

CSED101. Programming & Problem solving

Fall, 2018

Programming Assignment #5 (70 points)

장성보(jang.sb@postech.ac.kr)

■ **Due:** 2018.12.12 23:59

■ **Development Environment:** Windows Visual Studio 2017

■ **제출물**

- **C Code files (*.c, *.h)**
 - 제출시, 모든 파일을 하나의 파일로 압축해서 제출할 것. (압축파일명: assn5.zip)
 - 프로그램의 소스 코드를 이해하기 쉽도록 반드시 주석을 붙일 것.
- **보고서 파일** (.doc(x) or .hwp) 예) assn5.doc(x) 또는 assn5.hwp
 - AssnReadMe.pdf 를 참조하여 작성할 것.
 - 프로그램 실행화면 캡처하여 보고서에 포함시키고 간단히 설명 할 것.
 - 명예서약(Honor code): 표지에 다음의 내용을 포함한다. "나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다." 보고서 표지에 명예서약이 없는 경우는 과제를 제출하지 않은 것으로 처리한다.
- 소스코드와 보고서 파일을 LMS를 이용하여 제출한다.

■ **주의사항**

- 각 문제에 해당하는 요구사항을 반드시 지킬 것.
- 모든 문제의 출력 형식은 아래의 예시들과 동일해야 하며, 같지 않을 시는 감점이 된다.
- 이번 과제를 작성하는 데 있어서 전역변수를 사용할 수 없다.
- 각 문제에 제시되어 있는 파일이름으로 제출 할 것. 그 외의 다른 이름으로 제출하면 감점 또는 0점 처리된다.
- 컴파일 & 실행이 안되면 무조건 0점 처리된다.
- 하루 late시 20%가 감점되며, 3일 이상 지나면 받지 않는다. (0점 처리)
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 'POSTECH 전자컴퓨터공학부 부정행위 정의'를 따른다. (LMS의 과목 공지사항의 제목 [document about cheating]의 첨부파일인 disciplinary.pdf를 참조할 것.)
- 이번 과제에서는 추가 기능 구현에 대한 추가 점수는 없습니다.

■ 수건 돌리기 시뮬레이션

(소개)

수건 돌리기는 여러 명이 원으로 둘러앉아 즐길 수 있는 놀이이다. 기본적인 게임 방식은 다음과 같다.

- 술래가 수건을 들고 원 주위를 돌아 한 사람(A)을 선택해 그 뒤에 수건을 내려놓는다.
- 술래가 한 바퀴를 돌아 다시 그 자리로 돌아오면 A가 술래가 된다.
- 그 전에 A가 술래를 잡으면 술래는 탈락한다.

이번 과제에서는 술래가 잡히지 않는, 즉 술래가 수건을 놓는 자리에 있던 사람이 바로 술래가 되는 수건 돌리기 게임을 시뮬레이션하는 프로그램을 작성하는 것을 목표로 한다.

하나의 원에서 진행되는 기존의 수건 돌리기와는 다르게, 이번 과제에서는 5개의 조가 각각의 원을 이룬 뒤 함께 진행하도록 변형된 규칙을 적용한다. 변경된 규칙은 다음과 같다.

- 술래는 자신이 속한 조를 포함한 5개의 조에서 한 사람(A)을 선택해 수건을 내려놓는다.
- 선택된 순간 술래는 A의 자리를 차지하게 되고, A가 술래가 되어 본 과정을 반복한다.

(목적)

- Structure와 linked list의 사용법을 익힌다.
- 명령줄인수(argc, argv)의 활용 방법을 익힌다.
- 다중 소스파일의 사용법을 익힌다.

