

CSED101. Programming & Problem solving

Spring 2022

Programming Assignment #5 (70 points)

오하늘 (hnoh77@postech.ac.kr)

■ 제출 마감일: 2022.05.30 23:59

■ 개발 환경: Windows Visual Studio 2019

■ 제출물

- C 소스 코드 ([assn5.c](#))
 - 프로그램의 소스 코드에 채점자의 이해를 돕기 위한 주석을 반드시 붙여주세요.
- 보고서 파일 (.docx, .hwp 또는 .pdf; [assn5.docx](#), [assn5.hwp](#) 또는 [assn5.pdf](#))
 - 보고서는 AssnReadMe.pdf를 참조하여 작성하시면 됩니다.
 - 명예 서약 (Honor code): 표지에 다음의 서약을 기입하여 제출해 주세요: “나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.” 보고서 표지에 명예 서약이 기입되어 있지 않은 과제는 제출되지 않은 것으로 처리됩니다.
 - 작성한 소스 코드와 보고서 파일은 PLMS를 통해 제출해 주세요.

■ 주의 사항

- 컴파일이나 실행이 되지 않는 과제는 0점으로 채점됩니다.
- 제출 기한보다 하루 늦게 제출된 과제는 최종 20%, 이를 늦게 제출된 과제는 최종 40% 감점됩니다. 제출 기한보다 사흘 이상 늦으면 제출 받지 않습니다 (0점 처리).
- 각 문제의 제한 조건과 요구 사항을 반드시 지켜 주시기 바랍니다.
- 모든 문제의 출력 형식은 채점을 위해 아래에 제시된 예시들과 최대한 비슷하게 작성해 주세요.
- 부정행위에 관한 규정은 POSTECH 전자컴퓨터공학부 학부위원회의 “POSTECH 전자컴퓨터공학부 부정행위 정의”를 따릅니다 (PLMS의 본 과목 공지사항에 등록된 글 중, 제목이 [document about cheating]인 글에 첨부되어 있는 disciplinary.pdf를 참조하세요).
- 이번 과제는 추가 기능 구현과 관련된 추가 점수가 따로 없습니다.

■ Problem: 포항 지하철 노선 관리 프로그램

[문제]

포항에 지하철이 개설될 계획이고, 이를 위해 노선을 관리할 프로그램이 필요합니다. 포항 지하철 노선 관리 프로그램이란 기점부터 종점까지의 역 정보를 관리하는 프로그램으로, 기점부터 종점까지의 정렬된 각 역들에 대한 정보를 관리합니다. 이번 과제에서는 포항 지하철 노선에 관련된 역 정보를 연결리스트로 관리하는 것을 목적으로 합니다.

포항 지하철노선 관리 프로그램의 주요 기능은 아래와 같습니다.

1. 텍스트 파일로 저장된 역 정보를 읽고 역 리스트에 저장
2. 새로운 역 정보 추가
3. 기존의 역 정보 삭제
4. 기존의 역 정보 간의 순서 수정
5. 기점부터 종점까지의 역 정보 출력
6. 프로그램 종료

[목적]

이번 과제를 통하여 구조체(struct)와 연결리스트(linked list)의 사용법을 익힙니다.

[주의사항]

1. 이번 과제는 총 5개의 명령어 (표 2 참고)를 입력 받아 각 기능을 수행합니다. 최소한 각 명령어 별로 함수를 정의하여 사용합니다. 명령어 외에 필요한 함수는 정의하여 사용할 수 있습니다. 기능적으로 독립됐거나 반복적으로 사용되는 기능은 사용자 함수를 정의해서 구현하도록 합니다.
2. 이번 과제는 구조체와 연결리스트를 활용하는 것이 목표이므로, 문제에 구조체와 연결리스트를 언급한 부분(역 리스트 구현)을 배열을 이용하여 해결 시 감점 처리됩니다.
3. 전역 변수 및 goto 문은 사용할 수 없습니다.
4. string.h 에서 제공하는 라이브러리 함수를 사용해도 됩니다.
5. 명시된 에러 처리 외에는 고려하지 않아도 됩니다.
6. 문제의 출력 형식은 채점을 위해 아래의 실행 예시와 최대한 비슷하게 작성해 주세요.

