HW7. 자기 참조 구조체와 공유 라이브러리

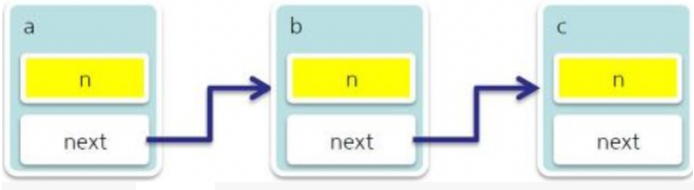
학과: 정보컴퓨터공학과
학번: 060분반 2020-55565
이름: 여지수
Github ID: duwltn1301@naver.com
제출일: 2020-06-13

1. 구현 내용에 대한 설명 (50점)

(1) 주요 변수 설명
(2) 주요 자료 구조 설명
- 자기 참조 구조체를 이용한 연결 리스트
(3) 주요 함수 구현 방법 설명

주요 변수는 extern 변수와 static 변수라고 말할 수 있다. 등록된 총 개인정보 수를 뜻하는 size 변수는 main 함수가 들어있는 소스코드에서 0으로 선언하고, 다른 소스코드에서는 extern으로 선언하여 다른 소스코드에서도 공유할 수 있도록 한다. register.c 소스코드에 있는 함수가 호출되면 size는 +1이 되도록, delete.c 소스코드에 있는 함수가 호출되면 size는 -1이 되도록 구현했다. 그리고 sort.c 소스코드에 있는 함수가 이 size 수로 버블정렬을 하도록 구현했다. size는 소스코드에서 공유되어 사용되기에 이처럼 extern으로 선언하게 된 것이다.
서비스가 몇번 호출되었는지 카운팅하는 count_service 변수는 static으로 선언하여 메인함수가 들어있는 소스파일 외의 다른 소스파일에서는 이 변수에 접근하지 못하도록 한다.

주요 자료 구조를 설명하자면 자기 참조 구조체 BOOK이 phone.h에 정의되어 있다. **구조체 멤버로 자기 자신의 구조체를 가르키는 포인터를 포함하는 구조체를 자기참조구조체** 라고 한다. 자기 참조 구조체BOOK에 이름 변수 Name[10], 전화번호 변수 PhoneNumber[13], 자기 자신의 구조체를 가르키는 포인터 변수(링크)를 변수 link로 정의를 했다. 각 구조체는 이 링크를 통하여 자기 자신의 구조체를 참조할 수 있다.
만약, struct BOOK *head;와 같이 구조체를 정의했을 때



head → n; head → next → n; head → next → next → n;

과 같은 방식으로 위 그림의 a,b,c 노드에 접근할 수 있다. 각 노드는 다음 노드의 주소를 가리키고, 마지막 노드는 마지막 노드는 NULL 값을 저장한다. 이 NULL 값은 더이상 연결된

노드가 없음을 의미한다.

