

CSED101 Programming & Problem Solving

Fall, 2018

Programming Assignment #5

무은재 새내기학부

20180038

박형규

POVIS ID : hyeongkyu

Honor Code : 나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.

1. 프로그램 개요

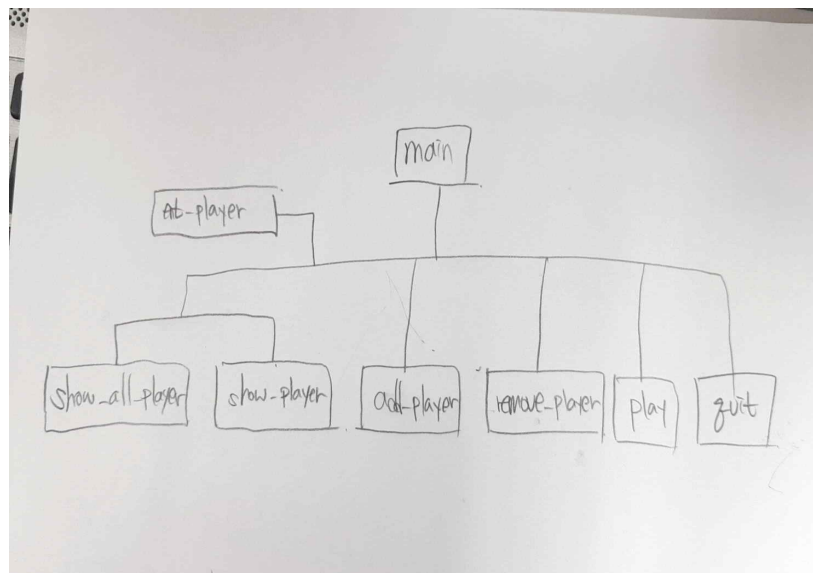
1.1 목적

이번 프로그램을 통하여 이전에 배웠던 파일 입출력을 이용하여 파일을 읽어 들이는 법을 복습하고 Structure와 linked list의 사용법을 익힌다. 또한 명령줄인수 (argc, argv)의 활용 방법을 익히며 다중 소스파일의 사용법을 익히는데 그 목적이 있다.

1.2 전체 구조

이번 과제는 우리에게 익숙한 수건 돌리기 게임을 술래가 잡히지 않는, 즉 술래가 수건을 놓는 자리에 있던 사람이 바로 술래가 되는 수건 돌리기 게임을 시뮬레이션 하게 된다. 5개의 그룹에 사람들이 존재하게 되며, 이 그룹은 linked list를 통하여 연결되어 있다.

2. 전체 구조도



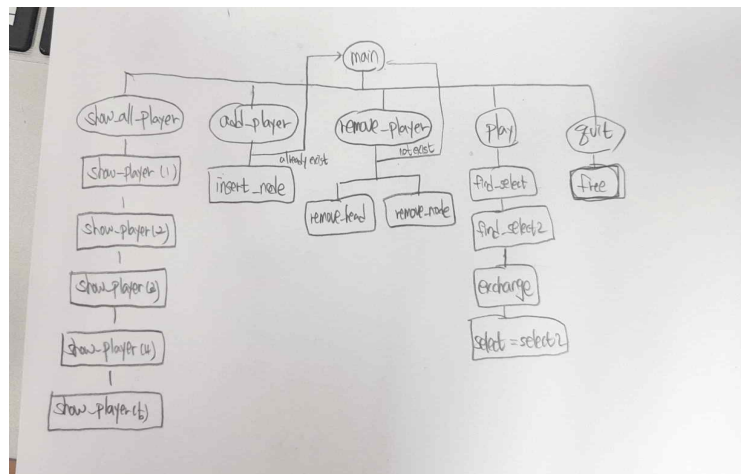
3. pseudocode

```

Algorithm Towel sharing
void ent_player(infile, plist);
void show_all_player(plist);
switch(num){
case add : add_player(plist);
case show : show_player(plist, gn);
case remove : remove_player(plist);
case play : play(infile, plist);
case quit : quit(plist);
}
return 0;

