

# 시스템프로그래밍실습 보고서

## Assignment 1-3

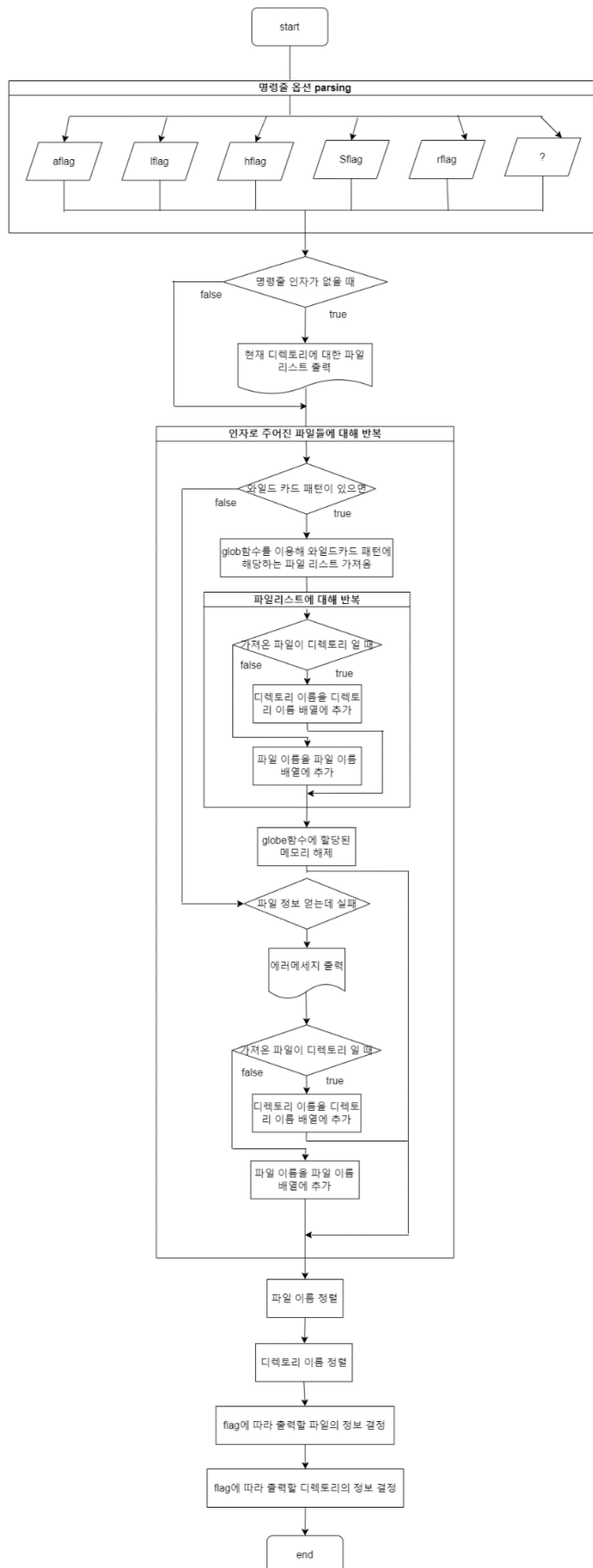
과목	시스템프로그래밍실습
담당교수	이기훈교수님
학과	컴퓨터정보공학부
학번	2021202058
이름	송채영