주요 함수 구현을 설명하자면, 메인 함수가 들어있는 hw7Main.c 에서는 phone.h 헤더파일을 포함하고 메인에서 쓰일 함수들(registerPhoneData, print, serachByNAme, deleteByName, sort)을 선언하며, 정적 변수 count_service 와 공유될 extern 변수 size 를 0 으로 선언한다.
함수 포인터는 함수의 반환값 자료형을 지정해주고, 함수 포인터 이름 앞에 * (애스터리스크)를 붙인 뒤 ( ) (괄호)로 묶어줍니다. 그리고 다시 괄호를 붙여 함수라는 것을 알려주는데, 이러한 함수 포인터를 main에서 사용할 것이다. main함수 이전에 함수 포인터를 아래와 같이 선언해준다.
void (*pFuncs[5])() = {registerPhoneData, print, serachByNAme, deleteByName, sort};
pFuncs뒤의 []안의 값에 따라 함수가 호출될 것이다. 예를 들어 [1]이면 pFuncs[1]==print이므로 pFuncs[1]을 불러오면 print 함수를 호출하는 것이다. 따라서 함수포인터를 이용하면 여러 개의 함수를 배열로 관리할 수가 있다.
그래서 이후 main함수에서 1,2,3,4,5,6 서비스 중 -1만큼에 해당되는 리스트 요소를 찾아 그 함수를 호출하도록 구현했다. 6(exit)를 입력하지 않는 이상 서비스는 while문을 통해 계속된다. 1을 입력하면 resgisterPhoneData() 함수가, 2를 입력하면 print() 함수가, 3을 입력하면 searchByName()함수가, 4를 입력하면 deleteByName() 함수가, 5를 입력하면 sort()함수가 호출된다. 마지막으로 서비스가 종료될 때 서비스를 몇 번 이용했는지 service count를 출력한다.
register.c 에는 extern int size를 선언하고 registerPhoneData()함수를 구현했다. 우선 등록할 수 있는 전화번호는 50개까지 이므로 size가 50이상이면 더이상 추가 할 수 없음을 출력한다. 만약 size가 50보다 작다면, 패스워드를 입력하여 등록하게 된다.
코드를 보면 while 문이 두개가 나오는 데 첫번째 while문은 전화번호부의 가장 마지막에 이름과 번호를 등록하기 위해, 노드가 null이 되기 직전까지 가리키는 곳을 가장 뒤쪽으로 옮기는 작업이라 보면 된다. 두번째 while문에서는 이제 패스워드를 입력받고 맞추면 등록할 수 있는 코드를 구현했다. 이때 만일 패스워드가 맞다면 구조체 포인터 변수가 malloc을 통해 동적 할당이 되고 strcpy를 통해 구조체 포인터 변수의 변수에 접근하여 입력 받은 변수를 복사 시켜 놓으면 된다. 그리고, 맞춰야 할 password를 qqq로 선언하고 strcmp를 이용해 입력 받은 문자열과 맞춰야 할 password가 같은지 비교하고 같으면 등록을 받도록 하고 다르면 에러 메세지를 띄운다. 3번을 초과하여 틀리면 더이상 패스워드를 입력받지 못하도록 한다. 맞췄다면, size번째의 PhoneBook 구조체에 이름과 전화번호를 저장하고 size는 1 늘린다.
print.c에는 구조체 포인터 변수 list를 선언하고 list가 null이 되기 전까지 list의 변수를 접근해 출력하고, 다음 연결된 리스트로 넘어가 그 다음 변수를 출력해야되기 때문에 list가 다시 자기 자신을 참조하도록 구현한다.
sort.c에는 print() 함수를 선언하고 bubblesort 함수를 구현해 두었다. 버블 정렬로 stcmp를 이용해 첫번째 구조체 포인터 변수의 Name 변수와 그 첫번째 구조체가 가리키고 있는 두번째 구조체 포인터 변수의 Name 변수를 비교하여 strcpy를 이용하여 값을 바꾸어 저장해 준다.
phone.h 헤더파일에는 이름과 전화번호를 저장할 구조체를 선언하고, #ifndef와 #endif 지시자를 통해서 중복 선언을 막았다.
search.c에는 searchByName()함수를 구현했다. 찾을 이름을 입력받고, 구조체 포인터 변수가 null

이 되기 전까지 입력받은 변수와 구조체의 이름 변수를 strcmp로 비교하고 같으면 0으로 선언했었던 find 변수를 1로 바꾸고 출력한다. 이때 구조체 포인터 변수는 다음 구조체 포인터 변수를 가르킬 수 있도록 자기 자신을 참조한다. find가 1이 되었다면 사용자 입력한 이름이 전화번호부에 있었다는 것이고, find가 변하지 않았다면 사용자가 전화번호부에 없었다는 것이므로, 찾았을 때 찾지 못했을 때를 각각 알맞게 출력한다. 찾았을 때는 이름과 저장한 phone 변수를 출력하고 찾지 못했을 때는 없다고 출력한다.

delete.c에는 deleteByName()함수를 구현했다. searchByName()함수와 마찬가지로 구현하는데, find 변수를 이용해 찾지 못하면 없다는 메세지를 출력하고, 만약 찾았다면, free를 이용하여 동적 할당되었던 메모리를 해제 시켜준다. .

