

2019. 5. 1.

Assignment # 3-1

금요일 - 이성원 교수님

2015722087 컴퓨터정보공학부

김민철

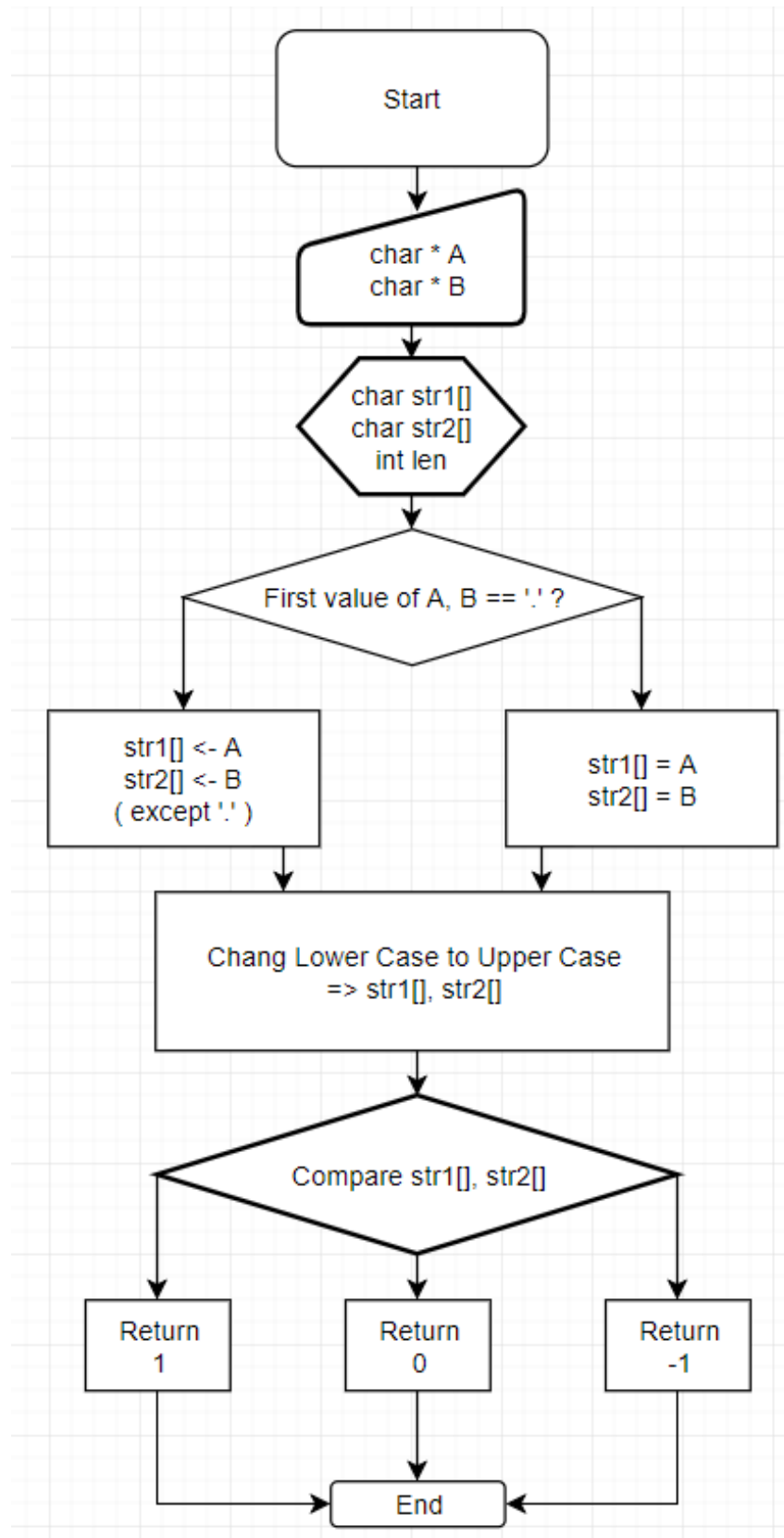
HTML_ls 구현

Introduction

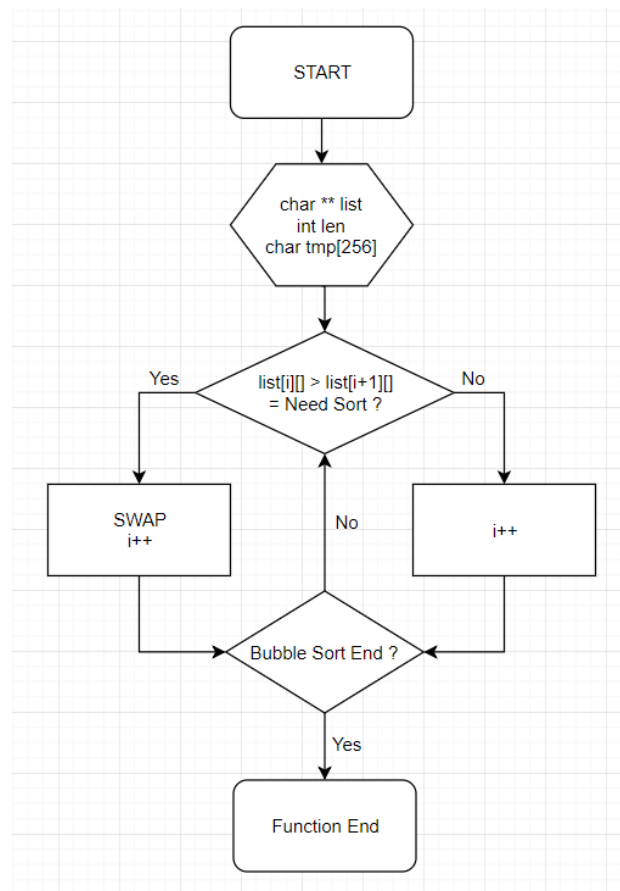
이번 과제에서는 2-3 과제에서 구현하였던 `final_ls` 를 이용하여 