[설명 및 요구사항]

- 프로그램 구현은 구조체와 연결리스트를 사용하여야 하며, 각 항목에 대한 제약은 다음과 같다.
- **역 리스트**: 역의 역명과 좌표 정보를 가진 구조체를 정의하고 연결리스트로 구현한다. 역 리스트 내의 역 정보 순서는 기점부터 종점까지의 순서이다.
- **역 정보**: 하나의 역 정보는 아래와 같은 구조체를 가지고 정의한다. 구조체 안의 변수에 대한 설명은 표 1 을 참고한다.

```
typedef struct Station {
    char name[21];
    int posX;
    int posY;
} STATION;
```

변수명	내용	제약
name	역명	띄어쓰기를 포함하지 않는 20 자 이내의 문자열
posX	좌표 X	0 부터 9 까지 범위 내의 정수
posY	좌표 Y	0 부터 9 까지 범위 내의 정수

[표 1. 구조체 내의 변수명에 대한 설명]

명령어	내용
show	전체 역 정보를 출력합니다.
add	역 정보 리스트에 새로운 역 정보를 추가합니다.
delete	역 정보 리스트에서 특정 역 정보를 삭제합니다.
move	역 정보 리스트에서 특정 역 정보의 순서를 변경합니다.
quit	프로그램을 종료합니다.

[표 2. 프로그램에서 사용되는 명령어]

[프로그램 시작 시]

1. 프로그램을 실행하면 아래와 같이 읽어올 파일명을 입력 받는다. 파일명에는 공백이 없으며 확장자를 포함하여 20 자를 넘지 않는다고 가정한다.

기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >>

- ✓ **예외처리 1:** 만약 존재하지 않는 파일명을 입력 받은 경우, 아래와 같은 에러 메시지를 출력 후 다시 파일명을 입력 받는다. (실행 예시의 밑줄은 사용자 입력에 해당)

기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >> wrong.txt

[Error #1] 파일 이름이 유효하지 않습니다.

기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >>

2. 파일을 열어 위에서부터 읽으며 연결리스트(이하 역 리스트)에 삽입한다. 최종적으로 역 리스트 내의 역 정보의 순서가 기점부터 종점까지의 순서가 된다. 주어진 파일 stations.txt의 내용과 구성은 아래와 같다.

```
Postech 0 0
Idong 0 4
CityHall 3 4
PohangStation 1 9
PohangAirport 6 8
```

- 텍스트 파일 내 역 정보는 기점부터 종점 순으로 정렬되어 있다고 가정한다.
 - 한 줄에 하나의 역 정보가 들어가며, 각 역 정보는 앞에서부터 차례로 역의 역명, 위치 정보인 x 좌표와 y 좌표로 구성된다. 이들은 공백으로 구분되어 있다.
 - 역명은 20 자 이내의 문자열이며, 문자열은 공백을 포함하지 않으며, 중복된 역명이 없는 유효한 정보라고 가정한다.
 - 위치 정보는 (0, 0) ~ (9, 9) 사이의 좌표 정보로 제한하여, x 좌표와 y 좌표는 0 부터 9 까지 사이의 정수이다. 또한 중복된 위치 정보는 없으며 역간 거리가 3 이상인 유효한 정보라고 가정한다. (역간 거리에 대한 설명은 6 쪽을 참조할 것)
3. 역 리스트 내의 전체 정보를 기점부터 종점 순으로 아래와 같이 출력한다. 이후 [역 리스트 관리]를 시작한다.