2. 실행 방법 설명 (20점)

(1) 사용한 운영체제 및 컴파일러의 종류

(2) Make 파일 설명 (10점)

(3) 실행 방법

(3) 숙제에서 제시한 동작들이 제대로 동작한다는 것을 확인할 수 있도록, 프로그램을 실행하고 그 결과를 캡쳐하며 보고서에 포함

(1)사용한 운영체제 및 컴파일러의 종류

-> 윈도우 운영체제에서 Dev-C++컴파일러를 이용하여 프로그램을 구현했다.

(2) makefile 설명

```
CC = gcc
CFLAGS = -Wall -g -c
INCLUDE= -I.
LIBS =-L. -lm

OBJS = hw7Main.o sort.o register.o delete.o print.o search.o
all : main

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

makefile에 대해 설명하자면,

CC=gcc

-> 매크로 CC를 정의한다. 컴파일러를 세팅해서 환경변수로 지정해주는 것이다. 버전을 바꿔서 컴파일할 때 유용하게 쓸 수 있다.

CFLAGS = -Wall -g -c

-> 매크로 CFLAGS를 정의한다. gcc의 옵션을 추가해주는 용도이다. -g는 디버그 정보를 추가하라

는 것이다.

INCLUDE = -I.

-> 디렉토리를 추가한다.

LIBS = -L. -lm