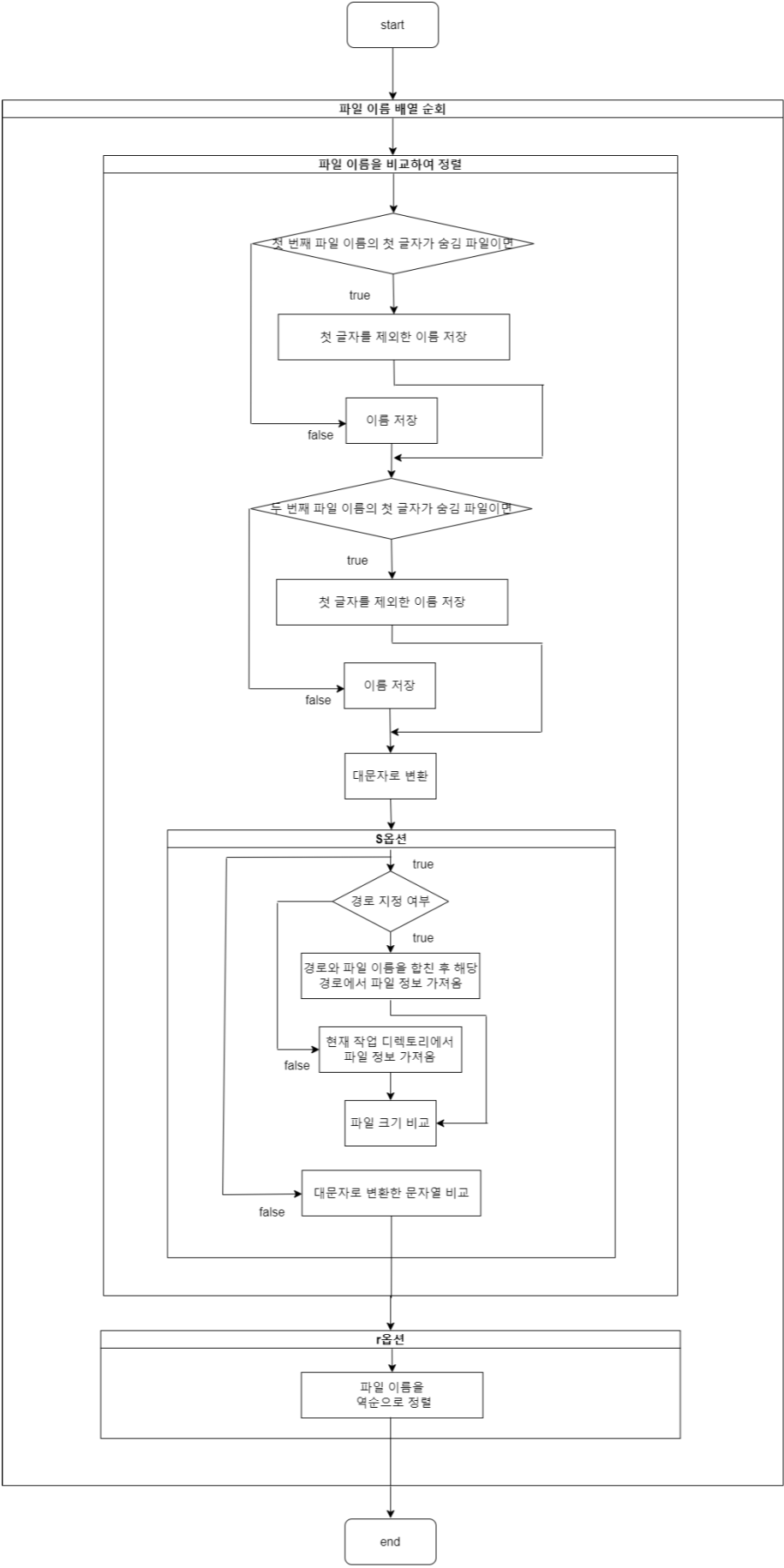
## 1. Introduction

지난 과제에는 ls 명령어와 ls 명령어의 옵션(-a, -l, -al)을 구현하였는데, 여기에 추가적으로 ls 명령어의 옵션(-h, -S, -r)을 직접 구현해본다. 또한 Wild card pattern 인 '\*', '?', '[seq]'와 같은 특수문자를 사용하여 파일이나 디렉토리를 검색할 수 있는 기능인 wildcard matching 을 추가하여 구현한다.

## 2. Flow chart

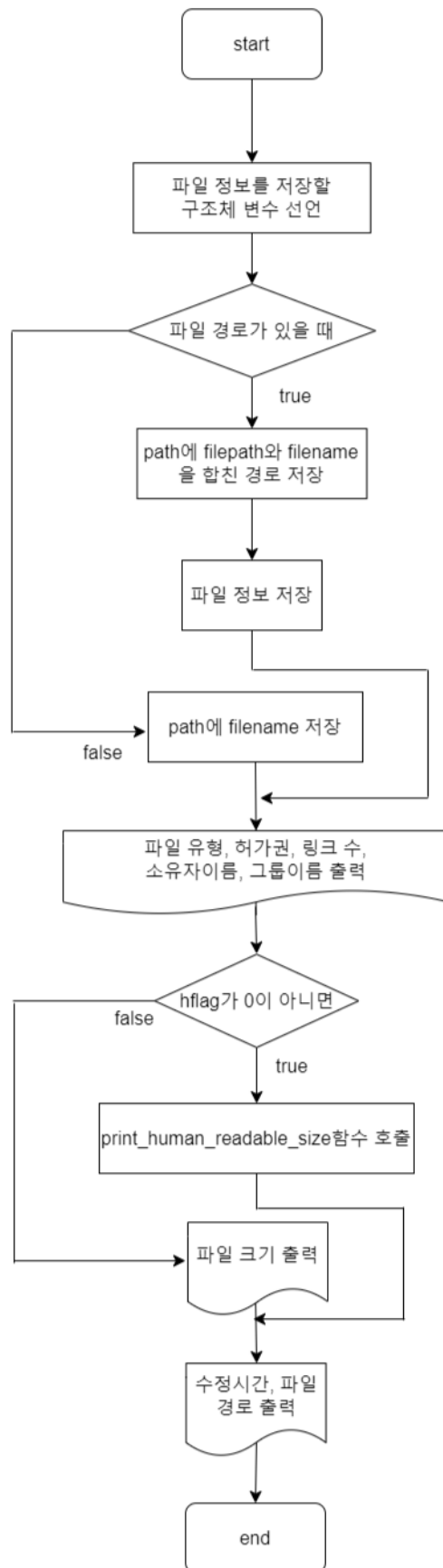
코드의 전체적인 흐름을 flow chart 로 나타내었다. main 함수에 대해 자세히 설명해보면 다음과 같다. 프로그램이 시작하면 getopt 함수를 사용해 명령줄 인자를 parsing 한다. getopt 함수는 argv 배열과 인자의 개수 argc, 그리고 옵션 문자열을 입력으로 받아 옵션 문자 하나를 읽어온다. 옵션이 있는지 없는지는 flag 로 판단한다. 인자가 없을 때, 현재 디렉토리에 대한 파일 리스트를 출력하며, 인자가 있을 때, 인자로 주어진 파일들에 대해 반복하며 와일드카드 패턴이 있으면, glob 함수를 이용해 와일드카드 패턴에 해당하는 파일들의 리스트를 가져온다. 가져온 파일이 디렉토리 일 때 디렉토리 이름을 디렉토리 이름 배열에 추가하며 파일일땐 파일 이름을 파일 이름 배열에 추가한 후 glob 함수에 할당된 메모리를 해제한다. 와일드카드 패턴이 없으면 파일 정보를 얻는 데 실패한 경우 에러 메시지를 출력해주며, 현재 가져온 파일이 디렉토리면, 디렉토리 이름을 디렉토리 이름 배열에, 파일이면 파일 이름을 파일 이름 배열에 추가한다. 이후 파일이름과 디렉토리 이름을 정렬해준 후 flag 에 따라 출력할 파일의 정보를 결정한다.

