

2022년 1학기

프로그래밍과 문제해결

Assignment #5

담당교수: 윤은영

학번: 20220100

학과: 무은재학부

이름: 박기현

POVIS ID: kihyun

명예서약(Honor Code)

“나는 이 프로그래밍 과제를 다른 사람의 부적절한 도움 없이 완수하였습니다.”

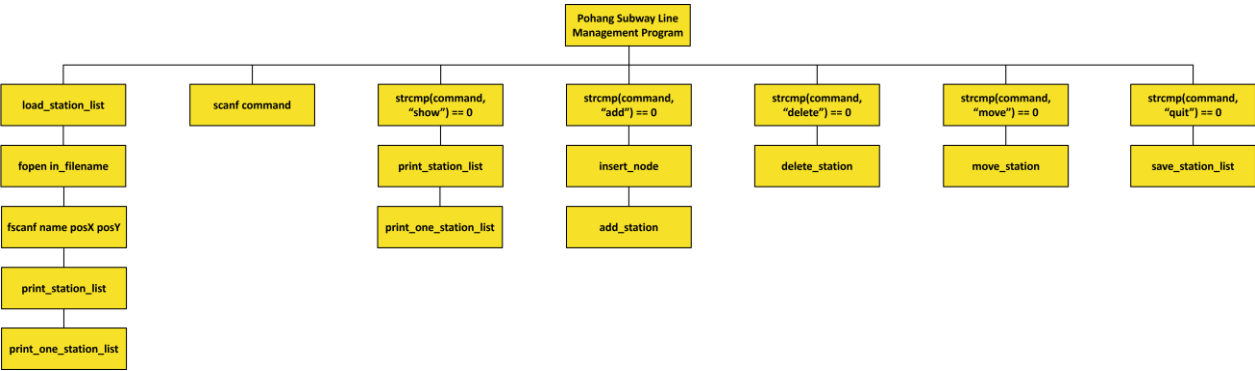
Problem. 포항 지하철 노선 관리 프로그램

1. 문제의 개요

본 프로그램을 간략히 설명하면 다음과 같다.

- 프로그램을 실행하면 읽어올 파일명을 입력 받아 역 정보를 출력한다.
- 명령어를 입력 받으면 명령어에 따른 작업을 수행한다.
- 프로그램을 종료하면 편집한 역 정보를 파일명을 입력 받아서 저장한다.

이 때 사용되는 구상 가능한 구조 차트(structure chart)는 아래와 같이 표현될 수 있다.



2. 알고리즘

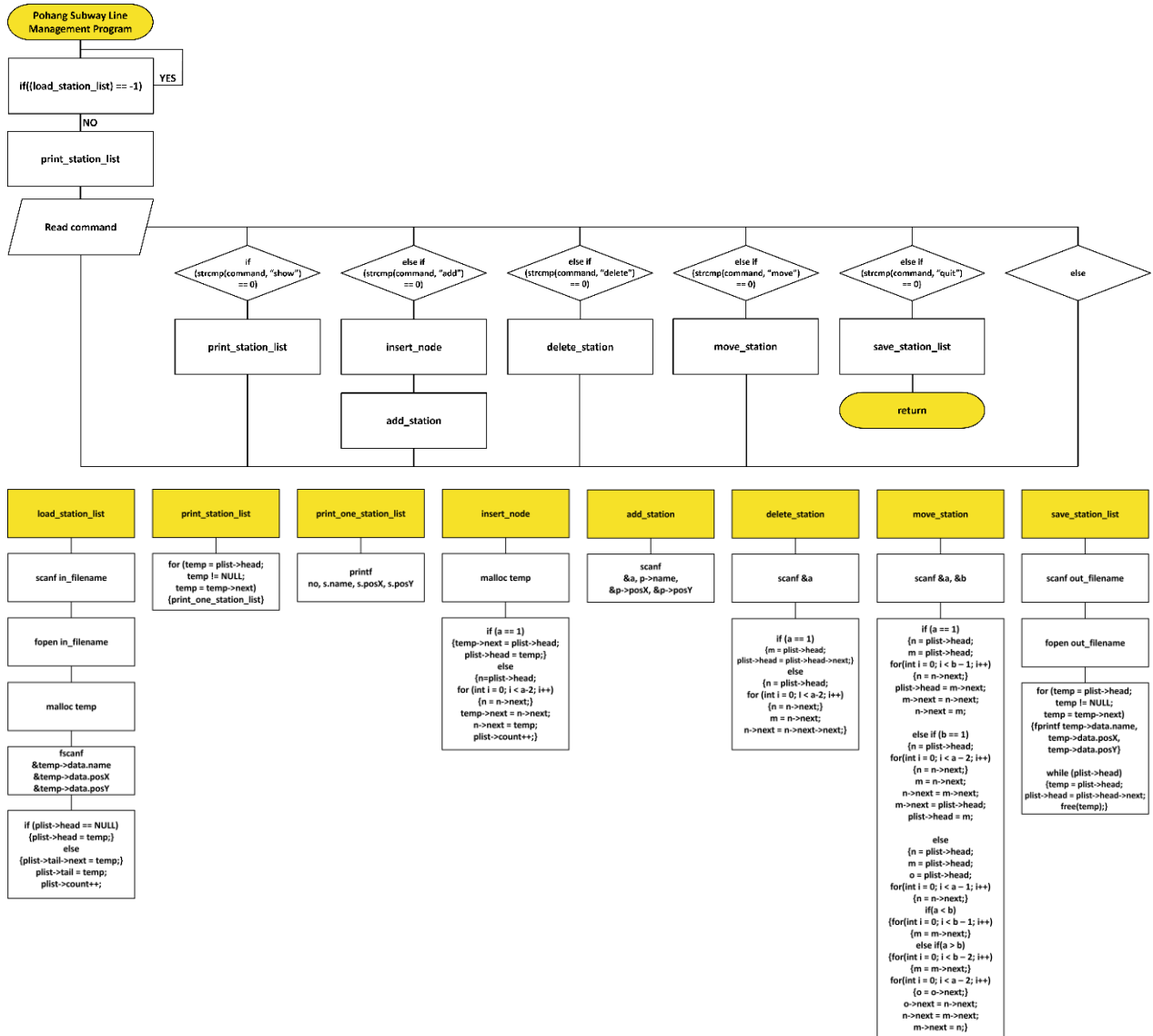
본 프로그램 작성을 위한 알고리즘을 Pseudo 코드 형태로 나타내면 다음과 같다.