```

4. flow chart



5. 프로그램 코드

mystructure.h

```

#ifndef MYSTRUCTURE_H
#define MYSTRUCTURE_H

typedef struct {
char name[20];
char gender;
char dept[20];
int ID;
int group;
}STU

typedef struct node {
STU data;
struct node *next;
}NODE

typedef struct {
int count;
NODE* head;
}LIST
#endif

```

헤더파일을 .c에서 include 할 때 이번 프로젝트에서 만든 3개의 구조체가 player.h와 group.h 두 개에 분할되어 있다면 함수의 선언에서 include의 순서에 따라 오류가 발생하기 때문에 mystructure.h라는 헤더 파일 하나를 더 만들어서 구조체만을 모아두고 include 순서를 group.h와 player.h보다 위쪽에 해 두었다.

player.h

```

#ifndef PLAYER_H
#define PLAYER_H

void ent_player(FILE *infile, LIST **plist);
void show_all_player(LIST **plist);
void show_player(LIST **plist, int gn);
void quit(LIST **plist);
void add_player(LIST **plist);
void remove_player(LIST **plist);
#endif

```

플레이어에 관한 함수들을 선언하였다.

group.h

```
#ifndef GROUP_H
#define GROUP_H

void add_node(LIST **plist, NODE *temp);
void remove_node(LIST **plist, NODE *fi, int gn);
void play(FILE *infile, LIST **plist);
#endif
```

linked list를 편집하는 행위가 사용되는 함수들을 선언해 두었다.

main 함수 plist 동적할당, 플레이어 목록 입력, 모든 플레이어 목록 보이기 부분

```
LIST **plist;
plist = (LIST**)malloc(sizeof(LIST*) * 5);
for (int i = 0; i < 5; i++)
    plist[i] = (LIST*)malloc(sizeof(LIST));
plist[0]->count = 0, plist[0]->head = NULL;
plist[1]->count = 0, plist[1]->head = NULL;
plist[2]->count = 0, plist[2]->head = NULL;
plist[3]->count = 0, plist[3]->head = NULL;
plist[4]->count = 0, plist[4]->head = NULL;
FILE *infile;
if (argc == 1)
{
    infile = fopen("players.txt", "r");
    ent_player(infile, plist);
}
else
{
    if (infile = fopen(argv[1], "r") == NULL)
    {
        printf("Cannot open file\n");
        return 1;
    }
    else
    {
        infile = fopen(argv[1], "r");
        ent_player(infile, plist);
    }
}
fclose(infile);
printf("NameGenderDeptStudentID\n");
show_all_player(plist);//
printf("\n");
char *command;
char real[20];
```

처음에 plist를 동적할당을 통하여 데이터 공간을 만들어 주고, argc와 argv를 사용하여 처음에 아무것도 입력하지 않았을 경우에는 players.txt를 자동으로 받아들이게 해 주고, 어떠한 것을 입력하였을 경우에는 그 파일을 받아드리게 한다. 또한 플레이어 목록을 모두 받아들인 후에는 show_all_player()함수를 통하여 모든 플레이어 목록을 print한다.

main함수 반복문 중 show, add, remove 부분

```
while (1)
{
printf(">> ");
scanf("%[^Wn]", real);
getchar();
command = strtok(real, " ");

if (strcmpi(command, "show") == 0)
{
int gn;
printf("Which group? (0: all) ");
scanf("%d", &gn);
getchar();
printf("NameGenderDeptStudentIDWn");
if (gn == 0)
show_all_player(plist);
else
show_player(plist, gn);
}

else if (strcmpi(command, "add") == 0)
{
add_player(plist);
}

else if (strcmpi(command, "remove") == 0)
{
remove_player(plist);
}
```

처음 입력받은 값에서 strtok()함수를 통하여 띄어쓰기 이전의 값만을 command에 저장하여 command에 따라 show, add, remove에 해당하는 함수를 불러와서 해당 기능을 수행한다. 또한 문자열을 비교할 때 strcmpi 함수를 통하여 대문자와 소문자에 관계없이 비교 되도록 하였다.

