

2019. 4. 9.

Assignment # 2-2

금요일 - 이성원 교수님

2015722087 컴퓨터정보공학부

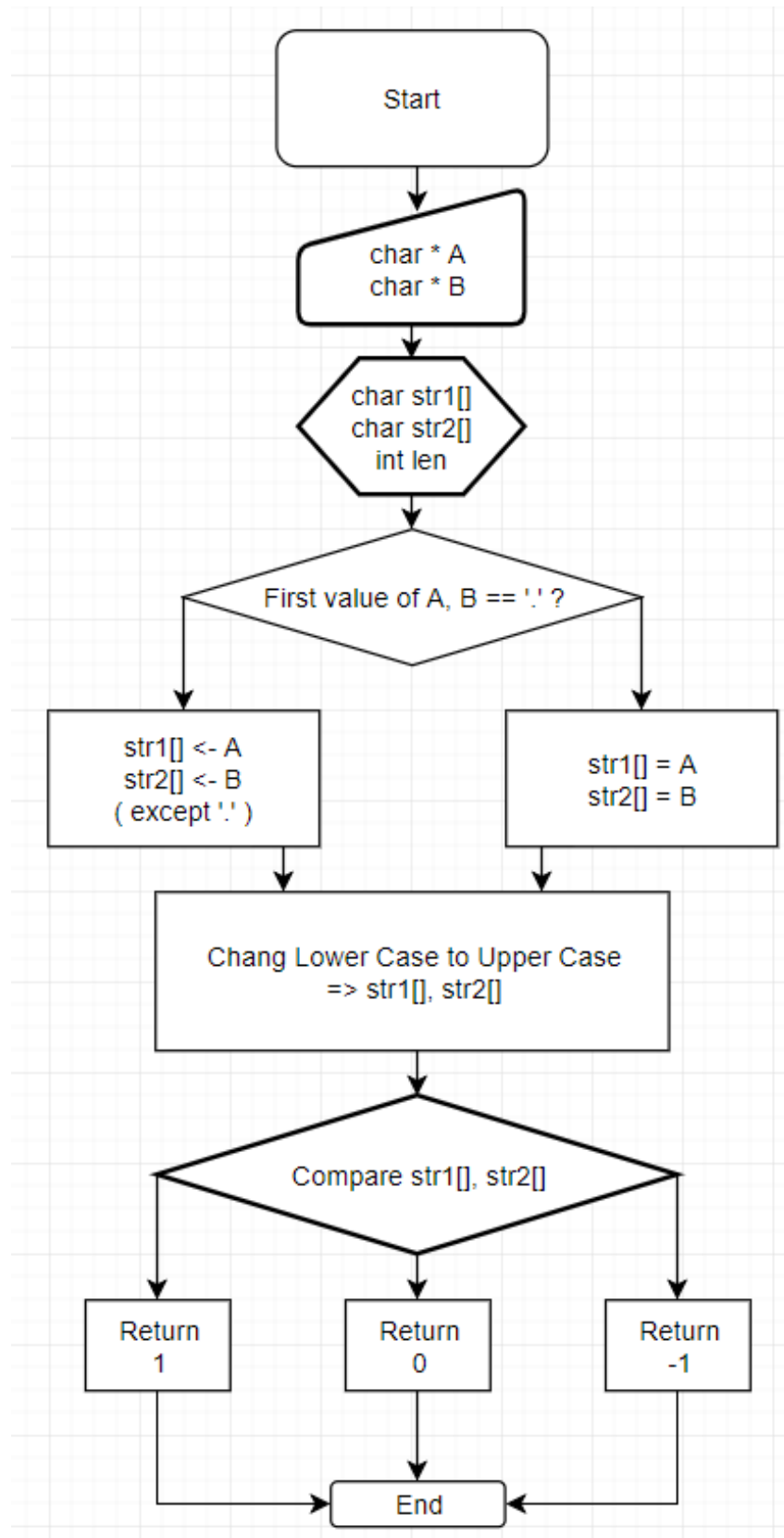
김민철

Introduction

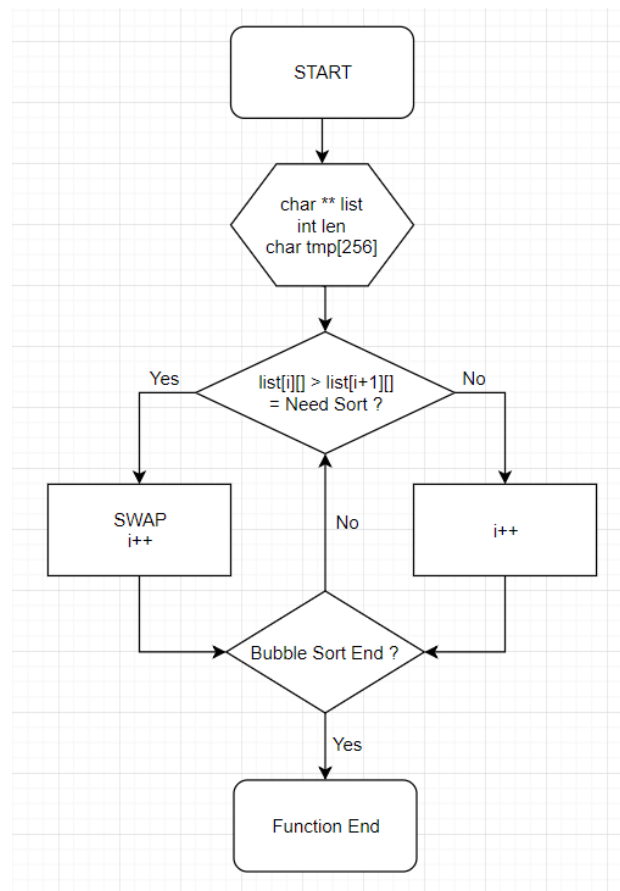
이번 과제는 2-1 에서 구현 하였던 simple_ls 보다 더 많은 기능이 있는 advanced_ls 를 구현하는 것이다. 기존의 simple_ls 에서 옵션 -a, -l, -la 가 추가되어 옵션에 따른 작업을 하고, 탐색한 폴더의 경로를 출력하고, 파일들의 모든 size 를 합한 total size 를 출력해주는 기능이 추가된다. 숨김 파일은 맨 앞의 '.'을 제외한 문자열을 기준으로 정렬하고, 폴더나 파일 이름 입력에는 개수 제한이 없다.