(주의사항)

1. 본 프로그램은 5개의 명령어(show, add, remove, play, quit)를 입력 받아 기능을 수행한다. 각 명령어 별로 함수를 정의하여 사용한다. 명령어 외에 필요한 함수는 정의하여 사용할 수 있다.
2. 코드는 여러 개의 파일로 분할하여 작성한다.
 - **플레이어 정보 (player.h, player.c):** 이름, 성별, 학과, 학번 등의 정보를 가진 플레이어 정보를 구조체로 선언하고, 플레이어 정보와 관련된 함수들(플레이어 정보 입력 등)이 선언 및 정의되어야 한다.
 - **그룹 리스트 (group.h, group.c):** 게임에 참여하는 각 그룹(조)은 연결리스트 구조로 선언하고, 연결리스트의 삽입, 삭제 등의 함수들이 선언 및 정의되어야 한다.
 - **assn5.c**
 - main() 함수를 포함하여 필요한 함수들 정의
 - main() 함수 내에서는 사용자로부터 명령어를 입력 받아 처리한다.
3. 이번 과제는 구조체와 연결리스트를 활용하는 것이 목표이므로, 문제에 구조체와 연결리스트를 언급한 부분을 배열을 통해서 해결할 경우 감점 처리한다.
4. 명시한 에러 처리 외에는 고려하지 않아도 된다.

(설명)

- 각 그룹의 구현은 구조체와 연결리스트를 사용하여야 하며, 각 항목에 대한 제약은 다음과 같다.

- **이름(name):** 영어 최대 20글자(성과 이름 사이에 공백 1개 있음)
- **성별(gender):** 'M'은 남성, 'F'는 여성(영문 1글자)
- **학과(dept):** 영어 최대 20글자, 공백 없다고 가정
- **학번(studentID):** 8자리 정수로, 모든 학생들이 제각기 다른 학번을 가지고 있으므로 식별자(identifier)로 사용한다.
- **조(group):** 조 번호를 나타내는 정수, 이번 과제에서는 총 5개의 조가 있다고 가정한다.

- 프로그램 실행은 다음과 같이 한다. (노란색으로 표시된 문자는 사용자 입력에 해당)

```
C:\W>assn5.exe players.txt
```

- 파일명을 입력하지 않은 경우, default로 "players.txt"로부터 데이터를 읽는다.
- 파일명이 입력된 경우, 입력된 파일로부터 데이터를 읽는다. 파일명의 최대 길이는 30자이며, 파일명에는 공백이 없다고 가정한다.
- 예외처리)
읽을 파일이 존재하지 않는 경우, 아래의 예시처럼 "Cannot open file!"을 출력한 후 프로그램을 종료한다.

```
Cannot open file!
```