main함수 반복문 중 play, quit 부분

```
else if (strcmp(command, "play") == 0)
{
    char *play_what;
    play_what = strtok(NULL, " ");
    if (play_what == NULL)
    {
        infile = fopen("game.txt", "r");
        if (infile == NULL)
        {
            printf("Cannot open file!");
            continue
        }
    }
    else
    {
        infile = fopen(play_what, "r");
        if (infile == NULL)
        {
            printf("Cannot open file!");
            continue
        }
    }
    play(infile, plist);
}

else if (strcmp(command, "quit") == 0)
{
    quit(plist);
    break
}
printf("Wn");
}
```

처음 입력받은 값에서 strtok()함수를 통하여 띄어쓰기 이전의 값만을 command에 저장하여 command에 따라 play, quit에 해당하는 함수를 불러와서 해당 기능을 수행한다. 또한 play일 경우에 strtok()함수를 다시 한 번 호출하여 처음 인자를 NULL로 받아 이전에 strtok()함수를 통해 남아있던 인자를 다시 한 번 호출하여 그 뒤의 값을 play_what에 저장하고, play_what에 어떤 값이 입력되어 있다면 그 값에 해당하는 파일을 불러들이고, 아무것도 입력되어 있지 않다면 game.txt를 자동으로 불러들이게 한다.

ent_player

```
NODE *fi;
int n;
while (1)
{
    NODE *temp = (NODE*)malloc(sizeof(NODE));
    n = fscanf(infile, "%[^Wt]Wt", temp->data.name);
    fscanf(infile, "%cWt", &temp->data.gender);
    fscanf(infile, "%[^Wt]Wt", temp->data.dept);
    fscanf(infile, "%dWt", &temp->data.ID);
    fscanf(infile, "%dWn", &temp->data.group);
    if (n == EOF)
        break;
    switch (temp->data.group)
    {
        case 1:
        case 2:
        case 3:
        case 4:
        case 5:
            add_node(plist,temp);
            break;
        default:
            printf("ERROR GROUP NUMBERWn");
            return;
    }
}
```

fscanf()함수를 통하여 플레이어 정보를 받아들이고 switch-case문을 통해 linked list를 활용하여 플레이어를 각 해당되는 그룹에 집어넣는다.

show_all_player

```
show_player(plist, 1);
show_player(plist, 2);
show_player(plist, 3);
show_player(plist, 4);
show_player(plist, 5);
```

show_player

```

NODE *pr = plist[gn - 1]->head;
printf("=====
=====Wn", gn);
if (plist[gn - 1]->head == NULL)
return
else
{
do {
printf("%-20s%c%-20s%dWn", pr->data.name, pr->data.gender, pr->data.dept, pr->data.ID);
pr = pr->next;
} while (pr != plist[gn - 1]->head);
}

```

gn 에 해당되는 그룹을 보여주는 역할을 한다.

add_player 플레이어 정보 입력받는 부분

```
NODE *temp = (NODE*)malloc(sizeof(NODE));
NODE *fi;
printf("StudentID: ");
scanf("%d", &temp->data.ID);
getchar();

for (int gn = 0; gn < 5; gn++)
{
    fi = plist[gn]->head;
    do
    {
        if (fi == NULL)
        {
            break
        }
        else
        {
            if (temp->data.ID == fi->data.ID)
            {
                printf("Player already exists\n");
                printf("NameGenderDeptStudentIDGroup\n");
                printf("%-20s%c%-20s%d%d\n", fi->data.name, fi->data.gender, fi->data.dept, fi->data.ID,
                    fi->data.group);
                return
            }
            fi = fi->next;
        }
    } while (fi != plist[gn]->head);
}

printf("Name: ");
scanf("%[^Wn]", temp->data.name);
getchar();
printf("Gender: ");
scanf("%c", &temp->data.gender);
getchar();
printf("Dept: ");
scanf("%[^Wn]", temp->data.dept);
getchar();
printf("Group: ");
scanf("%d", &temp->data.group);
getchar();
add_node(plist, temp);
printf("Player successfully added to group %d\n", temp->data.group);
```

remove_player 그룹에서 첫 노드 삭제하는 부분

```
NODE *temp;
NODE *prev;
int rID;
printf("StudentID: ");
scanf("%d", &rID);
getchar();

for (int gn = 0; gn < 5; gn++)
{
    NODE *fi = plist[gn]->head;
    if (fi != NULL)
    {
        do
        {
            if (rID == fi->data.ID)
            {
                remove_node(plist, fi, gn);
                printf("Player successfully removed from group %d\n", gn + 1);
                return
            }
            fi = fi->next;
        } while (fi != plist[gn]->head);
    }
}

printf("Players does not exist!\n");
```