Flow Chart

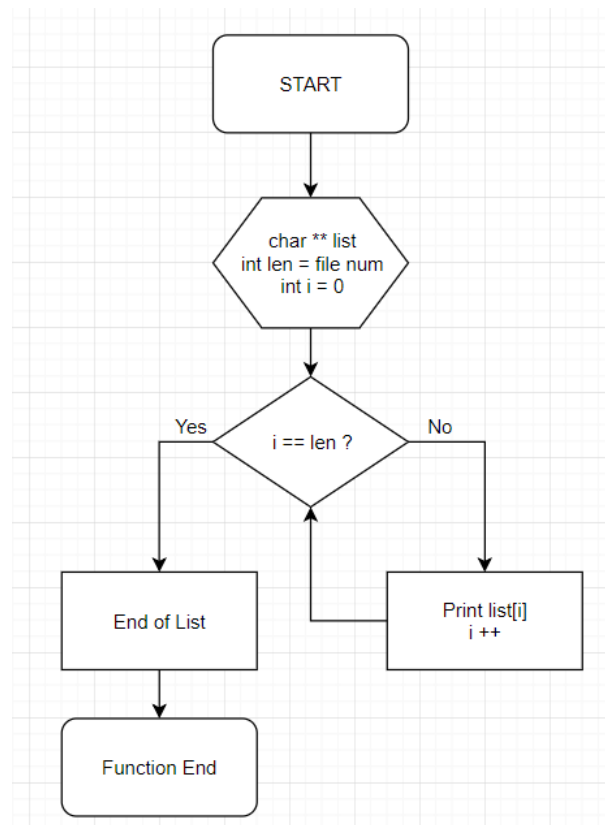
-int strcmp_i



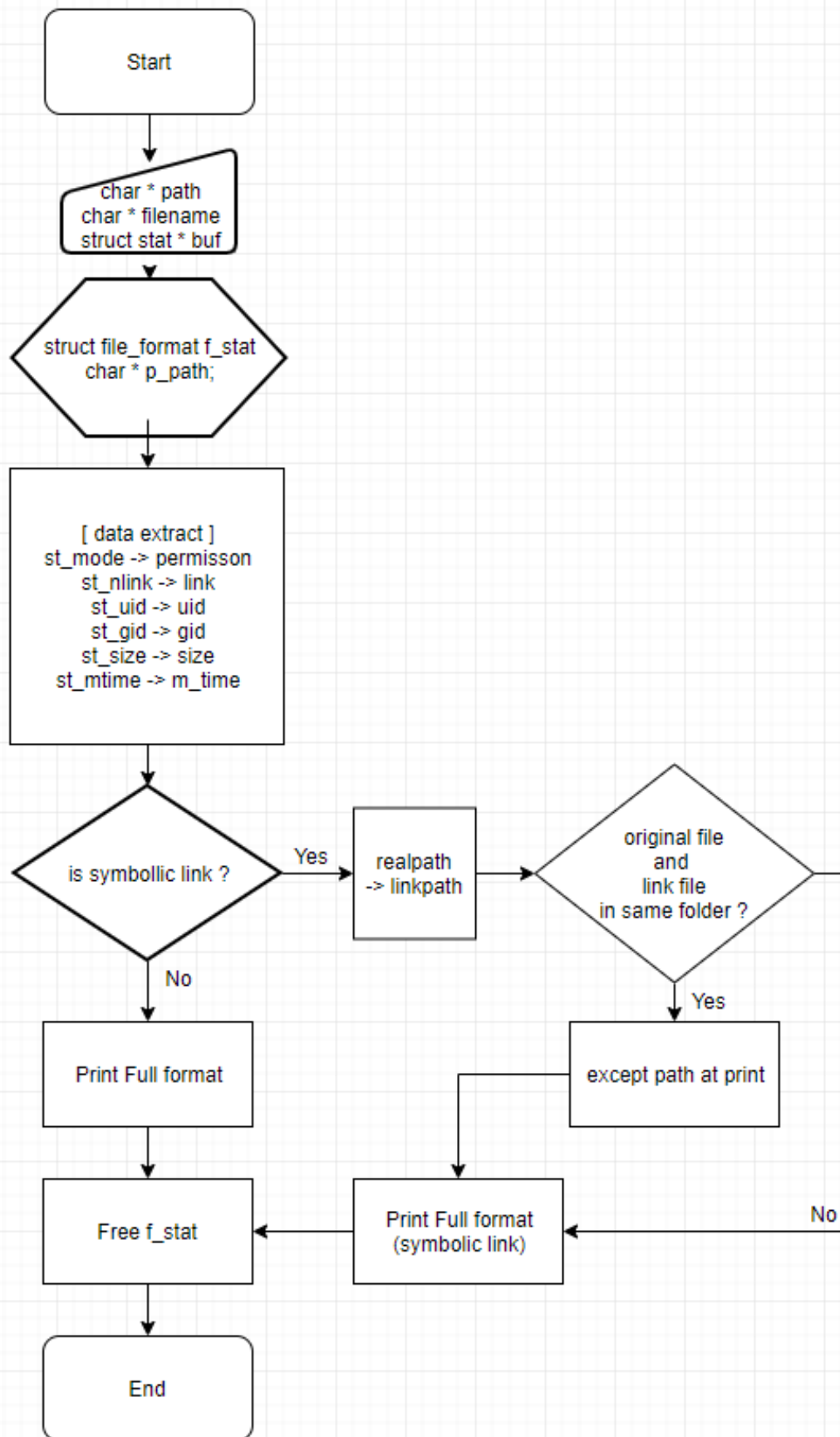
-void sort_list



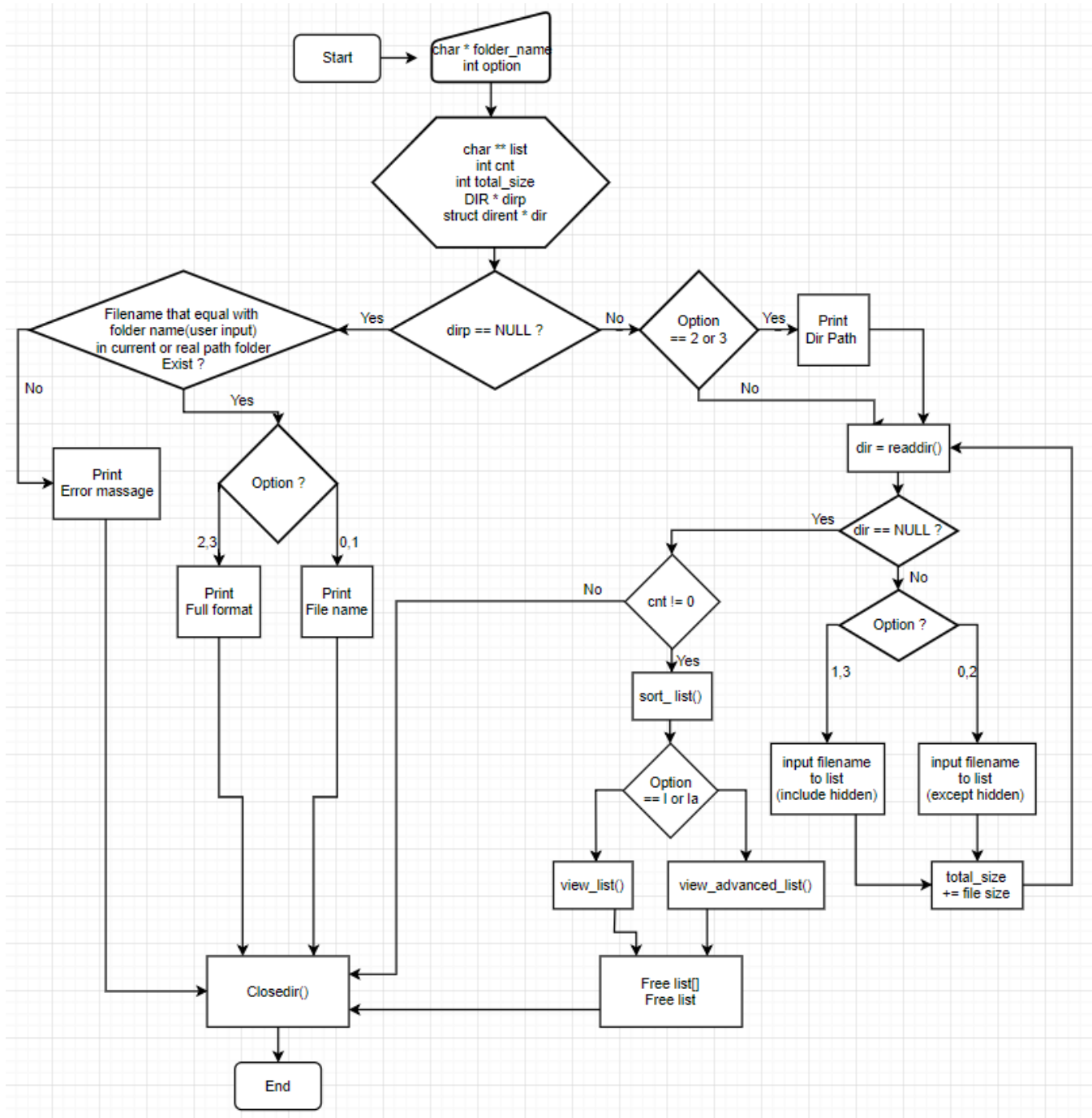
-void view_list



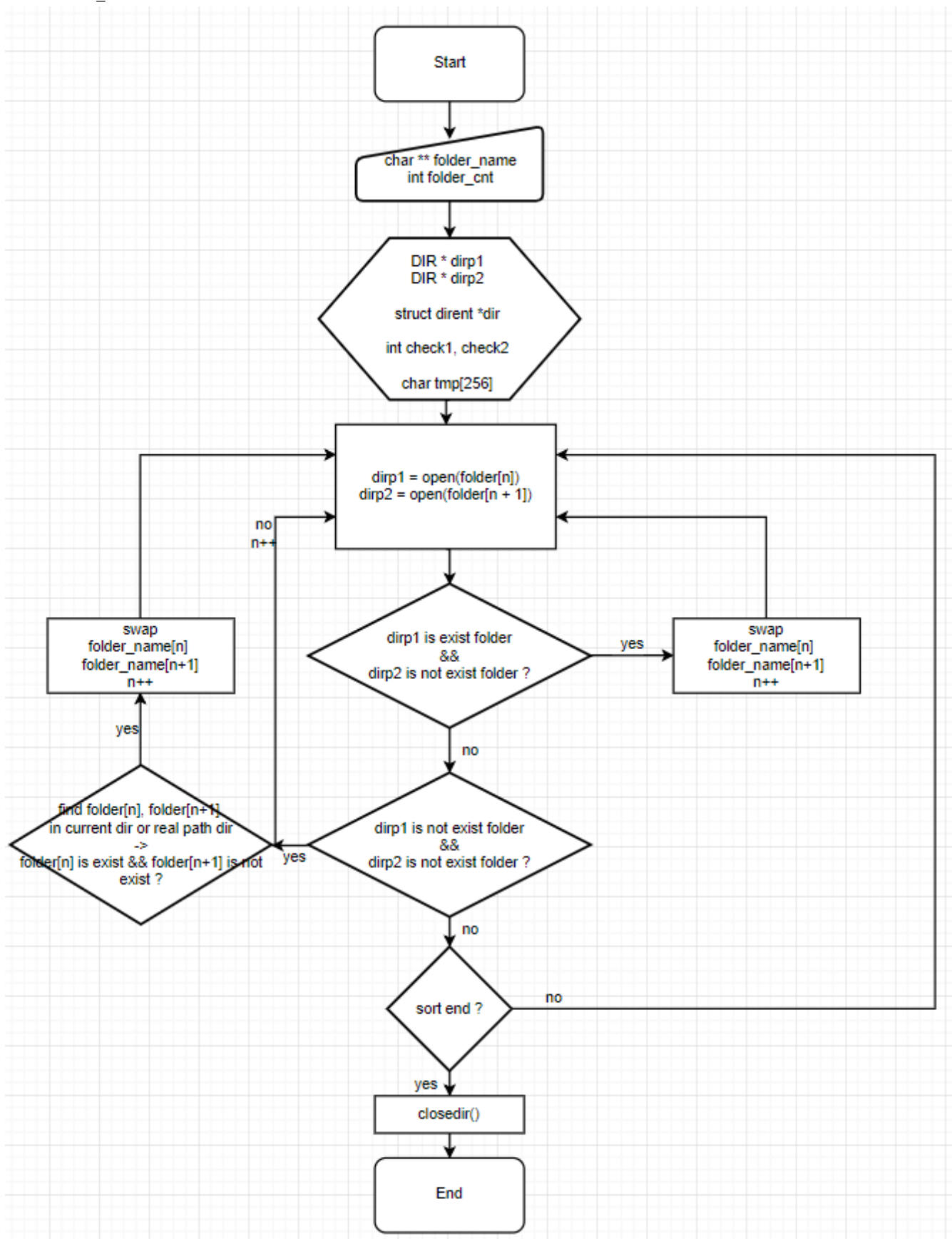
-void view_advanced_list



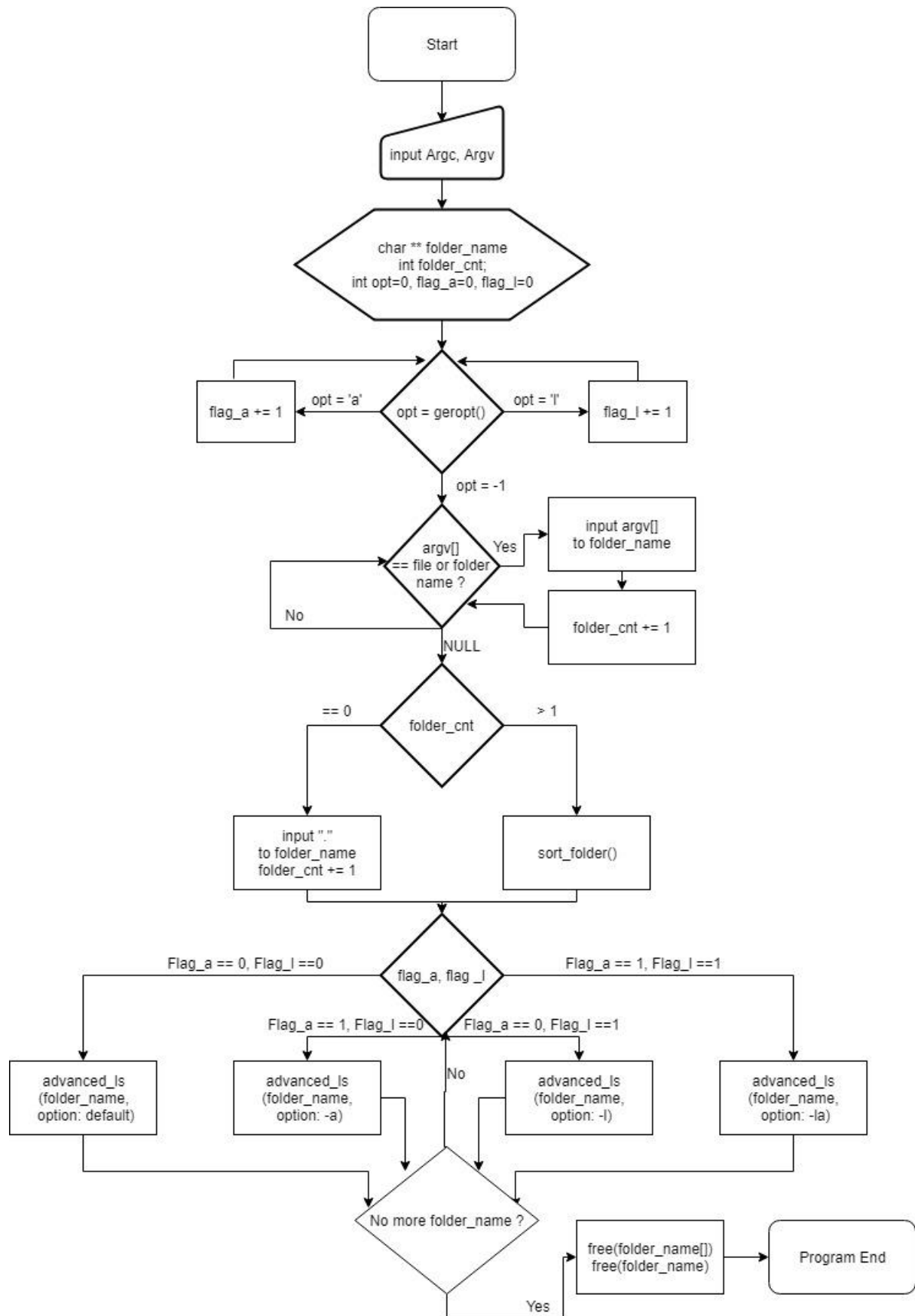
-void advanced_ls



-void sort_folder



-int main



Pseudo code

int strcmp_i (char * 비교할 첫 번째 문자열 A, char * 비교할 두 번째 문자열 B)

```
{  
    <예외 처리>  
    A 랑 B 가 ".", ".." 이면 "." 이 뒤로,  
    둘 중 하나만 "." 이면 "."이 앞으로  