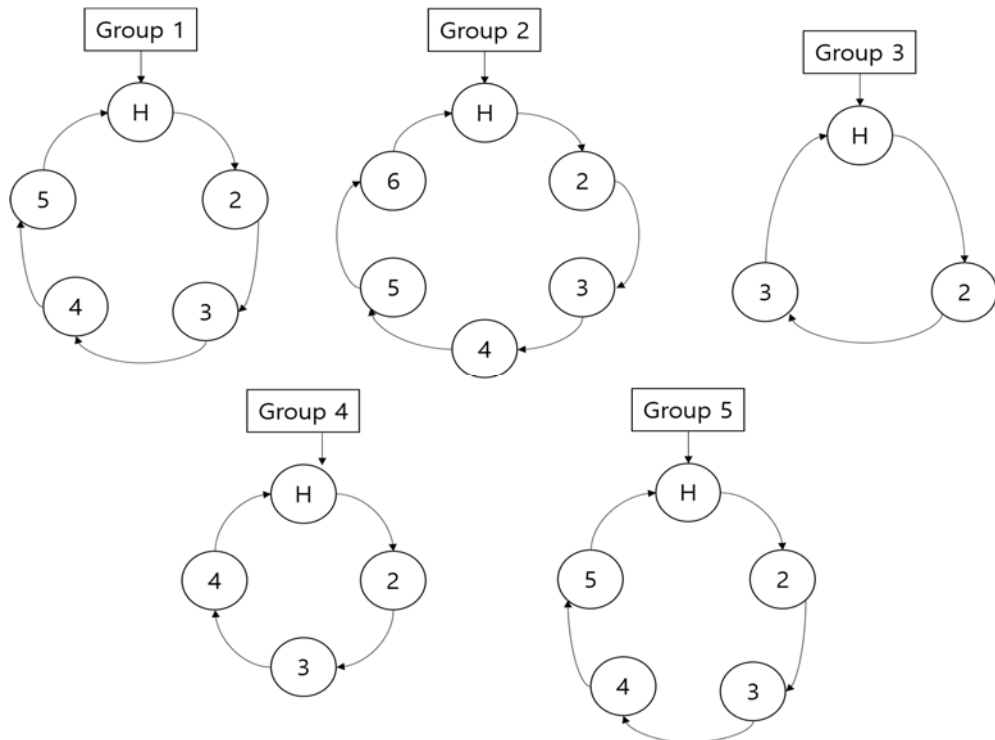
```
이 창을 닫으려면 아무 키나 누르세요.
```

- 프로그램 실행 시, 텍스트 파일로부터 플레이어에 대한 데이터를 읽으며, 텍스트 파일의 내용 및 구성은 아래와 같다.

Lee Jimin	F	Math	20160175	1
Kim Hyejeong	F	History	20180112	2
Go Jaewon	M	Art	20160525	4
Park Yeeun	F	Computer	20150132	3
Jang Kihun	M	Life	20170598	5
Choi Woong	M	Life	20130051	1
..				

(생략)

- 이름, 성별, 학과, 학번, 조로 구성되어 있으며 각 항목 사이는 tab(Wt)으로 구분된다.
- 한 사람의 정보와 정보 사이는 줄 바꿈 문자(Wn)로 구분된다.
Ex. [Name][Wt][Gender][Wt][Dept][Wt][StudentID][Wt][Group][Wn]
- 프로그램을 실행시키면 텍스트 파일로부터 데이터를 읽어 목록을 각 조의 연결리스트에 저장한다. 이 때, 파일에서 각 조별로 처음 등장하는 플레이어가 그 조의 조장(head)이 된다.
- 각 조가 하나의 원을 이루므로, 각 조의 마지막 플레이어가 다시 조장을 가리키도록 한다. 이를 그림으로 표현하면 다음과 같다. (H는 조장이자 1번 플레이어)



- 위 그림의 조별 인원은 임의로 나타낸 것이며, 실제 인원 수는 읽어들인 텍스트 파일로부터 정해진다.
- 그 후 전체 플레이어 목록을 그룹을 기준으로 1조부터 순서대로 아래의 예제처럼 출력한다.
- 전체 플레이어 목록을 출력한 후에는 명령어를 입력 받을 준비를 한다.

```

Name      Gender  Dept      StudentID
===== GROUP 1 =====
Lee Jimin   F      Math      20160175
Choi Woong  M      Life      20130051
.
.
(중략)
===== GROUP 5 =====
Jang Kihun  M      Life      20170598
.
.
(생략)

>>

```

(기능)

- 수건 돌리기 시뮬레이션 프로그램은 아래와 같은 기능을 가진다.
- 명시된 5가지 명령어 외의 명령어가 입력될 경우 무시하고 다시 명령어 입력을 대기한다. 이때, 5개의 명령어(show, add, remove, play, quit) 중 하나를 입력한다. 사용자가 명령어 입력 시, 대소문자를 구분하지 않고 동일한 명령어의 기능을 수행하도록 작성한다. 예를 들면, show, Show, sHow, SHOW 등은 동일한 동작을 수행한다.

1. SHOW 명령어

- show 입력 시, 어떤 조의 플레이어 정보를 출력할지 물어본 후 해당 조의 전체 플레이어의 정보를 출력한다. 이때 각 조 안에서 출력되는 순서는 head부터 저장되어 있는 순서이다.
- 조 번호로 0을 입력 받으면 전체 조의 플레이어 정보를 한꺼번에 출력한다. 아래의 예시를 참고한다.

```
>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Park Yeeun    F        Computer  20150132
Kang Wonsik   M        Art       20180144
Kim Taewoo    M        Computer  20170023
Won Jinah     F        Math      20160012
Jeong Taeyoung M        History   20180077

>>
```

- 특정 조가 비어 있는 경우 아래의 예시를 참고한다.

```
>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====

>>
```