StudentID를 통하여 삭제할 학생을 입력받고 linked list를 통하여 학생이 해당되는 그룹에서 학생을 삭제한다. 만약 삭제할 학생이 없는 경우 Players does not exist!를 출력한다.

play 처음 술래 정하는 부분

```

NODE *select = NULL
NODE *select2 = NULL
NODE *where;
NODE *where2;
NODE *temp;
NODE *temp2;
NODE *fi;
int gn, ng, np, n, i, num;
char filename[30] = { 'W0' };

for (gn = 0; gn < 5; gn++)
{
    if (plist[gn]->head != NULL)
    {
        select = plist[gn]->head;
        break
    }
}
if (select == NULL)
{
    printf("And then there were none...Wn");
    return
}

printf("FromToWn");
```

for문을 통하여 처음 술래를 가장 먼저 존재하는 그룹의 head로 정한다.

play 두 번째 술래 찾는 부분

```
while (1)
{
n = fscanf(infile, "%d %dWn", &ng, &np);
if (n == EOF)
break;
if (plist[ng - 1]->head == NULL) //
continue;

if (plist[select->data.group - 1]->head == select)
{
for (prev = plist[select->data.group - 1]->head; prev->next != plist[select->data.group - 1]->head; prev = prev->next);
plist[select->data.group - 1]->head = select->next;
prev->next = plist[select->data.group - 1]->head;
}

if (plist[ng - 1]->head != NULL)
{
if (np > plist[ng - 1]->count && np % plist[ng - 1]->count != 0)
{
np = np % plist[ng - 1]->count;
}
else if (np % plist[ng - 1]->count == 0)
{
np = plist[ng - 1]->count;
}

fi = plist[ng - 1]->head;
i = 1;
do {
if (i == np)
{
select2 = fi;
break
}
i++;
fi = fi->next;
} while (fi != plist[ng - 1]->head);
}

printf("%-20sWt(Group    %d)Wt->Wt%-20sWt(Group    %d)Wn",    select->data.name,
select->data.group, select2->data.name, select2->data.group);
```

ng와 np를 입력받아서 linked list를 통하여 두 번째 술래를 찾는다.

play 술래와 두 번째 술래의 위치를 바꾸어 주는 부분

```
for (prev2 = plist[select2->data.group - 1]->head; prev2->next != plist[select2->data.group - 1]->head; prev2 = prev2->next);

if (select2 == plist[select2->data.group - 1]->head)
where2 = prev2;
else
{
temp2 = plist[select2->data.group - 1]->head;
do {
if (temp2->next == select2)
break
temp2 = temp2->next;
} while (temp2->next != plist[select2->data.group - 1]->head);

if (temp2->next == select2)
where2 = temp2;
else
{
printf("No such position in this linked list!\n");
return
}
}

if (select2 == plist[select2->data.group - 1]->head)
plist[select2->data.group - 1]->head = select;

where2->next = select;
select->next = select2->next;

select->data.group = select2->data.group;

select = select2;
}

printf("\n\nGame over!\n");
printf("%s (Group %d) is out.", select->data.name, select->data.group);

remove_node(plist, select, select->data.group - 1);
```

현재의 술래와 두 번째 술래의 위치를 바꾸어 주고, 마지막에 두 번째 술래를 다음 턴에 현재 술래로 정할 수 있도록 해주고 마지막에 지목된 술래를 linked list를 이

용하여 해당되는 그룹에서 삭제한다.