기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >> stations.txt

=====

번호	역명	위치(좌표)
1 [기점]	Postech	(0, 0)
2	Idong	(0, 4)
3	CityHall	(3, 4)
4	PohangStation	(1, 9)
5 [종점]	PohangAirport	(6, 8)

=====

[total: 5 stations]

명령어를 입력해주세요. >>

- 위 예시와 같이 기점과 종점을 표시해주며, 역 정보가 1 개만 존재하는 경우에는 아래와 같이 기점만을 표시하면 된다.

기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >> stations.txt

```
=====
번호      역명      위치(좌표)
-----
1 [기점]   Postech      (0, 0)
=====
[total: 1 stations]

명령어를 입력해주세요. >>
```

✓ **예외처리 2:** 유효하지 않은 명령어를 입력 받은 경우, 아래와 같이 에러 메시지를 출력 후, 명령어를 재입력 받는다.

```
명령어를 입력해주세요. >> apple
[Error #2] 유효하지 않은 명령어입니다.

명령어를 입력해주세요. >>
```

[역 리스트 관리]

- 역 리스트 관리를 위해서는 "show", "add", "delete", "move", "quit" 로 총 5 개의 명령어가 필요하다. "quit" 명령어를 입력 받기 전까지는 계속해서 명령어를 입력 받고 그에 따른 작업을 수행한다.
- 단, 이 프로그램의 모든 예외처리는 모든 입력을 받고나서 처리하며, 관련 메시지를 모두 출력 후 해당 기능을 마치도록 한다.

■ show 명령어: 역 리스트 내의 모든 역 정보를 출력

1. 역 리스트 내의 모든 역 정보를 기점부터 종점까지의 순서대로, 순서와 함께 출력한다.
2. 맨 아래에는 역 정보의 총 개수를 출력한다.

```
명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]   Postech      (0, 0)
2          Idong      (0, 4)
3          CityHall    (3, 4)
4          PohangStation (1, 9)
5 [종점]   PohangAirport (6, 8)
=====
[total: 5 stations]

명령어를 입력해주세요. >>
```

✓ **예외처리 3:** 역 리스트에 역 정보가 존재하지 않을 경우, 아래와 같이 에러 메시지를 출력한다.

```
명령어를 입력해주세요. >> show
[Error #3] 역 리스트가 비어 있습니다.

명령어를 입력해주세요. >>
```

■ add: 역 리스트에 새로운 역 정보를 추가

1. 추가될 리스트 내 위치와 추가할 역 정보(name, posX, posY)를 순서대로 입력 받아 Station

구조체를 할당해 역 리스트의 올바른 위치에 추가한다.

```
명령어를 입력해주세요. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 6
+ 추가할 역 정보의 역명을 입력해주세요. >> Homigot
+ 추가할 역 정보의 위치 정보 x 좌표를 입력해주세요. >> 6
+ 추가할 역 정보의 위치 정보 y 좌표를 입력해주세요. >> 4

명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]   Postech    (0, 0)
2          Idong     (0, 4)
3          CityHall  (3, 4)
4          PohangStation (1, 9)
5          PohangAirport (6, 8)
6 [종점]   Homigot    (6, 4)
=====
[total: 6 stations]

명령어를 입력해주세요. >>
```

2. 추가될 리스트 내 위치는 1 부터 (현재 역 리스트의 총 역 정보 수 + 1) 사이의 수이다.

✓ **예외처리 4:** 범위 밖의 수를 입력된 경우, 아래와 같이 에러 메시지를 출력한다.

```
명령어를 입력해주세요. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 100
+ 추가할 역 정보의 역명을 입력해주세요. >> YeongiIdaeBeach
+ 추가할 역 정보의 위치 정보 x 좌표를 입력해주세요. >> 9
+ 추가할 역 정보의 위치 정보 y 좌표를 입력해주세요. >> 4
[Error #4] 100 번째의 위치에 추가할 수 없습니다.

명령어를 입력해주세요. >>
```

3. 추가할 역 정보(name, posX, posY)가 다음 조건에 부합하는 경우에만 새로운 역으로 추가한다.