2. ADD 명령어

- 새로운 플레이어가 게임에 참가하는 경우에 사용
- add 입력 시, 아래 예제와 같이 학번, 이름, 성별, 학과, 조를 순서대로 입력 받아 해당하는 조에 새 플레이어를 추가한다. 새로운 플레이어는 리스트의 마지막에 추가되도록 구현한다.
- show 입력 시, 새 플레이어의 정보가 포함되어 출력되어야 한다.

```
>>add
```

```

StudentID: 20190001
Name: Jang Seongbo
Gender: M
Dept: Computer
Group: 3
Player successfully added to group 3.

>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Park Yeeun    F        Computer  20150132
Kang Wonsik   M        Art       20180144
Kim Taewoo    M        Computer  20170023
Won Jinah     F        Math      20160012
Jeong Taeyoung M        History   20180077
Jang Seongbo  M        Computer  20190001

>>

```

- 이미 있는 플레이어를 입력한 경우, 이미 있다는 메시지와 함께 정보를 출력해 준 후 명령어 입력을 기다린다. 아래의 예시를 참고한다.

```

>>add
StudentID: 20160525
Player already exists!
Name      Gender  Dept      StudentID  Group
Go Jaewon  M        Art       20160525   4

>>

```

3. REMOVE 명령어

- 기존 플레이어가 게임에서 빠지는 경우에 사용
- remove 입력 시, 삭제하고자 하는 플레이어의 학번을 입력 받아 해당하는 플레이어 정보를 삭제한다.
- 아래의 예제는 플레이어 목록에 없는 학번을 입력한 경우의 예시이다.

```

>>remove
StudentID: 20190111
Player does not exist!

>>

```

- 아래의 예제는 목록에 있는 플레이어가 삭제된 예시로, show 입력 시 삭제된 플레이어 정보가 제외되어 출력된다.

```
>>remove
StudentID: 20190001
Player successfully removed from group 3.

>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Park Yeeun    F        Computer  20150132
Kang Wonsik   M        Art       20180144
Kim Taewoo    M        Computer  20170023
Won Jinah     F        Math      20160012
Jeong Taeyoung M        History   20180077

>>
```

4. PLAY 명령어

- 파일로부터 복수의 쿼리를 입력 받아 게임을 진행한 후, 게임 진행 결과를 화면에 출력한다.
- 쿼리 파일의 구성과 예시는 다음과 같다.

```
3 1
5 2
2 4
```

- 각 줄은 현재 술래가 수건을 넘겨줄 사람의 조 번호와 플레이어 번호로 구성되어 있다.
Ex. [nextGroup][][nextPlayer]
플레이어 1번은 조장을 의미하며, 나머지는 연결리스트에 저장된 순서를 따른다.
- 게임 시작 시 처음 술래는 1조 조장이다.
- 예를 들어, 위 파일의 순서대로 게임을 진행하면 다음과 같이 게임이 진행된다.
 1. 1조 조장(1번)이 3조 조장에게 수건을 건네고 자리를 차지한다. (즉, 1조 조장은 1조에서 사라지고 3조로 자리를 옮겨 새로운 조장이 된다.)
 2. 새로 술래가 된 3조 조장이 5조 2번에게 수건을 건네고 자리를 차지한다.
 3. 5조 2번이 2조 4번에게 수건을 건네고 자리를 차지한다.
 4. 여기서 게임이 종료되므로 2조 4번은 게임에서 추방된다. (즉, 플레이어 목록에서 사라진다.)
- nextGroup이 술래와 같은 조일 수 있다. 즉, 반드시 다른 조로 옮길 필요는 없다.
(실행 예시는 9쪽 "*play 명령어 참고 실행 예시*"를 참고할 것)
- nextPlayer가 해당 조의 플레이어 수보다 크더라도, 각 플레이어가 원형으로 연결되어

있기 때문에 그에 맞게 처리한다. 예를 들어, 조원이 5명인데 nextPlayer=7인 경우는 nextPlayer=2인 경우와 동일하게 처리한다.