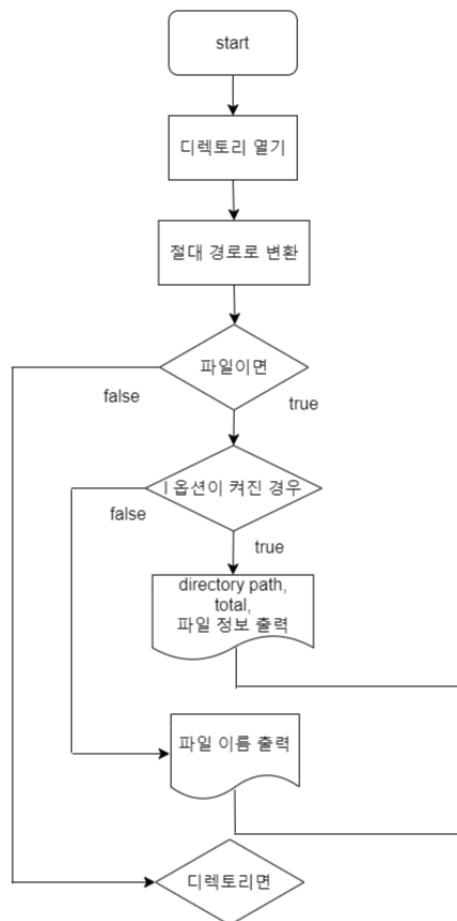
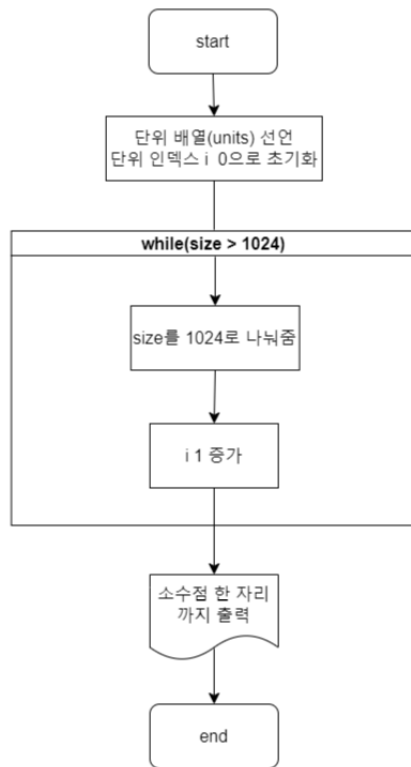


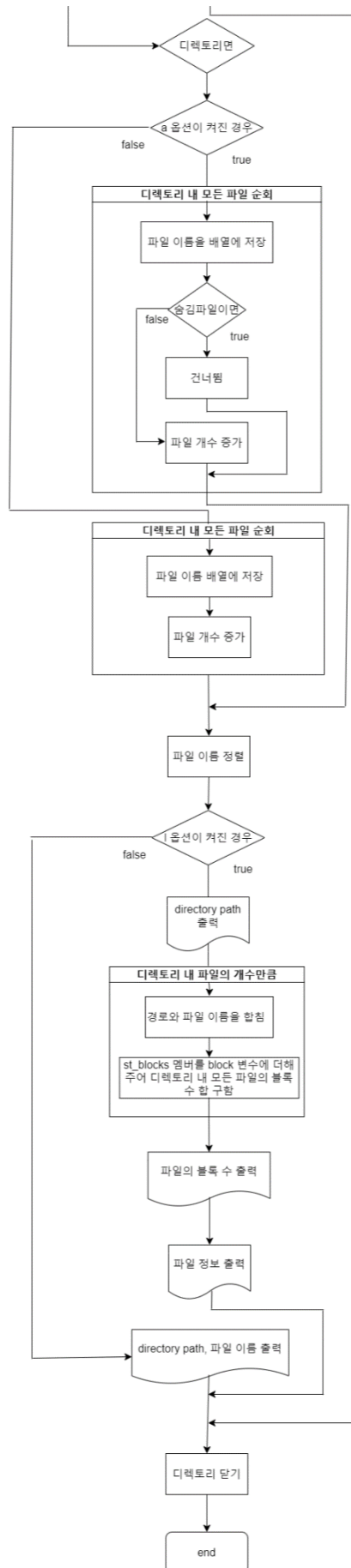


다음은 sort 함수의 flow chart 이다. 우선 파일의 개수만큼 반복한다. 첫 번째 파일이 숨김 파일일 경우, 파일의 이름을 길이만큼 반복하고 첫 글자를 제외한 이름을 저장한다. 숨김파일이 아닐 경우 이름을 저장해준다. 두 번째 파일도 동일하게 진행한다. 먼저 대문자로 변환해준 후 S 옵션을 선택한 경우와 아닌 경우로 나눈다. 전자는 경로가 지정되지 않은 경우 파일 이름만 stat 에 넣어주며, 경로가 있는 경우 경로와 파일 이름을 합친다. 이후 파일 크기를 비교해준다. 후자는 대문자로 변환한 문자열을 비교한다. r 옵션을 선택한 경우 파일 이름을 역순으로 정렬한다. 프로그램을 종료한다.

다음은 print\_file\_info, 파일 정보를 출력하는 함수의 flow chart 이다. 파일 정보를 읽어 구조체 변수인 filestat 에 저장한다. 파일 경로가 있고, filpath 와 filename 이 같지 않으면 경로와 파일 이름을 합치며, 파일 경로가 없는 경우 파일이름을 정보로 읽는다. 파일 유형, 허가권, 링크 수 , 소유자를 출력한다. hflag 가 0 이 아니면 함수를 호출해주며, 0 일 경우 파일 크기를 출력한다. 이어서 수정 시간, 파일 경로를 출력한 후 프로그램을 종료한다.