- A. 역 리스트 내에 name 을 역명으로 하는 역이 존재하지 않음
- B. 역 리스트 내에 (posX, posY)을 위치로 하는 역이 존재하지 않음
- C. 모든 역과의 구간 거리가 3 이상의 값을 가져야 함

$$\text{역 1 (posX=x}_1\text{, posY=y}_1\text{)과 역 2 (posX=x}_2\text{, posY=y}_2\text{) 간의 구간 거리} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

✓ **예외처리 5:** 위 조건 중 적어도 하나라도 부합되지 않는 경우, 아래와 같이 에러 메시지를 출력한다.

```
명령어를 입력해주세요. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 4
+ 추가할 역 정보의 역명을 입력해주세요. >> Postech
+ 추가할 역 정보의 위치 정보 x 좌표를 입력해주세요. >> 9
+ 추가할 역 정보의 위치 정보 y 좌표를 입력해주세요. >> 4
[Error #5] 유효하지 않은 역 정보입니다.

명령어를 입력해주세요. >>
```

- ✓ 앞서 말했듯이, 여러 개의 예외처리에 해당되는 경우에는 관련 예외처리에 대한 메시지를 모두 출력하게 된다. 아래의 예시는 예외처리 4와 5가 함께 해당되는 경우이다.

```
명령어를 입력해주세요. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 100
+ 추가할 역 정보의 역명을 입력해주세요. >> Postech
+ 추가할 역 정보의 위치 정보 x 좌표를 입력해주세요. >> 9
+ 추가할 역 정보의 위치 정보 y 좌표를 입력해주세요. >> 4
[Error #4] 100 번째의 위치에 추가할 수 없습니다.
[Error #5] 유효하지 않은 역 정보입니다.

명령어를 입력해주세요. >>
```

4. 아래는 정상적인 추가의 예시이다.

```
명령어를 입력해주세요. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 4
+ 추가할 역 정보의 역명을 입력해주세요. >> YeongildaeBeach
+ 추가할 역 정보의 위치 정보 x 좌표를 입력해주세요. >> 9
+ 추가할 역 정보의 위치 정보 y 좌표를 입력해주세요. >> 4

명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]   Postech   (0, 0)
2          Idong    (0, 4)
3          CityHall  (3, 4)
4          YeongildaeBeach (9, 4)
5          PohangStation (1, 9)
6          PohangAirport (6, 8)
7 [종점]   Homigot   (6, 4)
=====
[total: 7 stations]

명령어를 입력해주세요. >>
```

- delete 명령어: 역 리스트 내 특정 위치의 역 정보를 삭제

1. 몇 번째 순서의 역 정보를 삭제할지 입력 받는다.

- ✓ 예외처리 6: 1 보다 작거나 (현재 역 리스트의 총 개수) 보다 큰 수가 입력된 경우, 아래와 같이 여러 메시지를 출력한다.

```
명령어를 입력해주세요. >> delete
+ 몇 번째의 역 정보를 삭제할까요? >> -1
[Error #6] -1 번째의 역 정보는 존재하지 않습니다.

명령어를 입력해주세요. >>
```

2. 입력 받은 순서의 역 정보를 역 리스트에서 삭제한다. 삭제 시에는 해당 역 정보의 저장을 위해 동적 할당 받은 메모리를 할당 해제(free) 한다.

```
명령어를 입력해주세요. >> delete
+ 몇 번째의 역 정보를 삭제할까요? >> 1

명령어를 입력해주세요. >> show
```

```
=====
번호      역명      위치(좌표)
-----
1 [기점]   Idong      (0, 4)
2          CityHall (3, 4)
3          YeongildaeBeach (9, 4)
4          PohangStation (1, 9)
5          PohangAirport (6, 8)
6 [종점]   Homigot    (6, 4)
=====
[total: 6 stations]

명령어를 입력해주세요. >>
```

✓ **예외처리 7:** 만약 역 리스트가 비어 있는 경우, 아래와 같이 에러 메시지를 출력한다.

```
명령어를 입력해주세요. >> delete
+ 몇 번째의 역 정보를 삭제할까요? >> 1
[Error #7] 역 리스트가 비어 있습니다.
[Error #6] 1 번째의 역 정보는 존재하지 않습니다.

명령어를 입력해주세요. >>
```

● move 명령어: 역 리스트 내 역 정보의 순서 변경

1. 몇 번째 역 정보를 이동할 것인지와 해당 역 정보를 몇 번째 순서로 이동할 것인지를 입력 받는다.