- 예외처리) 비어 있는 조에 대하여:
 - nextGroup이 비어 있는 조일 경우, 이를 무시하고 다음 쿼리를 처리한다.
 - 1조가 비어 있는 경우 처음 술래는 2조 조장이다. 1~k조가 비어 있는 경우 처음 술래는 (k+1)조 조장이다. (실행 예시는 9쪽 "*play 명령어 참고 실행 예시*"를 참고할 것)
 - 모든 조가 비어 있는 경우, 즉 게임 참가자가 없는 경우는 게임을 진행할 수 없으므로 아래와 같은 메시지를 출력한 후 명령어 입력을 기다린다.

```
>>play game.txt
And then there were none...

>>
```

- 명령어는 아래와 같은 형태로 입력한다. 다음은 위 예시 파일이 "game.txt"라 가정했을 때의 실행 결과이다.

```
>>play game.txt
From                To
Lee Jimin (Group 1)  ->  Park Yeeun (Group 3)
Park Yeeun (Group 3) ->  Jang Sooah (Group 5)
Jang Sooah (Group 5) ->  Yoo Jiwoo (Group 2)

Game over!
Yoo Jiwoo (Group 2) is out.

>>
```

- 게임이 끝난 후 show를 하면 바뀐 자리와 탈락한 플레이어가 반영되어 출력되어야 한다. 아래의 예시로 게임 후, 1조 조장(20160175)이 3조 조장으로 자리가 변경되었음을 볼 수 있다.

```
>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Lee Jimin      F       Math      20160175
Kang Wonsik    M       Art        20180144
Kim Taewoo     M       Computer  20170023
Won Jinah      F       Math       20160012
Jeong Taeyoung M       History   20180077

>>
```


예외처리)

- 파일명 없이 단순히 play만 입력한 경우 기본값으로 "game.txt"로부터 쿼리를 읽어와 처리한다.
- 읽을 파일이 존재하지 않는 경우, 아래의 예시처럼 메시지 출력 후 다시 명령어 입력을 대기한다.

```
>>play fun.txt
Cannot open file!

>>
```

5. QUIT 명령어

- quit 입력 시, 프로그램을 종료한다.
- 해당 명령어를 실행할 때 현재 플레이어 목록을 "result_game.txt"에 저장한다. 파일 형식은 "players.txt"와 동일하게 한다.
[Name][Wt][Gender][Wt][Dept][Wt][StudentID][Wt][Group][Wn]
- 저장할 때 목록의 순서는 1조의 조장부터 현재 연결리스트 상의 순서와 동일하게 저장하도록 한다.
- 동적 할당된 memory를 모두 free 시킨 후, 종료한다.

```
>>quit
File saved. Bye!
이 창을 닫으려면 아무 키나 누르세요.
```

※ play 명령어 참고 실행 예시

- 1, 2조는 비어 있고 3조 목록이 아래와 같다고 가정하자.

```
>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Park Yeeun    F        Computer  20150132
Kang Wonsik   M        Art       20180144
Kim Taewoo    M        Computer  20170023
Won Jinah     F        Math      20160012
Jeong Taeyoung M        History   20180077

>>
```

- game.txt는 아래와 같이 구성되어 있다고 하자.

3 1

- 이 때 "play game.txt"를 명령어로 입력하면 첫 술래는 3조 조장인 "Park Yeeun"이 되고 다음 수건을 3조(같은 조) 조장에게 주게 되는데, 다음과 같은 순서를 따른다.
 1. "Park Yeeun"이 첫 술래로 3조에서 나가고 "Kang Wonsik"이 조장이 된다.
 2. 수건을 3조 조장에게 주어야 하므로 "Kang Wonsik"이 3조에서 나가고 "Park Yeeun"이 조장이 된다.
- 이를 정리하면 다음과 같다.

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

Game over!
Kang Wonsik (Group 3) is out.

>>show
Which group? (0: all) 3
Name          Gender  Dept      StudentID
===== GROUP 3 =====
Park Yeeun    F          Computer  20150132
Kim Taewoo    M          Computer  20170023
Won Jinah     F          Math      20160012
Jeong Taeyoung M          History   20180077

>>
```