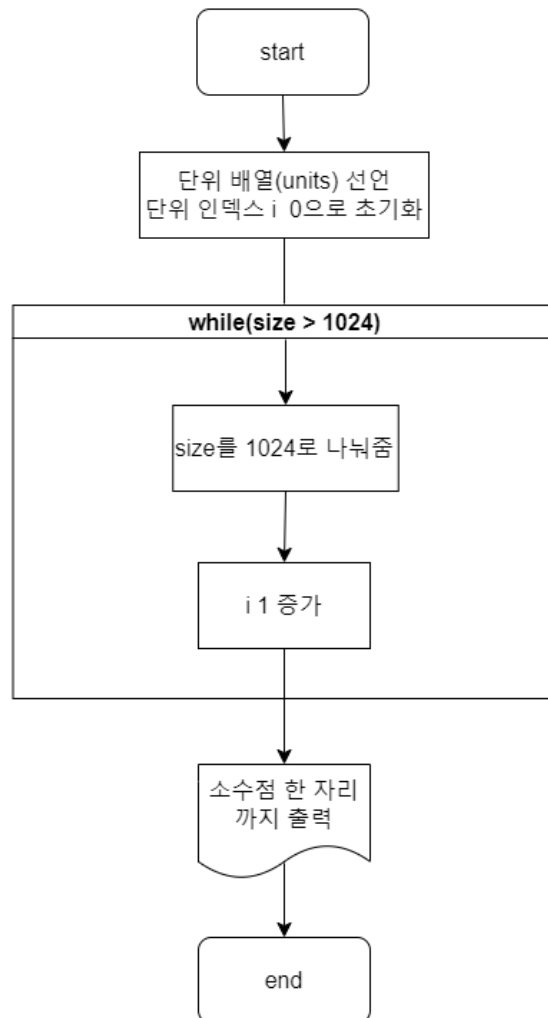




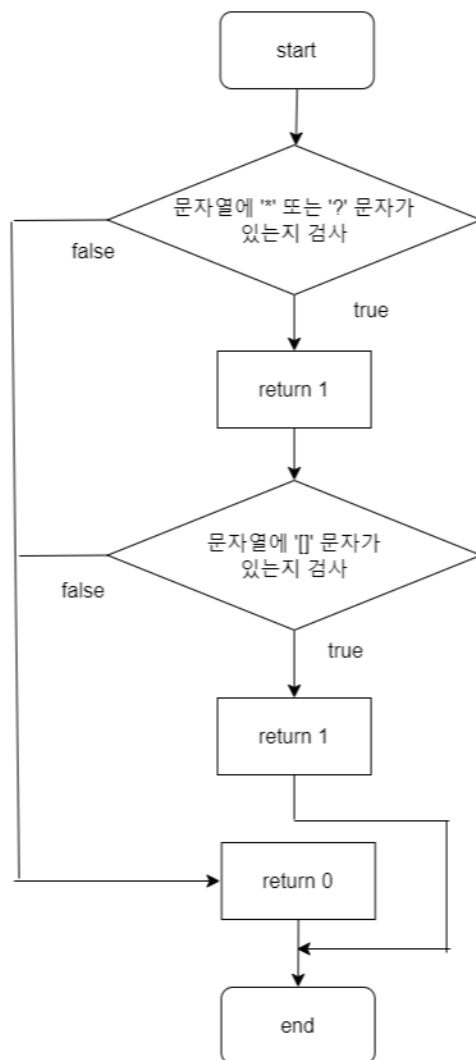




다음은 list files 함수의 flow chart 이다. 파일을 열어준 후 절대경로로 변환한다. 연  
파일이 파일일때와 디렉토리 일 때로 나눈다. 우선 파일이고 l 옵션이 켜져 있으면  
directory path 와 total 과 파일 정보를 출력한다. l 옵션이 꺼져 있을 경우 파일 이름만  
출력한다. 디렉토리일 경우 먼저 -a 옵션이 꺼져 있으면, 디렉토리 내의 모든 파일을  
순회하며, 파일 이름을 배열에 저장한다. 이때 숨김 파일이면 건너뛰며 아닐 경우 파일  
개수를 증가한다. a 옵션이 켜져 있을 경우, 디렉토리 내의 모든 파일을 순회하며 파일  
이름을 배열에 저장한다. 이후 파일 이름을 정렬한다. 다음으로 l 옵션이 켜져 있을 경우  
directory path, total 을 출력한다. 디렉토리 내의 파일 개수만큼 경로와 파일을 합친 후  
block 변수에 st\_blocks 멤버를 더해주어 디렉토리 내의 모든 파일의 블록수의 합을 구한  
후 출력해준다. 파일 정보를 출력한다. l 옵션이 꺼져 있을 경우, directory path 와 파일  
이름을 출력하고 디렉토리를 닫고 프로그램을 종료한다.



다음은 print\_human\_readable\_size 함수이며, 먼저 단위 배열인 units 를 선언하고 단위  
인덱스 i 를 0 으로 초기화한다. size 가 1024 가 넘어가면 단위를 증가시키며 나눠준다.  
소수점 한 자리까지 출력해주고 프로그램을 종료한다.



마지막으로 wildcardpattern 함수이다. 먼저 문자열에 '\*', '?' 문자가 있는지 확인하여 있으면 return 1 을 해준다. 문자열에 '[' 문자가 있는지 확인해서 문자가 있으면 return 1 을 한다. 와일드카드 문자가 없으면 0 을 반환하며 프로그램을 종료한다.

### 3. Pseudo Code

```

선택정렬() //sort
{
    for(파일 수만큼 파일 이름 배열 순회)
    {
        if(첫번째 파일 첫 글자가 숨김 파일이면)
        {
            filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
        }
        else(숨김파일이 아니면)
        {
            이름 저장
        }
        if(두번째 파일 첫 글자가 숨김 파일이면)
        {
            filename[i]의 길이만큼 반복하고 첫 글자 제외하고 저장
        }
        else(숨김파일이 아니면)
        {
            이름 저장
        }
        대문자로 변환
        if(S옵션을 선택한 경우)
        {
            경로가 지정되지 않은 경우 파일 이름 저장
            경로가 있으면 경로와 파일 이름을 합쳐 저장
        }
        파일 크기 비교
        else
        {
            대문자로 변환한 문자열 비교
        }
    }
    if(r 옵션을 선택한 경우)
    {
        파일 이름을 역순으로 정렬
    }
}