-> 링크할 때 필요한 라이브러리를 추가한다.

OBJS = hw7Main.o sort.o register.o delete.o print.o search.o

-> 오브젝트 파일들을 정의해준다.

all : main

-> make 는 makefile 을 순차적으로 읽어서 가장 처음에 나오는 규칙을 수행하게 된다. all 은 더미타겟(dummy target)이 바로 첫 번째 타겟으로써 작용하게 된다. 결과 파일이 많을 때 all 의 의존 관계(dependency)로써 정의해 두면 편리하다.

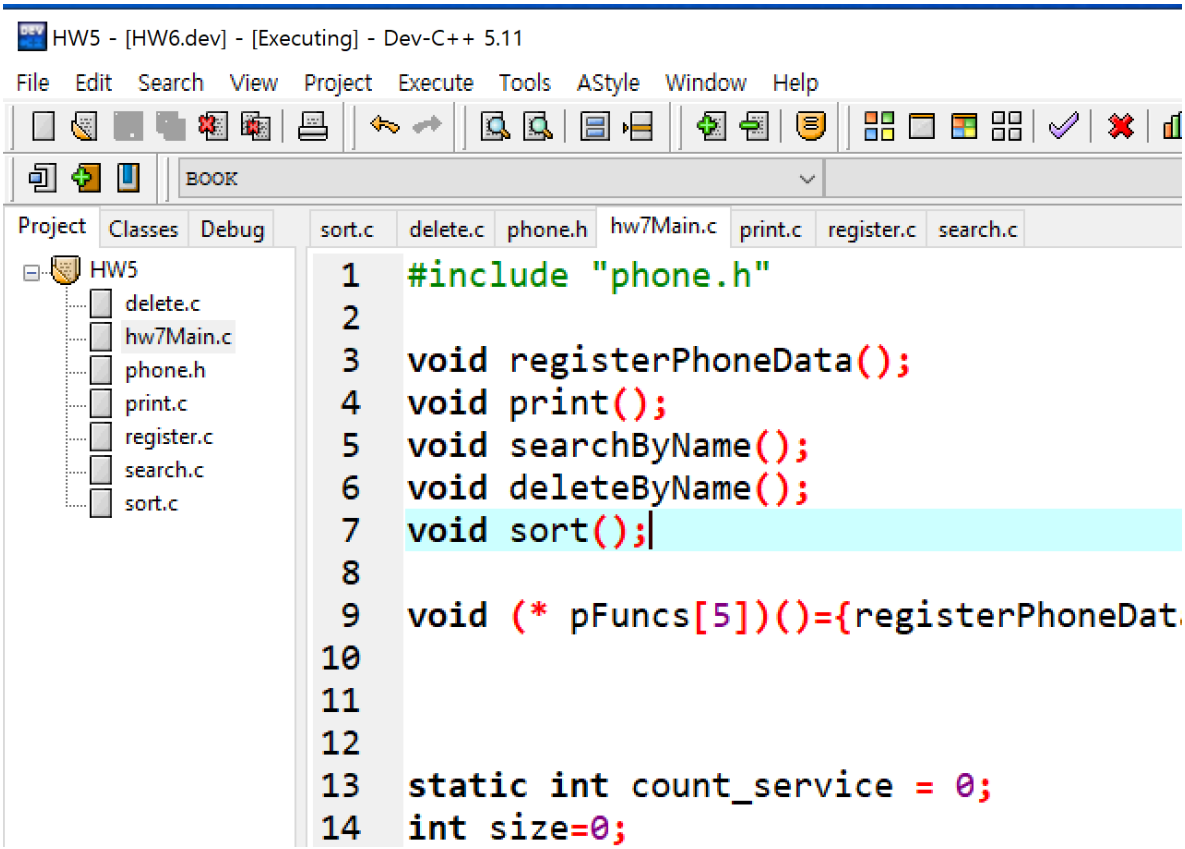
```
%.o: %.c
	$(CC) $(INCLUDE) $(CFLAGS) $<
```

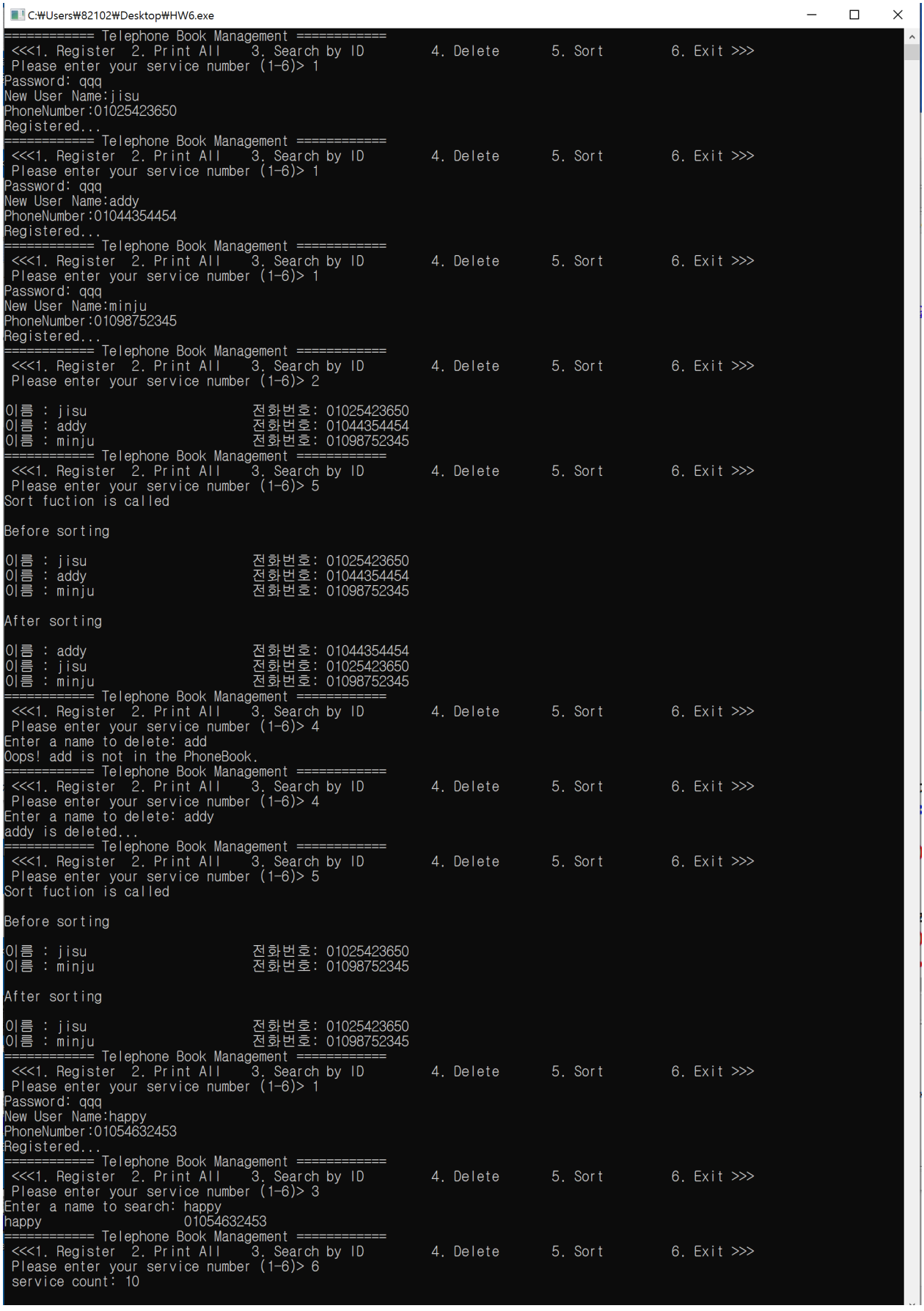
-> 내부적으로 확장자 규칙이 적용되었다. $(...)은 매크로의 사용을 의미한다. 매크로를 통해 복잡한 것을 간단하게 나타낼 수 있다. $<는 입력파일로 콜론의 오른쪽에 오는 패턴을 치 환하다는 의미를 담고있다. %는 일치하는 확장자를 제외한 파일명을 의미한다. 그래 서 %.c를 컴파일하여 같은 이름의 오브젝트 파일을 만들겠다는 것이다. 즉 어떤 .o 파일 이 필요할 경우 이 규칙에 의해서 해당 .c 파일을 찾고 아랫줄의 명령을 실행시켜서 .o 파일을 생성해내는 것이다.