✓ **예외처리 8:** 두 개의 입력으로 적어도 하나라도 1 보다 작거나 (현재 역 리스트의 총 개수) 보다 큰 수가 입력된 경우, 아래와 같이 에러 메시지를 출력한다.

```
명령어를 입력해주세요. >> move
+ 몇 번째의 역 정보를 이동할까요? >> -1
+ -1 번째 역 정보를 몇 번째로 이동할까요? >> 4
[Error #8] 유효하지 않은 입력입니다.

명령어를 입력해주세요. >>
```

2. 입력 받은 정보를 바탕으로 올바른 위치로 역 리스트를 수정한다.
 - 데이터 정보 swap 과는 다르다. 특정 역 정보의 순서가 변경됨에 따라 역 정보가 뒤로 밀리며 이동한다.
 - 이동할 위치와 이동될 위치가 같은 경우, 역 리스트 내 정보가 변경되면 안 된다.

```
명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]   Idong      (0, 4)
2          CityHall (3, 4)
3          YeongildaeBeach (9, 4)
4          PohangStation (1, 9)
5          PohangAirport (6, 8)
6 [종점]   Homigot    (6, 4)
=====
[total: 6 stations]

명령어를 입력해주세요. >> move
+ 몇 번째의 역 정보를 변경할까요? >> 1
```



```
+ 1 번째 역 정보를 몇 번째로 변경할까요? >> 3

명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]  CityHall  (3, 4)
2         YeongildaeBeach (9, 4)
3         Idong      (0, 4)
4         PohangStation (1, 9)
5         PohangAirport (6, 8)
6 [종점]  Homigot   (6, 4)
=====
[total: 6 stations]

명령어를 입력해주세요. >>
```

• quit: 프로그램 종료

1. 현재까지 편집한 내용을 파일명을 입력 받아서 저장하도록 한다.
앞서 본 "stations.txt"와 같은 형식으로 내용을 구성한다.
 - 파일명은 확장자를 포함하여 최대 길이가 20 자이며, 파일명에 공백이 없다고 가정한다.
 - 종료 시, 동적 할당 받은 memory 를 모두 free 시킨 후 종료한다.

```
명령어를 입력해주세요. >> show
=====
번호      역명      위치(좌표)
-----
1 [기점]  CityHall  (3, 4)
2         YeongildaeBeach (9, 4)
3         Idong      (0, 4)
4         PohangStation (1, 9)
5         PohangAirport (6, 8)
6 [종점]  Homigot   (6, 4)
=====
[total: 6 stations]

명령어를 입력해주세요. >> quit
+ 저장할 파일 이름을 입력해주세요. >> new.txt

이 창을 닫으려면 아무 키나 누르세요...
```

- 아래는 생성된 new.txt 파일의 내용이다.

```
CityHall 3 4
YeongildaeBeach 9 4
Idong 0 4
PohangStation 1 9
PohangAirport 6 8
Homigot 6 4
```

✓ **예외처리 9:** 역 리스트가 비어 있는 경우, 아래와 같이 에러 메시지를 출력하고 파일로 저장하지 않는다.

```
명령어를 입력해주세요. >> show  
[Error #3] 역 리스트가 비어 있습니다.  
  
명령어를 입력해주세요. >> quit  
+ 저장할 파일 이름을 입력해주세요. >> new.txt  
[Error #9] 역 리스트에 저장할 정보가 없습니다.  
  
이 창을 닫으려면 아무 키나 누르세요...
```