```

sort 함수의 pseudo code 이다.

```

파일 정보 출력() //print_file_info
{
    파일 정보 읽어 구조체 변수 filestat에 저장
    if(파일 경로가 있으면)
    {
        파일 경로와 파일 이름을 합치고 stat함수로 파일 정보 읽음
    }
    else
    {
        파일 이름으로 정보 읽음
    }
    파일 유형, 허가권, 링크 수 , 소유자, 그룹, 출력
    if(h옵션이 0이 아닌 경우 파일 크기를 사람이 읽을 수 있는 형태로 출력)
    else(파일 크기 출력)
    수정시간, 파일 경로 출력
}

```

print\_file\_info 함수의 pseudo code 이다.

```

인자로 전달된 파일 크기에 대한 단위를 계산하여 출력하는 함수 //print_human_readable_size
{
    단위 배열 선언
    단위 인덱스 선언 & 초기화
    size가 1024가 넘어가면 단위를 증가시키며 나눔
    소수점 첫번째 자리까지 출력함
}

```

print\_human\_readable\_size 함수의 pseudo code 이다.

```

파일 목록 출력() //list_files
{
    디렉토리 열기
    절대경로로 변환
    if(파일이면)
    {
        if(1옵션이 켜진경우)
        {
            directory path출력
            total출력
            파일 정보 출력
        }
        else
        {
            파일 이름 출력
        }
    }
    else
    {
        if(a 옵션이 꺼져있는 경우)
        {
            while(디렉토리 내 모든 파일 순회)
            {
                숨김파일이면 넘어감
                아니면 파일 개수 증가
            }
        }
        else
        {
            while(디렉토리 내 모든 파일 순회)
            {
                파일 이름 배열에 저장
            }
        }
        파일 이름 정렬
        if(1 옵션이 켜져있는 경우)
        {
            directory path, total, 파일 정보 출력
        }
        else
        {
            파일 이름 배열에 저장
        }
    }
    디렉토리 닫기
}

```

list\_files 함수의 pseudo code 이다.

```

파일 이름에 와일드 카드 문자가 있는지 검사하는 함수 //wildcardpattern
{
    if( 문자열에 '*' 또는 '?' 문자가 있는지 검사)
        있으면 1 반환
    문자열에 '[' 문자가 있는지 검사
        있으면 1 반환
    없으면 0 반환
}

```

wildcardpattern 함수의 pseudo code 이다.

```

main(argc, argv)
{
    명령줄 옵션 parsing
    {
        a 옵션일 때 aflag +1
        l 옵션일 때 lflag +1
        h 옵션일 때 hflag +1
        S 옵션일 때 Sflag +1
        r 옵션일 때 rflag +1
        알 수 없는 옵션일 때 에러메세지 출력
    }
    if(인자가 없을 때)
    {
        현재 디렉토리에 대한 파일 리스트 출력
    }
    else(인자가 있을 때)
    {
        if(와일드카드 패턴이 있으면)
        {
            glob함수를 이용해 와일드카드 패턴에 해당하는 파일 리스트 가져옴
            for(파일 리스트 반복)
                폴더이면 폴더 배열에 저장
                파일이면 파일 배열에 저장
        }
        glob함수에 할당된 메모리 해제
        else
        {
            파일 정보 얻는데 실패시 에러메세지 출력
            가져온 파일이 폴더이면 폴더 배열에 저장
            가져온 파일이 파일이면 파일 배열에 저장
        }
    }
    파일 이름 정렬
    디렉토리 이름 정렬
    flag에 따라 출력할 파일 정보 결정
}

```

main 함수의 pseudo code 이다.

#### 4. 결과화면

```

kw2021202058@ubuntu:~/work$ ls
2021202058_advanced_ls.c  advanced_ls  Makefile  tavl  text1.txt  tqwt
2021202058_final_ls.c    fnmatch     simple_ls  test  text4.txt  tyut
2021202058_simple_ls.c   fnmatch.c   spls_final test.c  text9.txt
kw2021202058@ubuntu:~/work$ ./spls_final
Directory path: /home/kw2021202058/work

2021202058_advanced_ls.c
2021202058_final_ls.c
2021202058_simple_ls.c
advanced_ls
fnmatch
fnmatch.c
Makefile
simple_ls
spls_final
tavl
test
test.c
text1.txt
text4.txt
text9.txt
tqwt
tyut

```

ls 명령어와 구현한 ls 명령어를 출력한 사진이다. ls 명령어를 사용했을 때와 동일하게 출력되며, 정렬 역시 잘 된 것을 확인할 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -a
.          2021202058_final_ls.c  .empty.txt  Makefile  tavl  text1.txt  tqwt
..         2021202058_simple_ls.c  fnmatch     simple_ls  test  text4.txt  tyut
2021202058_advanced_ls.c  advanced_ls   fnmatch.c   spls_final test.c  text9.txt
kw2021202058@ubuntu:~/work$ ./spls_final -a
Directory path: /home/kw2021202058/work

.
..
2021202058_advanced_ls.c
2021202058_final_ls.c
2021202058_simple_ls.c
advanced_ls
.empty.txt
fnmatch
fnmatch.c
Makefile
simple_ls
spls_final
tavl
test
test.c
text1.txt
text4.txt
text9.txt
tqwt
tyut

```

ls -a 옵션과 구현한 -a 옵션을 출력한 사진으로 ls -a 옵션을 사용했을 때와 동일하게 출력된다. 숨김 파일이 출력되며 정렬 역시 처음 '.'을 제외한 문자열이 동일한 조건으로 정렬되어 있는 것을 확인할 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -l
total 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final -l
Directory path: /home/kw2021202058/work
total: 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Tue Apr 11 20:29:31 2023 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Thu Apr 6 23:03:58 2023 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Thu Apr 6 23:03:55 2023 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Tue Apr 11 06:42:59 2023 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Tue Apr 11 20:30:02 2023 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Fri Mar 31 00:49:28 2023 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Fri Mar 31 00:50:29 2023 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tyut