    둘 중 하나만 ".." 이면 ".."이 앞으로
```

```
    if( A 가 B 보다 길면 )  
        len = B 의 길이;  
    else  
        len = A 의 길이
```

```
    for(len 만큼 반복)  
    {  
        if(소문자라면)  
            대문자로 바꾼다  
    }  
    if( A > B )        return = 1;  
    else if( B > A )    return = -1;  
    else                return = 0;
```

```
}
```

void sort_list(char ** li, int len)

```
{  
    for (저장된 파일이름 개수-1 만큼 반복)  
    {  
        if (더 이상 내용이 없다)  
            break;  
  
        for (저장된 파일이름 개수-1 만큼 반복)  
        {  
            if ( 앞에 저장된 파일이름 > 뒤에 저장된 파일 이름)  
                두 파일 이름을 swap
```

```

    }
};
}

```

```

void view_list(char ** li, int len)
{
    for (파일 이름 개수 만큼 반복)
    {
        for (파일 이름 다 출력할 때까지 반복)
        {
            파일 이름의 알파벳 하나씩 출력
        }
    }
}

```

```

void view_advanced_list(char * 파일 경로, char * 파일 이름, struct stat * 파일 정보)
{
    St_mode에 저장되어 있는 파일 타입에 따라 permission[0] 결정
    St_mode에 저장되어 있는 user 권한에 따라 permission[1~3] 결정
    St_mode에 저장되어 있는 group 권한에 따라 permission[4~6] 결정
    St_mode에 저장되어 있는 other 권한에 따라 permission[7~9] 결정
    ➔ Permission = "- --- --- ---"

    st_nlink -> f_stat->nlink
    st_uid -> f_stat->uid
    st_gid -> f_stat->gid
    st_size -> f_stat->size
    st_mtime -> f_stat->m_time

    // Full Format 출력
    if 파일 type 이 symbolic link 라면
    {
        Symbolic link 파일과 가리키는 경로 파일이 같은 폴더에 있다면 -> 가리키는 파일명만 출력
        Symbolic link 파일과 가리키는 경로 파일이 다른 폴더에 있다면 -> 절대 경로 모두 출력
    }
    else 파일 type 이 symbolic link 가 아니라면
        Full Format 출력시 파일 이름만 출력
}