quit

```
FILE *outfile;
NODE *pr;
NODE *fr;
outfile = fopen("result_game.txt", "w");
for (int gn = 0; gn < 5; gn++)
{
    pr = plist[gn]->head;
    if (pr != NULL)
    {
        do {
            fprintf(outfile, "%-20s%c%-20s%d%dWn", pr->data.name, pr->data.gender, pr->data.dept,
            pr->data.ID, pr->data.group);
            pr = pr->next;
        } while (pr != plist[gn]->head);
    }
}

NODE *fr2;
for (int gn = 0; gn < 5; gn++)
{
    fr = plist[gn]->head;
    if (fr != NULL)
    {
        do {
            fr2 = fr->next;
            free(fr);
            fr = fr2;
        } while (fr != plist[gn]->head);
    }
}

for (int gn = 0; gn < 5; gn++)
    free(plist[gn]);
free(plist);

printf("File saved. Bye!");
return;
```

마지막에 남아있는 플레이어들의 목록을 fprintf() 함수를 통하여 출력하며 동적할당 해제를 통하여 남아있는 데이터를 해제하고 프로그램을 종료시킨다.

add_node

```

NODE *fi;
if (plist[temp->data.group - 1]->count == 0) //
{
temp->next = temp
plist[temp->data.group - 1]->head = temp
}
else //
{
for (fi = plist[temp->data.group - 1]->head; fi->next != plist[temp->data.group - 1]->head;
fi = fi->next);
fi->next = temp
temp->next = plist[temp->data.group - 1]->head;
}
(plist[temp->data.group - 1]->count)++ ;

```

링크드리스트에 노드를 삽입하는 함수이다.

remove_node

```

NODE *temp;
NODE *prev;
if (plist[gn]->head == fi) //
{
if (plist[gn]->count == 1) //
{
plist[gn]->head = NULL;
free(fi);
}
else //2
{
plist[gn]->head = fi->next;
for (prev = plist[gn]->head; prev->next != fi; prev = prev->next);
prev->next = fi->next;
free(fi);
}
}
else //or
{
temp = plist[gn]->head;
do {
if (temp->next == fi)
break;
temp = temp->next;
} while (temp->next != plist[gn]->head);

if (temp->next == fi)
{
temp->next = fi->next;
free(fi);
}
}
(plist[gn]->count)--;

```

링크드리스트에서 노드를 삭제할 때 사용하는 함수이다.

6. 프로그램 실행 방법

cmd를 통하여 처음에 assn5.exe를 실행시키고 그 옆에 파일명을 입력하면 해당되는 파일을 열고, 입력하지 않으면 자동으로 players.txt를 열어 플레이어 목록을 입력받는다. 처음에는 모든 플레이어의 목록을 입력받게 되며 후에 show, add, remove, play, quit을 입력하여 해당되는 기능을 사용할 수 있다.

7. 프로그램 실행 예제

```
C:\Users\박형규\source\repos\assn5\Debug>assn5.exe players.txt
Name Gender Dept StudentID
===== GROUP 1 =====
Lee Jimin F Math 20160175
Choi Moong M Life 20130051
Woo Jiho M Computer 20150248
Seo Juwon M Math 20140332
Min Jia F History 20180123
Ki Eunjeong F Life 20170153
Ji Youngho M History 20180099
===== GROUP 2 =====
Kim Hyejeong F History 20180112
Seong Harin F Art 20130351
Yoon Haeun F Computer 20160065
Yoo Jiwoo F Art 20150034
Lee Donghyun M Math 20180369
Han Youngcheol M Computer 20130421
Jeong Sanghun M Life 20150222
Park Woojin M Life 20180056
Jo Sooyeon F History 20170515
===== GROUP 3 =====
Park Veeun F Computer 20150132
Kang Wonsik M Art 20180144
Kim Taewoo M Computer 20170023
Won Jinah F Math 20160012
Jeong Taeyoung M History 20180077
===== GROUP 4 =====
Go Jaewon M Art 20160525
Choi Seojun M Life 20150379
Kang Minseo F History 20150432
Ha Chaewon F Math 20180087
Jin Dongwon M Computer 20170431
Lim Jihye F History 20160013
Um Soobin F Computer 20180521
Kim Soojin F Life 20180445
Lee Seoyun F Art 20140663
Park Jeonghun M Math 20160647
===== GROUP 5 =====
Jang Kihun M Life 20170598
Jang Soah F Art 20170193
Kim Jihun M Life 20180111
Gwon Hyunwoo M History 20160003
Son Yoojin F Computer 20150182
Kim Eunyoung F Computer 20160552
Hong Seongho M Computer 20180078
>> _
```