```
main: $(OBJS)
	$(CC) -o main $(OBJS) $(LIBS)
clean:
	rm -f main $(OBJS)
```

(3) 컴파일 방법 및 실행 방법 -> Dev-c++ 컴파일로 작성한 코드를 Compile & Run 한다. 컴파일은 전처리-> 컴파일-> 어셈블-> 링크 이 네 가지 단계로 진행된다. 우선, 전처리 단계에서는 전처리기가 소스코드에서 #으로 시작하는 지시자를 처리한다. 전처리기는 #include를 만나면 뒤의 헤더파일을 그 파일의 내용을 순차적으로 삽입 한다. .c 파일로부터 .i 파일이 생성된다. 컴파일 단계에서는 컴파일러가 전처리가 끝난 파일을 컴파일하여 운영체제가 인식할 수 있는 형태의 개체 파일을 만든다. .i 파일로부터 .s 파일이 생성된다. 어셈블 단계에서는 어셈블러가 완전한 기계어로 바꾸어준다. .s 파일로부터 .o 파일이 생성된다. 링크 단계에서는 오브젝트 파일에 표준 C 라이브러리, 사용자 라이브러리가 결합되어 main 함수를 호출하여 프로그램의 코드가 실행되도록 한다. .o 파일로부터 .exe 파일이 생성된다.

(3) 동작을 확인할 수 있는 실행 화면 캡처


HW5 - [HW6.dev] - [Executing] - Dev-C++ 5.11
File Edit Search View Project Execute Tools AStyle Window Help
BOOK
Project Classes Debug
HW5
delete.c
hw7Main.c
phone.h
print.c
register.c
search.c
sort.c
sort.c delete.c phone.h hw7Main.c print.c register.c search.c
1 #include "phone.h"
2
3 void registerPhoneData();
4 void print();
5 void searchByName();
6 void deleteByName();
7 void sort();
8
9 void (* pFuncs[5])()={registerPhoneDat
10
11
12
13 static int count_service = 0;
14 int size=0;


C:₩Users₩82102₩Desktop₩HW6.exe
========== 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
Password: qqq
New User Name:jisu
PhoneNumber:01025423650
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
Password: qqq
New User Name:addy
PhoneNumber:01044354454
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
Password: qqq
New User Name:minju
PhoneNumber:01098752345
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)> 2
이름 : jisu 전화번호: 01025423650
이름 : addy 전화번호: 01044354454
이름 : minju 전화번호: 01098752345
========== 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
이름 : jisu 전화번호: 01025423650
이름 : addy 전화번호: 01044354454
이름 : minju 전화번호: 01098752345
After sorting
이름 : addy 전화번호: 01044354454
이름 : jisu 전화번호: 01025423650
이름 : minju 전화번호: 01098752345
========== 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
Enter a name to delete: add
Oops! add is not in the PhoneBook.
========== 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
Enter a name to delete: addy
addy 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)> 5
Sort fuction is called
Before sorting
이름 : jisu 전화번호: 01025423650
이름 : minju 전화번호: 01098752345
After sorting
이름 : jisu 전화번호: 01025423650
이름 : minju 전화번호: 01098752345
========== 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
Password: qqq
New User Name:happy
PhoneNumber:01054632453
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)> 3
Enter a name to search: happy
happy 01054632453
========== 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)> 6
service count: 10

3. Github 화면 (20점)

(1) cloning, adding, committing, push를 위한 github 명령들을 포함

(2) 소스 코드와 makefile을 push한 후, 본인의 Github repository를 스크린 캡쳐하여 포함