Pseudo-algorithm for Pohang Subway Line Management Program

// 프로그램에 필요한 변수들은 미리 선언해놓은 것으로 가정한다.

```
1 include an essential header <stdio.h> / <stdlib.h> / <string.h> / <math.h>
2 typedef struct STATION
3 typedef struct NODE
4 typedef struct LIST
5 while(1)
    if ((load_station_list(&station)) == -1)
        continue
    print_station_list(&station)
    while(1)
        printf
        scanf command
        if (strcmp(command, "show") == 0)
            print_station_list
            continue
        else if (strcmp(command, "add") == 0)
            insert_node
            continue
        else if (strcmp(command, "delete") == 0)
            delete_station
            continue
        else if (strcmp(command, "move") == 0)
            move_station
            continue
        else if (strcmp(command, "quit") == 0)
            save_station_list
            break
```

위의 의사 알고리즘을 flow chart를 통해 표현하면 아래와 같다.



3. 프로그램 구조 및 설명

a) 프로그램 시작

- 프로그램을 실행하면 읽어올 파일명을 입력 받는다.
- 만약 존재하지 않는 파일명을 입력 받은 경우, [Error #1]을 출력 후 다시 파일명을 입력 받는다.
- 파일을 열어 위에서부터 읽으며 역 리스트에 삽입한다.
- 최종적으로 역 리스트 내의 역 정보의 순서가 기점부터 종점까지의 순서가 된다.
- 역 리스트 내의 전체 정보를 기점부터 종점 순으로 출력한다.
- 이후 역 리스트 관리를 시작한다.

b) 역 리스트 관리 - 명령어 입력

- 명령어를 입력 받아 역 리스트를 관리한다.
- 역 리스트 관리를 위해서는 "show", "add", "delete", "move", "quit"로 총 5개의 명령어가 필요하다.
- "quit" 명령어를 입력 받기 전까지는 계속해서 명령어를 입력 받고 그에 따른 작업을 수행한다.
- 유효하지 않은 명령어를 입력 받은 경우, [Error #2]를 출력 후, 명령어를 재입력 받는다.

c) show 명령어

- 역 리스트 내의 모든 역 정보를 기점부터 종점까지의 순서대로, 순서와 함께 출력한다.
- 맨 아래에는 역 정보의 총 개수를 출력한다.
- 역 리스트에 역 정보가 존재하지 않을 경우, [Error #3]을 출력한다.

d) add 명령어

- 추가될 리스트 내 위치와 추가할 역 정보(name, posX, posY)를 순서대로 입력 받아 Station 구조체를 할당해 역 리스트의 올바른 위치에 추가한다.
- 추가될 리스트 내 위치는 1부터 (현재 역 리스트의 총 역 정보 수 +1) 사이의 수다.
- 범위 밖의 수가 입력된 경우, [Error #4]를 출력한다.
- 추가할 역 정보(name, posX, posY)가 다음 조건에 부합하는 경우에만 새로운 역으로 추가한다.