```

ls -l 옵션과 구현한 -l 옵션을 출력한 사진으로 ls -l 옵션을 사용했을 때와 동일하게 출력된다. 파일 유형, 허가권, 링크 수, 소유자, 그룹, 파일 크기, 수정 시간 등 출력 포맷이 동일하며 정렬되어 출력된 것을 확인할 수 있다. 또한 directory path 와 total 을 출력하는 조건 역시 만족한 것을 확인할 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -al
total 132
drwxrwxr-x 2 kw2021202058 kw2021202058 4096 Apr 11 20:30 .
drwxr-xr-x 20 kw2021202058 kw2021202058 4096 Apr 11 19:59 ..
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 4 11:01 .empty.txt
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final -al
Directory path: /home/kw2021202058/work
total: 132
drwxrwxr-x 2 kw2021202058 kw2021202058 4096 Tue Apr 11 20:30:02 2023 .
drwxr-xr-x 20 kw2021202058 kw2021202058 4096 Tue Apr 11 19:59:08 2023 ..
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Tue Apr 11 20:29:31 2023 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 4 11:01:45 2023 .empty.txt
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Thu Apr 6 23:03:58 2023 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Thu Apr 6 23:03:55 2023 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Tue Apr 11 06:42:59 2023 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Tue Apr 11 20:30:02 2023 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Fri Mar 31 00:49:28 2023 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Fri Mar 31 00:50:29 2023 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tyut

```

ls -al 옵션과 구현한 -al 옵션을 출력한 사진으로 ls -al 옵션을 사용했을 때와 동일하게 출력된다. 파일 유형, 허가권, 링크 수, 소유자, 그룹, 파일 크기, 수정 시간 등 출력 포맷이 동일하며 숨김 파일 역시 출력되는 것을 볼 수 있다. 또한 정렬 역시 처음 ''을

제외한 문자열이 동일한 조건으로 정렬되어 출력된 것을 확인할 수 있으며, directory path 와 total 을 출력하는 조건 역시 만족한 것을 확인할 수 있다.

```
kw2021202058@ubuntu:~/work$ ./spls_final -al ..
Directory path: /home/kw2021202058
total: 152
drwxr-xr-x 20      kw2021202058 kw2021202058      4096      Tue Apr 11 19:59:08 2023 .
drwxr-xr-x 3       root root      4096      Fri Mar 31 00:05:09 2023 ..
-rw-r--r-- 1      kw2021202058 kw2021202058      3771      Fri Mar 31 00:05:09 2023 .bashrc
-rw-r--r-- 1      kw2021202058 kw2021202058     10759      Tue Apr 11 13:48:52 2023 .bash_history
-rw-r--r-- 1      kw2021202058 kw2021202058      220      Fri Mar 31 00:05:09 2023 .bash_logout
drwx----- 18      kw2021202058 kw2021202058      4096      Tue Apr 11 20:17:22 2023 .cache
drwx----- 18      kw2021202058 kw2021202058      4096      Tue Apr 11 19:55:53 2023 .config
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Wed Apr 5 01:25:49 2023 Desktop
-rw-r--r-- 1      kw2021202058 kw2021202058       25      Fri Mar 31 00:11:09 2023 .dmrc
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Documents
drwxrwxr-x 3       kw2021202058 kw2021202058      4096      Tue Apr 11 12:08:29 2023 .dotnet
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Downloads
-rw-r--r-- 1      kw2021202058 kw2021202058     8980      Fri Mar 31 00:05:09 2023 examples.desktop
drwx----- 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:42 2023 .gconf
drwx----- 3       kw2021202058 kw2021202058      4096      Tue Apr 11 19:59:02 2023 .gnupg
-rw-r--r-- 1      kw2021202058 kw2021202058     8268      Tue Apr 11 19:59:08 2023 .ICEauthority
drwx----- 3       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 .local
drwx----- 5       kw2021202058 kw2021202058      4096      Tue Apr 11 12:05:56 2023 .mozilla
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Music
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Pictures
drwx----- 3       kw2021202058 kw2021202058      4096      Tue Apr 11 12:07:55 2023 .pkg
-rw-r--r-- 1      kw2021202058 kw2021202058      655      Fri Mar 31 00:05:09 2023 .profile
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Public
-rw-r--r-- 1      kw2021202058 kw2021202058       0      Fri Mar 31 00:18:41 2023 .sudo_as_admin_successful
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Templates
drwxr-xr-x 2       kw2021202058 kw2021202058      4096      Fri Mar 31 00:11:10 2023 Videos
-rw-r--r-- 1      kw2021202058 kw2021202058     4866      Tue Apr 11 06:42:59 2023 .viminfo
drwxrwxr-x 3       kw2021202058 kw2021202058      4096      Tue Apr 11 12:07:55 2023 .vscode
drwxrwxr-x 2       kw2021202058 kw2021202058      4096      Tue Apr 11 20:30:02 2023 work
-rw-r--r-- 1      kw2021202058 kw2021202058       51      Tue Apr 11 19:58:59 2023 .xauthority
-rw-r--r-- 1      kw2021202058 kw2021202058       82      Tue Apr 11 19:58:59 2023 .xsession-errors
-rw-r--r-- 1      kw2021202058 kw2021202058     1245      Tue Apr 11 19:56:53 2023 .xsession-errors.old
kw2021202058@ubuntu:~/work$
```

지난 과제에서 출력되지 않은 부분을 찾아 수정하였다. 경로와 파일 이름을 합치는 과정을 추가해주어 오류를 해결하였고, 이로 인해 이전 directory 가 home 일 때 결과가 알맞게 출력되는 것을 볼 수 있다.