```
202055565@CSEDell:~/test2$ git clone https://github.com/datalab-pnu/unix-hw7-YeoJiSu
Cloning into 'unix-hw7-YeoJiSu'...
Username for 'https://github.com': duwltn1301@naver.com
Password for 'https://duwltn1301@naver.com@github.com':
remote: Enumerating objects: 11, done.
remote: Counting objects: 100% (11/11), done.
remote: Compressing objects: 100% (11/11), done.
remote: Total 11 (delta 0), reused 10 (delta 0), pack-reused 0
Unpacking objects: 100% (11/11), done.
202055565@CSEDell:~/test2$ touch jisu_yeo_202055565.txt
202055565@CSEDell:~/test2$ git add jisu_yeo_202055565.txt
fatal: not a git repository (or any parent up to mount point /)
Stopping at filesystem boundary (GIT_DISCOVERY_ACROSS_FILESYSTEM not set).
202055565@CSEDell:~/test2$ cd unix-hw7-YeoJiSu
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ touch jisu_yeo_202055565.txt
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git add jisu_yeo_202055565.txt
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ ls -al
total 80
drwxr-xr-x 3 202055565 sys059 4096 Jun 13 22:35 .
drwxr-xr-x 3 202055565 sys059 4096 Jun 13 22:35 ..
drwxr-xr-x 8 202055565 sys059 4096 Jun 13 22:34 .git
-rw-r--r-- 1 202055565 sys059  514 Jun 13 22:30 Makefile
-rw-r--r-- 1 202055565 sys059 1522 Jun 13 22:30 README.md
-rw-r--r-- 1 202055565 sys059  553 Jun 13 22:12 delete.c
-rw-r--r-- 1 202055565 sys059  142 Jun 13 22:30 div.c
-rw-r--r-- 1 202055565 sys059   99 Jun 13 22:30 div.h
-rw-r--r-- 1 202055565 sys059  819 Jun 13 22:09 hw7Main.c
-rw-r--r-- 1 202055565 sys059    0 Jun 13 22:34 jisu_yeo_202055565.txt
-rw-r--r-- 1 202055565 sys059  331 Jun 13 22:30 main.c
-rw-r--r-- 1 202055565 sys059  244 Jun 13 22:28 makefile
-rw-r--r-- 1 202055565 sys059  133 Jun 13 22:30 mult.c
-rw-r--r-- 1 202055565 sys059   98 Jun 13 22:30 multi..h
-rw-r--r-- 1 202055565 sys059  247 Jun 13 22:08 phone.h
-rw-r--r-- 1 202055565 sys059  224 Jun 13 22:12 print.c
-rw-r--r-- 1 202055565 sys059 1381 Jun 13 22:12 register.c
-rw-r--r-- 1 202055565 sys059  436 Jun 13 22:12 search.c
-rw-r--r-- 1 202055565 sys059  902 Jun 13 22:07 sort.c
-rw-r--r-- 1 202055565 sys059   88 Jun 13 22:30 sum.c
-rw-r--r-- 1 202055565 sys059   96 Jun 13 22:30 sum.h
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