A. 역 리스트 내에 name을 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

- 위 조건 중 적어도 하나라도 부합되지 않은 경우, [Error #5]를 출력한다.
- 여러 개의 예외처리에 해당되는 경우에는 관련 예외처리에 대한 메시지를 모두 출력한다.

e) delete 명령어

- 몇 번째 순서의 역 정보를 삭제할지 입력 받는다.
- 1보다 작거나 (현재 역 리스트의 총 개수)보다 큰 수가 입력된 경우, [Error #6]을 출력한다.
- 입력 받은 순서의 역 정보를 역 리스트에서 삭제한다.
- 삭제 시에는 해당 역 정보의 저장을 위해 동적 할당 받은 메모리를 할당 해제한다.
- 만약 역 리스트가 비어 있는 경우, [Error #7]을 출력한다.

f) move 명령어

- 몇 번째 역 정보를 이동할 것인지와 해당 역 정보를 몇 번째 순서로 이동할 것인지를 입력 받는다.
- 두 개의 입력 중 적어도 하나라도 1보다 작거나 (현재 역 리스트의 총 개수)보다 큰 수가 입력된 경우, [Error #8]을 출력한다.
- 입력 받은 정보를 바탕으로 올바른 위치로 역 리스트를 수정한다.

g) quit 명령어

- 현재까지 편집한 내용을 파일명을 입력 받아서 저장한다.
- 종료 시, 동적 할당 받은 memory를 모두 free 시킨 후 종료한다.
- 역 리스트가 비어 있는 경우, [Error #9]를 출력하고 파일로 저장하지 않는다.

4. 프로그램 실험방법 및 예제

```
C:\Windows\system32\cmd.exe
기존의 역 리스트를 저장한 파일 이름을 입력해주세요. >> wrong.txt
[Error #1] 파일 이름이 유효하지 않습니다.
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. >> apple
[Error #2] 유효하지 않은 명령어입니다.
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. >> 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. >> add
+ 몇 번째의 역 정보로 추가할까요? >> 100
+ 추가할 역 정보의 역명을 입력해주세요. >> YeongildaeBeach
+ 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. >> 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. >> 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.
```



```
C:\Windows\system32\cmd.exe
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
+ 몇 번째의 역 정보를 삭제할까요? >> -1

[Error #6] -1 번째의 역 정보는 존재하지 않습니다.

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. >> move
+ 몇 번째의 역 정보를 이동할까요? >> -1
+ -1 번째 역 정보를 몇 번째로 이동할까요? >> 4

[Error #8] 유효하지 않은 입력입니다.

명령어를 입력해주세요. >> move
+ 몇 번째의 역 정보를 이동할까요? >> 1
+ 1 번째 역 정보를 몇 번째로 이동할까요? >> 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
+ 저장할 파일 이름을 입력해주세요. >> new.txt

계속하려면 아무 키나 누르십시오 . . .
```

5. 토론

- 연결 리스트 구현에 있어 '->'를 써야 할지, '.'을 써야 할지 이해하는 데 어려움이 있었다.
- 명령어에 따라 구현해야 하는 기능이 다양해서 하나하나 구현하는데 시간이 걸렸다.
- move 명령어를 구현할 때 조건을 나누어야 했던 점과 새로운 NODE 선언을 통해 새로 연결하는 과정이 머릿속으로 잘 그려지지 않아 많이 헤맸던 것 같다.

6. 결론

- 본 과제는 사용자 정의 함수와 반복문, 조건문, 포인터, 구조체, 연결리스트, 텍스트 파일을 통한 입출력 등 C언어의 다양한 종합적인 기능을 정확히 이해하고 있어야 하는 문제였다. 문제를 해결하는 과정에서 어려움을 겪었던 연결리스트를 다시 한 번 이해하는 시간을 가질 수 있었다. 디버깅을 하는 시간이 많아지면서 디버깅에 대해서 훨씬 능숙해질 수 있었던 것 같다.

7. 개선방향

- 이때까지 배웠던 개념을 종합적으로 활용하는 문제였던 만큼 무엇을 활용해서 기능을 구현하면 훨씬 더 좋은 프로그램을 만들 수 있을지 고민하고 개선해 나가면 더 좋은 프로그램을 만들 수 있을 것이라 생각했다.