```
kw2021202058@ubuntu:~/work$ ls -lh
total 124K
-rw-rw-r-- 1 kw2021202058 kw2021202058 14K Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18K Apr 11 20:29 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3.7K Apr 3 20:57 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 14K Apr 11 06:29 advanced_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8.6K Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9.0K Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18K Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8.7K Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final -lh
Directory path: /home/kw2021202058/work
total: 124
-rw-rw-r-- 1      kw2021202058 kw2021202058      13.0K      Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rw-rw-r-- 1      kw2021202058 kw2021202058      18.0K      Tue Apr 11 20:29:31 2023 2021202058_final_ls.c
-rwxrwxr-x 1      kw2021202058 kw2021202058       3.7K      Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rwxrwxr-x 1      kw2021202058 kw2021202058      13.3K      Tue Apr 11 06:29:55 2023 advanced_ls
-rwxrwxr-x 1      kw2021202058 kw2021202058       8.5K      Thu Apr 6 23:03:58 2023 fnmatch
-rw-rw-r-- 1      kw2021202058 kw2021202058      413.0      Thu Apr 6 23:03:55 2023 fnmatch.c
-rw-rw-r-- 1      kw2021202058 kw2021202058       75.0      Tue Apr 11 06:42:59 2023 Makefile
-rwxrwxr-x 1      kw2021202058 kw2021202058       8.9K      Mon Apr 3 20:57:25 2023 simple_ls
-rwxrwxr-x 1      kw2021202058 kw2021202058      17.7K      Tue Apr 11 20:30:02 2023 spls_final
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:46:52 2023 tavg
-rwxrwxr-x 1      kw2021202058 kw2021202058       8.6K      Fri Mar 31 00:49:28 2023 test
-rw-rw-r-- 1      kw2021202058 kw2021202058      531.0      Fri Mar 31 00:50:29 2023 test.c
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1      kw2021202058 kw2021202058       0.0      Tue Apr 11 13:46:52 2023 tyut
```

새롭게 구현한 ls -h 옵션의 예시이며 구현한 -h 옵션을 -l 옵션과 함께 사용한 결과를 출력한 사진으로 사람이 읽을 수 있게 파일의 크기를 출력해주는 것을 볼 수 있다. 하지만 숫자가 미세하게 다른 점을 확인해 볼 수 있는데, 이는 첫번째 자리에서 소수점으로 나타내 주었기 때문에 조금의 오차가 발생할 수 있다는 점을 고려하면 알맞게 출력된 것을 확인할 수 있다.



```

kw2021202058@ubuntu:~/work$ ls -ls
total 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rwxrwx-rw- 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final -ls
Directory path: /home/kw2021202058/work
total: 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Tue Apr 11 20:29:31 2023 2021202058_final_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Tue Apr 11 20:30:02 2023 spls_final
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Fri Mar 31 00:49:28 2023 test
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Thu Apr 6 23:03:58 2023 fnmatch
-rwxrwx-rw- 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Fri Mar 31 00:50:29 2023 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Thu Apr 6 23:03:55 2023 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Tue Apr 11 06:42:59 2023 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tavg
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tyut

```

새롭게 구현한 ls -S 옵션의 예시이며 구현한 -S 옵션을 -l 옵션과 함께 사용한 결과를 출력한 사진으로 file size 의 크기를 기준으로 큰 것부터 정렬한 것을 확인할 수 있다. 파일의 크기가 같을 경우 알파벳 순서대로 출력하여 주었으며, ls 명령어의 결과와 일치하며 잘 출력된 것을 확인할 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -lr
total 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rwxrwx-rw- 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
kw2021202058@ubuntu:~/work$ ./spls_final -lr
Directory path: /home/kw2021202058/work
total: 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tyut
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Fri Mar 31 00:50:29 2023 test.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Fri Mar 31 00:49:28 2023 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Tue Apr 11 20:30:02 2023 spls_final
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Tue Apr 11 06:42:59 2023 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Thu Apr 6 23:03:55 2023 fnmatch.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Thu Apr 6 23:03:58 2023 fnmatch
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rwxrwx-rw- 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Tue Apr 11 20:29:31 2023 2021202058_final_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c

```

새롭게 구현한 ls -r 옵션의 예시이며 구현한 -r 옵션을 -l 옵션과 함께 사용한 결과를 출력한 사진으로 문자열을 역순으로 정렬하여 출력한 것을 확인할 수 있다. ls 명령어의 결과와 일치하며 잘 출력된 것을 확인할 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -larS
total 132
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 4 11:01 .empty.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
drwxr-xr-x 20 kw2021202058 kw2021202058 4096 Apr 11 19:59 ..
drwxrwxr-x 2 kw2021202058 kw2021202058 4096 Apr 11 20:30 .
-rw-rw-r-- 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
kw2021202058@ubuntu:~/work$ ./spls_final -larS
Directory path: /home/kw2021202058/work
total: 132
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tyut
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:47:40 2023 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 11 13:46:52 2023 tavg
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Tue Apr 4 11:01:45 2023 .empty.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Tue Apr 11 06:42:59 2023 Makefile
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Thu Apr 6 23:03:55 2023 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Fri Mar 31 00:50:29 2023 test.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Mon Apr 3 20:57:05 2023 2021202058_simple_ls.c
drwxr-xr-x 20 kw2021202058 kw2021202058 4096 Tue Apr 11 19:59:08 2023 ..
drwxrwxr-x 2 kw2021202058 kw2021202058 4096 Tue Apr 11 20:30:02 2023 .
-rw-rw-r-- 1 kw2021202058 kw2021202058 8712 Thu Apr 6 23:03:58 2023 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 8840 Fri Mar 31 00:49:28 2023 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 9120 Mon Apr 3 20:57:25 2023 simple_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Tue Apr 11 06:27:20 2023 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 13664 Tue Apr 11 06:29:55 2023 advanced_ls
-rw-rw-r-- 1 kw2021202058 kw2021202058 18104 Tue Apr 11 20:30:02 2023 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Tue Apr 11 20:29:31 2023 2021202058_final_ls.c

```

r 옵션과 S 옵션을 같이 사용한 경우 파일의 크기가 같으면 문자열을 역순으로 정렬하는 것을 확인할 수 있으므로 결과가 잘 나오는 것을 알 수 있다.