```

```

void advanced_ls(char * 입력한 폴더, int 옵션)
{
    if 폴더가 아니라면

```

```

{
    입력된 폴더 이름에 경로가 따로 있는지 확인
    ⇒ 경로가 있을 경우 해당 경로 opendir()
    ⇒ 경로가 없을 경우 현재 디렉토리 opendir()
    while 디렉토리 탐색
    {
        if 파일을 찾았다면
        {
            If 옵션이 a 나 default 라면
                파일 이름만 출력
            else if 옵션이 l 이나 la 라면
                Full Format 출력

            함수 종료
        }
    }
    파일을 못 찾았다면 에러 메시지 출력
}
else // is folder
{
    if 옵션이 l 이나 la 라면
        작업중인 디렉토리 출력

    while 입력된 폴더 탐색
    {
        if 파일이 없다면
            반복문 탈출

        if 히든 파일이고 옵션이 -l 이나 default 라면
        {
            List에 파일 이름 입력
            Total size += 파일 size
        }
        else if 옵션이 -a 이나 -la 라면 모든 파일 입력
        {
            List에 파일 이름 입력
            Total size += 파일 size
        }
    }
}

if 파일이 존재한다면
{
    List 정렬.

    if 옵션이 -a 거나 default 라면
        모든 파일 이름 출력
    else if 옵션이 -l 이거나 -la 라면
    {
        토탈 사이즈 출력

        for (파일 개수만큼 반복)
        {
            Full Format 으로 파일 이름 출력

```

```

    }
}

파일 리스트 동적 할당 해제

}
폴더 닫기
}

```

```

void sort_folder(char ** 폴더 list, int 폴더 개수)
{
    int check1;
    int check2;

    for (폴더 개수 - 1 만큼 반복)
    {
        for (폴더 개수 - 1 만큼 반복)
        {
            디렉토리, 체크 변수, 카운트 변수, copy할 변수 초기화.

            리스트에 연속된 폴더 이름 2개로 폴더 open : A폴더(앞), B폴더(뒤)

            if A폴더는 존재하고 B폴더는 존재하는 폴더가 아닌 경우
                폴더 리스트에서 swap
            else if 둘 다 존재하는 폴더가 아닌 경우
            {
                A,B 각각 입력된 폴더 이름에 경로가 따로 있는지 확인
                => 경로가 있을 경우 해당 경로 opendir()
                => 경로가 없을 경우 현재 디렉토리 opendir()

                while A 탐색
                {
                    if A폴더 이름의 파일이 존재하는 경우
                        check1 = 1;
                }
                while B 탐색
                {
                    if B폴더 이름의 파일이 존재하는 경우
                        check2 = 1;
                }

                if A폴더 이름의 파일만 존재하는 경우 => B폴더 이름은 에러
                    폴더 리스트에서 swap
            }
        }
    }

    폴더 닫기
}

```

```

int main(int 입력한 인자 개수, char * 입력한 인자 문자열)
{
    while 옵션 구하기
    {
        switch (옵션)
        {
            case 'a': // include option a : print all file (include hidden)
                a 옵션이 있다면 a 옵션 flag ++
                break;
            case 'l': // include option l : print all format of file
                l 옵션이 있다면 l 옵션 flag ++
                break;
        }
    }

    for(입력한 인자 개수 - 1 만큼 반복 (첫 번째 인자는 파일 실행이므로 제외)
    {
        // find folder name. (or file)
        if (옵션이 아니고 폴더 이름이라면, (".", ".." 포함) )
        {
            폴더 리스트에 폴더 이름 추가
            폴더 개수 카운트 ++
        }
    }

    if 폴더 이름이 없다면 -> default = 현재 디렉토리
    {
        폴더 리스트에 현재 디렉토리 추가
        폴더 개수 카운트 ++
    }
    else 폴더 개수가 2개 이상이라면
        sort_folder()

    for (폴더 개수 만큼 반복)// Print folders !
    {
        현재 작업중인 디렉토리로 초기화 chdir(origin_wd)
        if option : a
            advanced_ls(folder_name[i], 1); 출력
        else if option : l
            advanced_ls(folder_name[i], 2); 출력
        else if option : al, la
            advanced_ls(folder_name[i], 3); 출력
        else if option : default
            advanced_ls(folder_name[i], 0); 출력
    }

    폴더 리스트 동적 할당 해제

    프로그램 종료
}