        new file:   jisu_yeo_202055565.txt

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

        delete.c
        hw7Main.c
        makefile
        phone.h
        print.c
        register.c
        search.c
        sort.c

202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git add *.c
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git add *.h
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git add makefile
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git status
On branch main
Your branch is up to date with 'origin/main'.

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

        new file:   delete.c
        new file:   hw7Main.c
        new file:   jisu_yeo_202055565.txt
        new file:   makefile
        new file:   phone.h
        new file:   print.c
        new file:   register.c
        new file:   search.c
        new file:   sort.c
```

```
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git commit -m "Add a txt file and source codes"

*** Please tell me who you are.

Run

  git config --global user.email "you@example.com"
  git config --global user.name "Your Name"

to set your account's default identity.
Omit --global to set the identity only in this repository.

fatal: unable to auto-detect email address (got '202055565@CSEDell.(none)')
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git config --global user.email "duwltn1301@naver.com"
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git config --global user.name "YeoJuSu"
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git commit -m "Add a txt file and source codes"
[main ddee1bc] Add a txt file and source codes
 9 files changed, 254 insertions(+)
 create mode 100644 delete.c
 create mode 100644 hw7Main.c
 create mode 100644 jisu_yeo_202055565.txt
 create mode 100644 makefile
 create mode 100644 phone.h
 create mode 100644 print.c
 create mode 100644 register.c
 create mode 100644 search.c
 create mode 100644 sort.c
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$ git push
Username for 'https://github.com': duwltn1301@naver.com
Password for 'https://duwltn1301@naver.com@github.com':
Counting objects: 11, done.
Delta compression using up to 64 threads.
Compressing objects: 100% (10/10), done.
Writing objects: 100% (11/11), 3.20 KiB | 1.60 MiB/s, done.
Total 11 (delta 0), reused 0 (delta 0)
To https://github.com/datalab-pnu/unix-hw7-YeoJiSu
   418ea5f..ddee1bc  main -> main
202055565@CSEDell:~/test2/unix-hw7-YeoJiSu$
```

4. 논의 사항 (10점)

- 숙제를 하면서 새로이 알게 된 내용(수업 시간 이외의 내용),
- 숙제를 하는 중에 어려웠던 점 등을 기술

sort()함수를 구현하는 데에 몹시 어려움을 겪었다. 새로운 포인터 구조체 temp를 선언하여,
temp=list;
list->link=list;
list=list->link;
처음에 이렇게 접근을 하였는데 바뀌어지지 않음을 발견했다. 구글링을 해봐도 가능한 코드 인것처럼 보이는데 왜 되지 않는 것인지 이해가 잘 되지 않았다. 결국에는 strcpy를 사용하여 구조체의 Name 변수와 PhoneNumber 변수에 직접 접근하여 각각을 모두 바꿔주었더니 switch가 되었다. strcpy로 구현되는 방법 밖에 찾아지지 않았는데 왜 자기 참조 포인터로만의 접근은 되지 않는지 그 이유에 대해 함께 논의 해보면 좋을 것 같다.