players.txt를 입력하였을 때 초기화면

```
C:\Users\박형규\source\repos\assn5\Debug>assn5.exe
Name Gender Dept StudentID
===== GROUP 1 =====
Lee Jimin F Math 20160175
Choi Moong M Life 20130051
Woo Jiho M Computer 20150248
Seo Juwon M Math 20140332
Min Jia F History 20180123
Ki Eunjeong F Life 20170153
Ji Youngho M History 20180099
===== GROUP 2 =====
Kim Hyejeong F History 20180112
Seong Harin F Art 20130351
Yoon Haeun F Computer 20160065
Yoo Jiwoo F Art 20150034
Lee Donghyun M Math 20180369
Han Youngcheol M Computer 20130421
Jeong Sanghun M Life 20150222
Park Woojin M Life 20180056
Jo Sooyeon F History 20170515
===== GROUP 3 =====
Park Veeun F Computer 20150132
Kang Wonsik M Art 20180144
Kim Taewoo M Computer 20170023
Won Jinah F Math 20160012
Jeong Taeyoung M History 20180077
===== GROUP 4 =====
Go Jaewon M Art 20160525
Choi Seojun M Life 20150379
Kang Minseo F History 20150432
Ha Chaewon F Math 20180087
Jin Dongwon M Computer 20170431
Lim Jihye F History 20160013
Um Soobin F Computer 20180521
Kim Soojin F Life 20180445
Lee Seoyun F Art 20140663
Park Jeonghun M Math 20160647
===== GROUP 5 =====
Jang Kihun M Life 20170598
Jang Soah F Art 20170193
Kim Jihun M Life 20180111
Gwon Hyunwoo M History 20160003
Son Yoojin F Computer 20150182
Kim Eunyoung F Computer 20160552
Hong Seongho M Computer 20180078
>>
```

assn5.exe만을 입력하였을 때 초기화면

```
>> show
Which group? (0: all) 1
Name      Gender      Dept      StudentID
===== GROUP 1 =====
Lee Jimin      F      Math      20160175
Choi Woong     M      Life      20130051
Moo Jiho       M      Computer  20150248
Seo Juwon      M      Math      20140332
Min Jia        F      History   20180123
Ki Eunjeong    F      Life      20170153
Ji Younggho    M      History   20180099
>>
```

show 예시

```
>> add
StudentID: 20190038
Name: Sim ChaeHyeon
Gender: F
Dept: Computer
Group: 1
Player successfully added to group 1

>> show
Which group? (0: all) 1
Name      Gender      Dept      StudentID
===== GROUP 1 =====
Lee Jimin      F      Math      20160175
Choi Woong     M      Life      20130051
Moo Jiho       M      Computer  20150248
Seo Juwon      M      Math      20140332
Min Jia        F      History   20180123
Ki Eunjeong    F      Life      20170153
Ji Younggho    M      History   20180099
Sim ChaeHyeon  F      Computer  20190038
>>
```

add 예시

```
>> show
Which group? (0: all) 1
Name      Gender      Dept      StudentID
===== GROUP 1 =====
Lee Jimin      F      Math      20160175
Choi Woong     M      Life      20130051
Moo Jiho       M      Computer  20150248
Seo Juwon      M      Math      20140332
Min Jia        F      History   20180123
Ki Eunjeong    F      Life      20170153
Ji Younggho    M      History   20180099
Sim ChaeHyeon  F      Computer  20190038

>> remove
StudentID: 20190038
Player successfully removed from group 1

>> show
Which group? (0: all) 1
Name      Gender      Dept      StudentID
===== GROUP 1 =====
Lee Jimin      F      Math      20160175
Choi Woong     M      Life      20130051
Moo Jiho       M      Computer  20150248
Seo Juwon      M      Math      20140332
Min Jia        F      History   20180123
Ki Eunjeong    F      Life      20170153
Ji Younggho    M      History   20180099
>>
```