```

Result

```
sp2015722087@ubuntu:~/sp/shared/2-2$ ls
a advanced_ls advanced_ls_divide.c b Makefile spls.c
sp2015722087@ubuntu:~/sp/shared/2-2$ ./advanced_ls
a
advanced_ls
advanced_ls_divide.c
b
Makefile
spls.c
```

```
sp2015722087@ubuntu:~/sp/shared/2-2$ ./advanced_ls -la
Directory path: /mnt/hgfs/sh/2-2
total : 72
drwxrwxrwx    1    root    root    4096    Apr  9 20:36    .
drwxrwxrwx    1    root    root    4096    Apr  9 05:05    ..
-rwxrwxrwx    1    root    root     0    Apr  6 16:05    a
-rwxrwxrwx    1    root    root   24216    Apr  9 20:36    advanced_ls
-rwxrwxrwx    1    root    root   21069    Apr  5 15:16    advanced_ls_divide.c
-rwxrwxrwx    1    root    root     0    Apr  6 23:52    b
-rwxrwxrwx    1    root    root     51    Apr  9 00:21    Makefile
-rwxrwxrwx    1    root    root   19825    Apr  9 20:33    spls.c
```

```
sp2015722087@ubuntu:~/sp/shared/2-2$ ./advanced_ls -la
Directory path: /home/sp2015722087
total : 114524
drwxr-xr-x    18    sp2015722087    sp2015722087    4096    Apr  9 18:14    .
drwxr-xr-x     3    root    root    4096    Mar 15 16:02    ..
-rw-r--r--     1    sp2015722087    sp2015722087     0    Apr  5 16:02    1
-rw-r--r--     1    sp2015722087    sp2015722087   3771    Mar 15 16:02    .bashrc
-rw-----     1    sp2015722087    sp2015722087  19626    Apr  9 08:23    .bash_history
-rw-r--r--     1    sp2015722087    sp2015722087    220    Mar 15 16:02    .bash_logout
drwx-----    15    sp2015722087    sp2015722087    4096    Apr  5 19:45    .cache
drwx-----     3    sp2015722087    sp2015722087    4096    Apr  5 19:55    .compiz
drwx-----    15    sp2015722087    sp2015722087    4096    Mar 15 17:07    .config
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Desktop
-rw-r--r--     1    sp2015722087    sp2015722087     25    Mar 15 16:23    .dmrc
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Documents
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Downloads
-rw-r--r--     1    sp2015722087    sp2015722087   8980    Mar 15 16:02    examples.desktop
-rw-rw-r--     1    sp2015722087    sp2015722087     6    Mar 15 17:32    file.txt
drwx-----     2    sp2015722087    sp2015722087    4096    Mar 29 17:16    .gconf
drwx-----     3    sp2015722087    sp2015722087    4096    Apr  9 18:14    .gnupg
-rw-----     1    sp2015722087    sp2015722087    4096    Mar 15 17:56    .hello.swp
-rw-----     1    sp2015722087    sp2015722087   6996    Apr  9 18:14    .ICEauthority
-rw-r--r--     1    sp2015722087    sp2015722087  54487776    Mar 15 17:32    initrd.img
-rw-r--r--     1    sp2015722087    sp2015722087  54487776    Mar 15 17:32    initrd.img.old
drwx-----     3    sp2015722087    sp2015722087    4096    Mar 15 16:23    .local
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Music
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Pictures
-rw-r--r--     1    sp2015722087    sp2015722087    655    Mar 15 16:02    .profile
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Public
drwxrwxr-x     4    sp2015722087    sp2015722087    4096    Apr  9 00:19    sp
-rw-r--r--     1    sp2015722087    sp2015722087     0    Mar 15 16:25    .sudo_as_admin_successful
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Templates
drwxrwxr-x     5    sp2015722087    sp2015722087    4096    Mar 29 17:13    test
drwxr-xr-x     2    sp2015722087    sp2015722087    4096    Mar 15 16:23    Videos
-rw-----     1    sp2015722087    sp2015722087   5549    Apr  9 07:27    .viminfo
-rw-r--r--     1    sp2015722087    sp2015722087  8128088    Mar 15 17:32    vmlinuz
-rw-----     1    sp2015722087    sp2015722087     51    Apr  9 18:14    .xauthority
-rw-rw-r--     1    sp2015722087    sp2015722087    131    Mar 15 16:27    .xinputrc
-rw-----     1    sp2015722087    sp2015722087     82    Apr  9 18:14    .xsession-errors
-rw-----     1    sp2015722087    sp2015722087   1162    Apr  9 08:23    .xsession-errors.old
```

저번 과제에서 옵션을 추가하기 위해 getopt() 함수를 이용하여 구현하였다. -a, -l, -la 를 구별하기 위해 int 형 변수 flag_a, flag_l 을 선언하여 -a 옵션이 있으면 flag_a 를, -l 옵션이 있으면 flag_l 을 1 로 바꿔 표시하였다. 또 저번 과제에서는 폴더 입력을 하나만 받을 수 있었지만, 이번에는 제한 없이 입력 받게 해야 하기 때문에 입력한 폴더명들을 먼저 배열로 받아와서 폴더 이름을 정렬 후 순서대로 폴더를 읽고 해당 폴더의 파일을 출력하는 방식으로 구현하였다.

```

sp2015722087@ubuntu:~/sp/shared/2-2$ ./advanced_ls /home /var -la
Directory path: /home
total : 12
drwxr-xr-x 3 root root 4096 Mar 15 16:02 .
drwxr-xr-x 24 root root 4096 Apr 3 01:33 ..
drwxr-xr-x 18 sp2015722087 sp2015722087 4096 Apr 9 18:14 sp2015722087
Directory path: /var
total : 56
drwxr-xr-x 14 root root 4096 Feb 27 09:06 .
drwxr-xr-x 24 root root 4096 Apr 3 01:33 ..
drwxr-xr-x 2 root root 4096 Apr 9 00:22 backups
drwxr-xr-x 16 root root 4096 Mar 15 16:03 cache
drwxrwxrwx 2 root whoopsie 4096 Apr 9 04:56 crash
drwxr-xr-x 66 root root 4096 Mar 15 16:23 lib
drwxrwxr-x 2 root staff 4096 Apr 13 05:14 local
lrwxrwxrwx 1 root root 9 Mar 15 15:52 lock -> /run/lock
drwxrwxr-x 13 root syslog 4096 Apr 9 18:11 log
drwxrwxr-x 2 root mail 4096 Feb 27 08:57 mail
drwxrwxrwx 2 root whoopsie 4096 Feb 27 09:02 metrics
drwxr-xr-x 2 root root 4096 Feb 27 08:57 opt
lrwxrwxrwx 1 root root 4 Mar 15 15:52 run -> /run
drwxr-xr-x 2 root root 4096 Jan 30 08:27 snap
drwxr-xr-x 7 root root 4096 Feb 27 09:00 spool
drwxrwxrwx 6 root root 4096 Apr 9 18:15 tmp

sp2015722087@ubuntu:~/sp/shared/2-2$ ./advanced_ls Makefile spls.c ~/vmlinuz .. -l ERROR
cannot access ERROR: No such file or directory
-rwxrwxrwx 1 root root 51 Apr 9 00:21 Makefile
-rwxrwxrwx 1 root root 19825 Apr 9 20:33 spls.c
-rw-r--r-- 1 sp2015722087 sp2015722087 8128088 Mar 15 17:32 /home/sp2015722087/vmlinuz
Directory path: /mnt/hgfs/sh
total : 8
drwxrwxrwx 1 root root 0 Apr 5 19:39 2-1
drwxrwxrwx 1 root root 4096 Apr 9 20:36 2-2
drwxrwxrwx 1 root root 4096 Apr 9 20:35 2-3
-rwxrwxrwx 1 root root 46 Mar 29 16:58 desktop.ini

