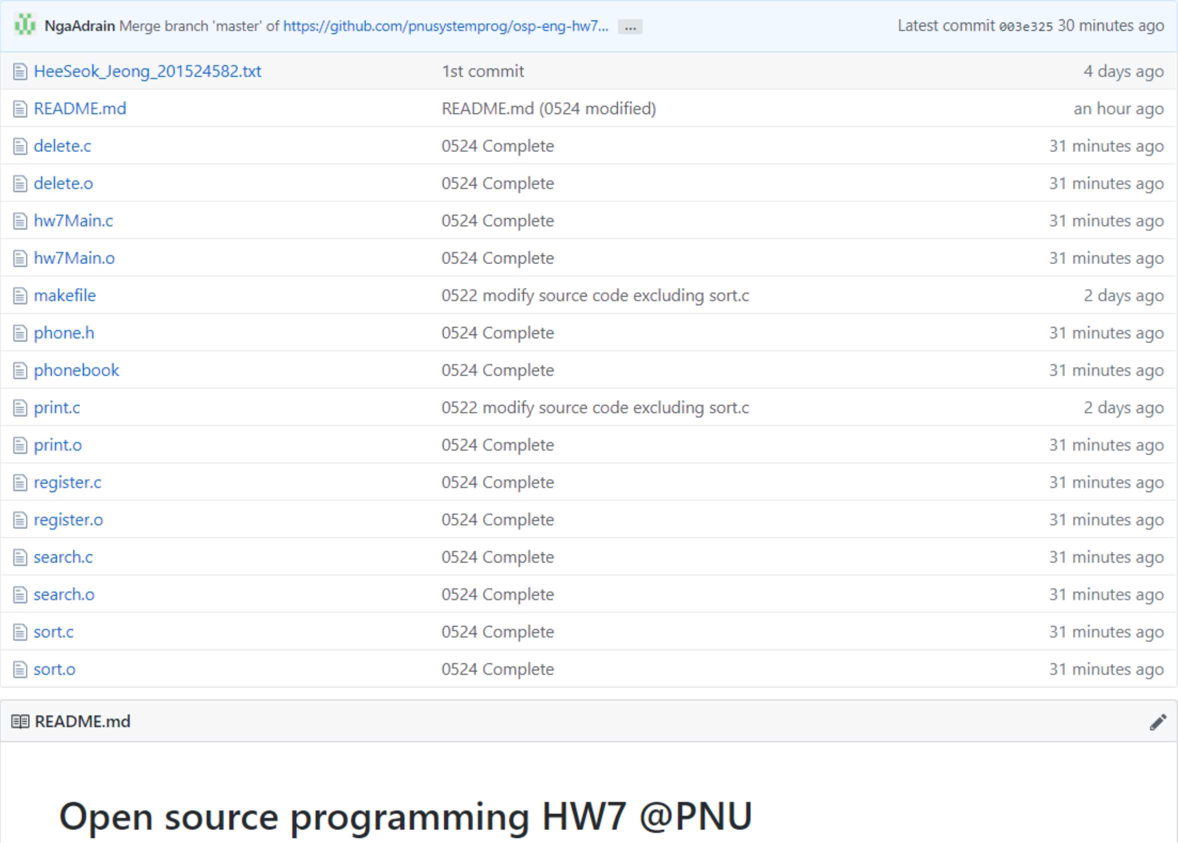
- ➔ Check status with 'git status' command and git add *(with wild card) append all changes to track



```
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git commit -m "0524 Complete"
[master a145e48] 0524 Complete
 9 files changed, 18 insertions(+), 11 deletions(-)
 create mode 100644 delete.o
 create mode 100644 hw7Main.o
 create mode 100755 phonebook
 create mode 100644 print.o
 create mode 100644 register.o
 create mode 100644 search.o
 create mode 100644 sort.o
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
adrain@ubuntu:~/homework7/osp-eng-hw7-NgaAdrain$ git push
Username for 'https://github.com': NgaAdrain
Password for 'https://NgaAdrain@github.com':
Counting objects: 11, done.
Delta compression using up to 2 threads.
Compressing objects: 100% (11/11), done.
Writing objects: 100% (11/11), 20.79 KiB | 2.60 MiB/s, done.
Total 11 (delta 6), reused 0 (delta 0)
remote: Resolving deltas: 100% (6/6), completed with 2 local objects.
To https://github.com/pnusystemprog/osp-eng-hw7-NgaAdrain
   0d97abf..a145e48  master -> master
```

- ➔ Commit file from track to local repository with message "0524 complete"
- ➔ Check status and push to remote repository.

(2) After pushing the source code and the makefile to your github directory, include the screen capture image of the Github repository



NgaAdrain Merge branch 'master' of https://github.com/pnusystemprog/osp-eng-hw7... … Latest commit 003e325 30 minutes ago

HeeSeok_Jeong_201524582.txt	1st commit	4 days ago
README.md	README.md (0524 modified)	an hour ago
delete.c	0524 Complete	31 minutes ago
delete.o	0524 Complete	31 minutes ago
hw7Main.c	0524 Complete	31 minutes ago
hw7Main.o	0524 Complete	31 minutes ago
makefile	0522 modify source code excluding sort.c	2 days ago
phone.h	0524 Complete	31 minutes ago
phonebook	0524 Complete	31 minutes ago
print.c	0522 modify source code excluding sort.c	2 days ago
print.o	0524 Complete	31 minutes ago
register.c	0524 Complete	31 minutes ago
register.o	0524 Complete	31 minutes ago
search.c	0524 Complete	31 minutes ago
search.o	0524 Complete	31 minutes ago
sort.c	0524 Complete	31 minutes ago
sort.o	0524 Complete	31 minutes ago

README.md

Open source programming HW7 @PNU

➔ This is my repository, I can commit and push successfully.

4. Discussion (10 points)

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

A. I can learn about function pointer, and void pointer. And when I was modify README.md file, after I try to push my local repository, error occurred. That case try once pull request, retry push. It can solve problem. And function pointer can replace switch-case condition, I think this is good for machine, but if we use this way, less readability. And I think void pointer is similar to c++'s template class. Because we use void pointer for express unknown type's address, In c++'s class use template class for express unknown class. Just my opinion. and I can review memory copy, string compare function.

- Describe difficulties during homework

A. It was easy homework because I can reuse previous homework's code, and sorting algorithm already determined with bubble sort. I wonder why select bubble sort? Not selection sort. Both of two algorithm are inefficient but we usually use selection sort. One difficulty thing is confusing of pointer's type casting on string compare. But lecture note was very helpful.

Anyway, I think this homework is good for exercising void pointer, function pointer.

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