remove 예시

```

>> play
From Lee Jimin (Group 1) -> To Kang Wonsik (Group 3)
Kang Wonsik (Group 3) -> Ki Eunjeong (Group 1)
Ki Eunjeong (Group 1) -> Han Youngcheol (Group 2)
Han Youngcheol (Group 2) -> Park Jeonghun (Group 4)
Park Jeonghun (Group 4) -> Jang Sooah (Group 5)
Jang Sooah (Group 5) -> Seo Juwon (Group 1)
Seo Juwon (Group 1) -> Kim Hyejeong (Group 2)
Kim Hyejeong (Group 2) -> Lee Jimin (Group 3)
Lee Jimin (Group 3) -> Gwon Hyunwoo (Group 5)
Gwon Hyunwoo (Group 5) -> Kim Soojin (Group 4)

Game over!
Kim Soojin (Group 4) is out.
>> show
Which group? (0: all) 4
===== GROUP 4 =====
Name Gender Dept StudentID
Go Jaewon M Art 20160525
Choi Seojun M Life 20150379
Kang Minseo F History 20150432
Ha Chaewon F Math 20180087
Jin Dongwon M Computer 20170431
Lim Jihye F History 20160013
Um Soobin F Computer 20180521
Gwon Hyunwoo M History 20160003
Lee Seoyun F Art 20140663
Han Youngcheol M Computer 20130421
>>

```

play 예시

```

>> quit
File saved. Bye!
C:\Users\박창규\source\repos\asn5\Debug>

```

quit 예시

```

C:\Users\박창규\source\repos\asn5\Debug>asn5.exe no12.txt
===== GROUP 1 =====
Name Gender Dept StudentID
===== GROUP 2 =====
===== GROUP 3 =====
Park Yeeun F Computer 20150132
Kang Wonsik M Art 20180144
Kim Taewoo M Computer 20170023
Mon Jinah F Math 20160012
Jeong Taeyoung M History 20180077
===== GROUP 4 =====
Go Jaewon M Art 20160525
Choi Seojun M Life 20150379
Kang Minseo F History 20150432
Ha Chaewon F Math 20180087
Jin Dongwon M Computer 20170431
Lim Jihye F History 20160013
Um Soobin F Computer 20180521
Kim Soojin F Life 20180445
Lee Seoyun F Art 20140663
Park Jeonghun M Math 20160647
===== GROUP 5 =====
Jang Kihun M Life 20170538
Jang Sooah F Art 20170193
Kim Jihun M Life 20180111
Gwon Hyunwoo M History 20160003
Son Voojin F Computer 20150182
Kim Eunyoung F Computer 20160552
Hong Seongho M Computer 20180078

>> play 3_1.txt
From Park Yeeun (Group 3) -> To Kang Wonsik (Group 3)

Game over!
Kang Wonsik (Group 3) is out.
>> show
Which group? (0: all) 3
===== GROUP 3 =====
Name Gender Dept StudentID
Park Yeeun F Computer 20150132
Kim Taewoo M Computer 20170023
Mon Jinah F Math 20160012
Jeong Taeyoung M History 20180077
>>

```

‘play 명령어 참고 실행 예시’에 대한 예시

8. 토론 및 결론

이번 assignment를 통하여 C언어에서 가장 어렵다고 불리는 linked list를 이해하게 되었다. 또한 앞에서 배운 파일 입출력 부분도 다시 한 번 다뤄보며 잊어버리지 않게 되었고, 반복문, 조건문, 사용자 함수 생성, 구조체 등 앞에서 배운 여러 가지 개념들도 적용시켜보며 지금까지 배운 개념들을 다잡는 계기가 된 것 같다. 배열과 linked list 사이의 관계를 생각해보며 각각의 장점과 단점을 생각하며 프로그램을 구현시킬 수 있었다.