```

입력한 폴더 이름은 폴더명일수도 있고, 파일명일수도 존재하지 않는 파일명일수도 있다.

그리고 < 존재하지 않는 파일 -> 존재하는 파일 -> 존재하는 폴더 > 순서대로 출력해야하기 때문에, 입력한 폴더 이름을 배열로 받아와서 sort_folder() 함수를 구현하여 출력해야 하는 순서대로 폴더 이름들을 정렬하였다. 이 과정에서 존재하는 폴더인지 확인하고, 아니면 파일이 존재하는지 확인하여 정렬하였다.

폴더 이름 정렬 후 순서대로 해당 폴더의 파일들을 출력하는 과정에서도 옵션에 따른 구현이 필요하였다. -a 옵션을 가장 먼저 구분하였다. 모든 파일을 list 에 입력할 것인가, '으로 시작하지 않는 파일들만 list 에 입력할 것 인가 선택하고, 각 파일을 list 에 입력하면서 total size 에 각 파일들의 size 를 더해주었다. 출력은 옵션 -i 이 있어야 하는 것이기 때문에 -i 옵션이 없을 경우에는 더해 놓기만 하고 출력은 하지 않는다.

-i 옵션이 아닐 경우에는 파일의 이름만 출력하는 함수 view_list()를 이용했지만, -i 옵션일 경우에는 Full Format 과 함께 출력하여야 하기 때문에 새로운 함수 view_advanced_list()를 구현하였다. 인자로 파일의 lstat 정보를 같이 받아오기 때문에, 이 함수는 리스트의 출력이 아닌 하나의 파일만을 출력한다. 때문에 advanced_ls() 에서는 파일의 개수만큼 함수를 반복 호출해서 사용한다. Symbolic link 파일인 경우에는 링크된 파일의 경로를 함께 출력해줘야 하기 때문에 stat 정보가 아닌 lstat 정보를 넘겨주었다.

하나의 폴더가 출력이 완료되면 해당 폴더의 파일 list 를 동적 할당 해제해주었고, 모든 폴더가 출력이 완료되면 폴더 list 를 동적 할당 해제해주었다. 모든 동적 할당이 해제되고나면 프로그램이 종료된다.

Conclusion

이번 과제는 저번 과제의 결과물에서 더 기능이 추가된 결과물을 만드는 과제이다. 내가 옵션이 있는 ls 프로그램을 구현할 수 있을까 싶었지만, 막상 강의시간에 배운 함수들을 사용해보면서 익히고 나니, 적용하기만 하면 프로그램이 작동하는 경우도 있어서 괜히 걱정 했나 싶은 생각도 들었다.

가장 어려웠던 부분은 chdir()을 이용할 때 였다. 메인 함수에서 작업중인 디렉토리가 a/b/c 였고, 함수를 호출해서 함수에서 chdir()을 이용해서 a/b 디렉토리에서 작업을 하다가 함수가 종료가 되면 다시 작업중인 디렉토리가 a/b/c 일 줄 알았지만 a/b 그대로 남아있어서 오류 발생시 뭐가 문제인지 찾느라 시간을 들였다.

그래도 이 문제를 해결하느라 현재 작업중인 디렉토리가 어디인지 코드만 봐도 알 수 있게 되었고, 절대 경로에 있는 파일을 찾는 것을 다른 학생들이 조금 어려워하는 것을 보고 chdir()을 이용하여 작업중인 디렉토리를 변경하여 작업하는 것이 수월한 부분은 조금 알려주기도 하였다.

이번 과제를 하면서도 시간은 많이 들었지만 끝내고 보면 어렵지는 않은 과제였지만, 다음 과제가 나오면 또 똑같은 걱정을 하면서도 지금처럼 다음 과제도 해결할 수 있을 것 같다는 자신감이 생기는 과제였다.