```

kw2021202058@ubuntu:~/work$ ls -l
total 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rwxrwxrwx 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavg
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final 'text[1-9].txt'
text1.txt
text4.txt
text9.txt

```

[1-9]까지의 있는 문자 중 하나라도 있으면 출력해야 하므로 text1.txt, text4.txt, text9.txt 를 출력하는 것을 확인할 수 있으므로, 결과가 잘 나오는 것을 알 수 있다.

```
kw2021202058@ubuntu:~/work$ ls -l
total 124
-rw-rw-r-- 1 kw2021202058 kw2021202058 13354 Apr 11 06:27 2021202058_advanced_ls.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 18427 Apr 11 20:29 2021202058_final_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 3788 Apr 3 20:57 2021202058_simple_ls.c
-rwxrwxr-x 1 kw2021202058 kw2021202058 13664 Apr 11 06:29 advanced_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 8712 Apr 6 23:03 fnmatch
-rw-rw-r-- 1 kw2021202058 kw2021202058 413 Apr 6 23:03 fnmatch.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 75 Apr 11 06:42 Makefile
-rwxrwxr-x 1 kw2021202058 kw2021202058 9120 Apr 3 20:57 simple_ls
-rwxrwxr-x 1 kw2021202058 kw2021202058 18104 Apr 11 20:30 spls_final
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tavl
-rwxrwxr-x 1 kw2021202058 kw2021202058 8840 Mar 31 00:49 test
-rw-rw-r-- 1 kw2021202058 kw2021202058 531 Mar 31 00:50 test.c
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text1.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text4.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:47 text9.txt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tqwt
-rw-rw-r-- 1 kw2021202058 kw2021202058 0 Apr 11 13:46 tyut
kw2021202058@ubuntu:~/work$ ./spls_final 't??t'
tavl
test
tqwt
tyut
```

'?'는 하나의 문자를 나타내므로, 임의의문자로 대체될 수 있다. t 로 시작하고 t로 끝나며 길이가 4 인 문자열을 출력해야 하므로, tavl, test, tqwt, tyup 를 출력해야한다. 따라서 알맞은 결과가 나온 것을 알 수 있다.

```
kw2021202058@ubuntu:~/work$ ./spls_final ~ '*'
2021202058_advanced_ls.c
2021202058_final_ls.c
2021202058_simple_ls.c
advanced_ls
fnmatch
fnmatch.c
Makefile
simple_ls
spls_final
tavl
test
test.c
text1.txt
text4.txt
text9.txt
tqwt
tyut
Directory path: /home/kw2021202058

Desktop
Documents
Downloads
examples.desktop
Music
Pictures
Public
Templates
Videos
work
```

'\*'는 임의의 문자열을 나타내므로, 현재 사용자의 홈 디렉터리에 있는 모든 파일과 폴더에 대한 와일드카드 매칭을 의미하므로, 알맞게 출력된 것을 알 수 있다.

```

kw2021202058@ubuntu:~/work$ ./spls_final '/home/kw2021202058/*'
/home/kw2021202058/examples.desktop
Directory path: /home/kw2021202058/Desktop

Directory path: /home/kw2021202058/Documents

Directory path: /home/kw2021202058/Downloads

Directory path: /home/kw2021202058/Music

Directory path: /home/kw2021202058/Pictures

Directory path: /home/kw2021202058/Public

Directory path: /home/kw2021202058/Templates

Directory path: /home/kw2021202058/Videos

Directory path: /home/kw2021202058/work

2021202058_advanced_ls.c
2021202058_final_ls.c
2021202058_simple_ls.c
advanced_ls
fnmatch
fnmatch.c
Makefile
simple_ls
spls_final
tavg
test
test.c
text1.txt
text4.txt
text9.txt
tqwt
tyut

```

위의 사진은 home 디렉토리 아래의 kw2021202058 디렉토리에 있는 모든 파일과 폴더에 대한 와일드카드 매칭을 의미한다. 따라서 home 디렉토리 아래의 폴더인 kw2021202058 에 있는 모든 파일과 폴더를 출력해주었으므로, 결과가 알맞게 출력된 것을 알 수 있다.

## 5. 고찰

1 차과제와 2 차과제를 바탕으로 이번 과제를 진행했기 때문에, 코드를 새롭게 구현하는 점 보다는, 기존의 코드를 수정하는 과정이 더 오래 걸리고 어려웠던 것 같다. 우선 대소문자 구분 없이 정렬하는 것은 동일하지만, 특수문자가 들어간 경우 대문자를 기준으로 정렬해야 했다. 아스키 코드표를 기준으로 대문자와 소문자 사이에 있는 특수문자, 예를 들어 '\_' 같은 경우 소문자보다 뒤에 나왔기 때문에 기존의 코드를 수정해 주어야 했다. 기존의 코드는 strcasecmp 함수를 사용하였지만 이 함수를

사용할 경우 대문자를 소문자 기준으로 생각하기 때문에, toupper 함수를 사용하여 대문자로 먼저 다 바꿔준 후 strcmp 함수를 사용하여 비교해주었다. 여기서 toupper 함수는 매개변수로 받은 문자를 대문자로 변환한 값을 반환하는 함수이다. 두 번째로 기존의 코드는 option 을 받아 옵션이 없을 때 0, a 옵션일 때 1 이런 식으로 구현하였다. 하지만 이렇게 할 경우 경우의수가 늘어날 뿐만 아니라, 다음 과제에서 또다른 옵션이 나온다면 옵션의 수가 더 늘어나게 되어 비효율적이다. 따라서 flag 에 따라 구현하도록 수정해주었다.

## 6. Reference

시스템프로그래밍실습 강의자료