결과 출력을 Shell 에 하는 것이 아닌, Html 파일의 형태로 출력하여 웹 브라우저에서 결과를 확인할 수 있도록 하는 것이 목적이다. 추가적인 옵션이나 기능은 없으며, 결과 출력시 html 파일은 제외시키고, 하이퍼 링크, 파일 타입별 색상을 넣어 출력하도록 한다.

Flow Chart

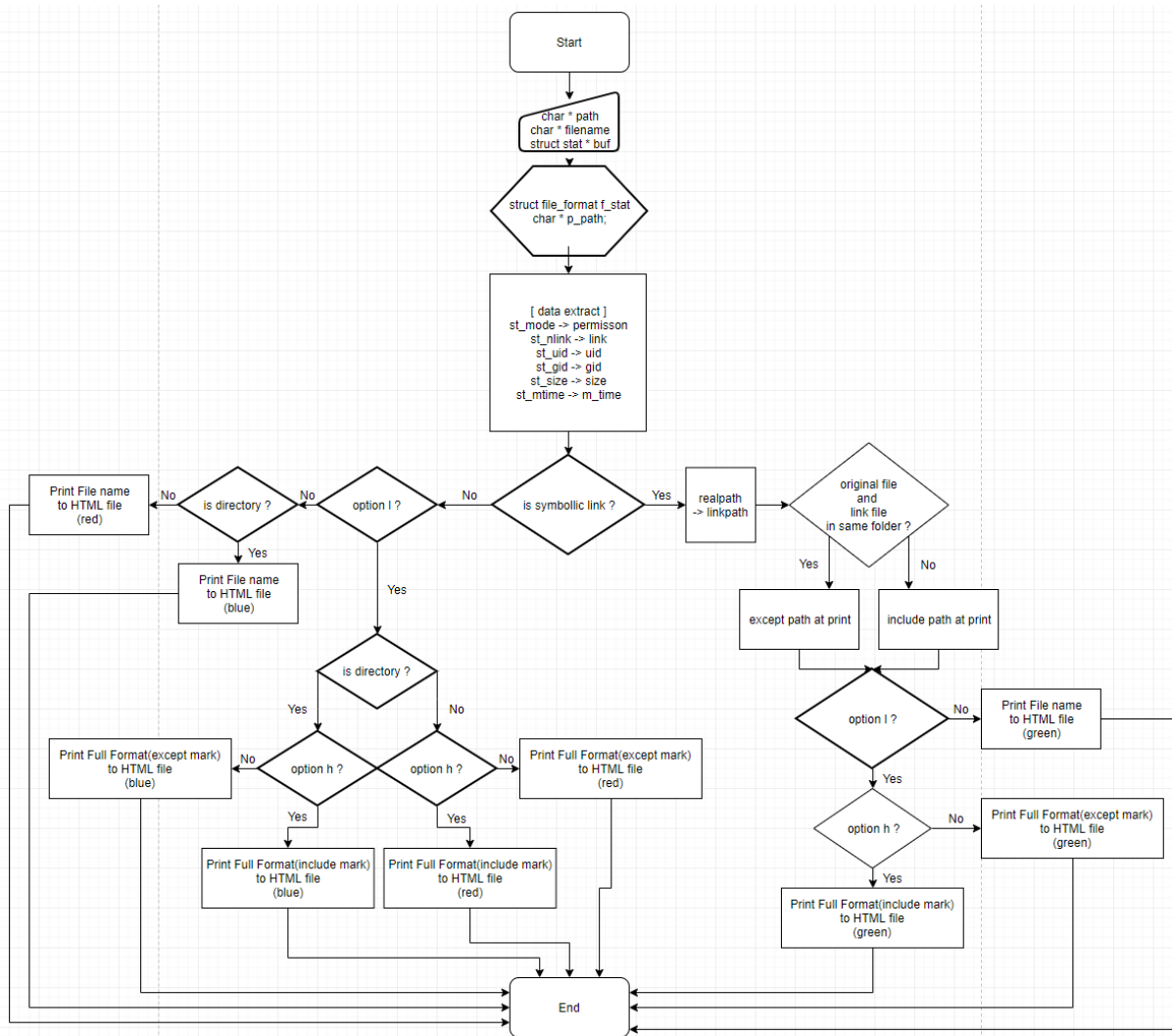
-int strcmp_i



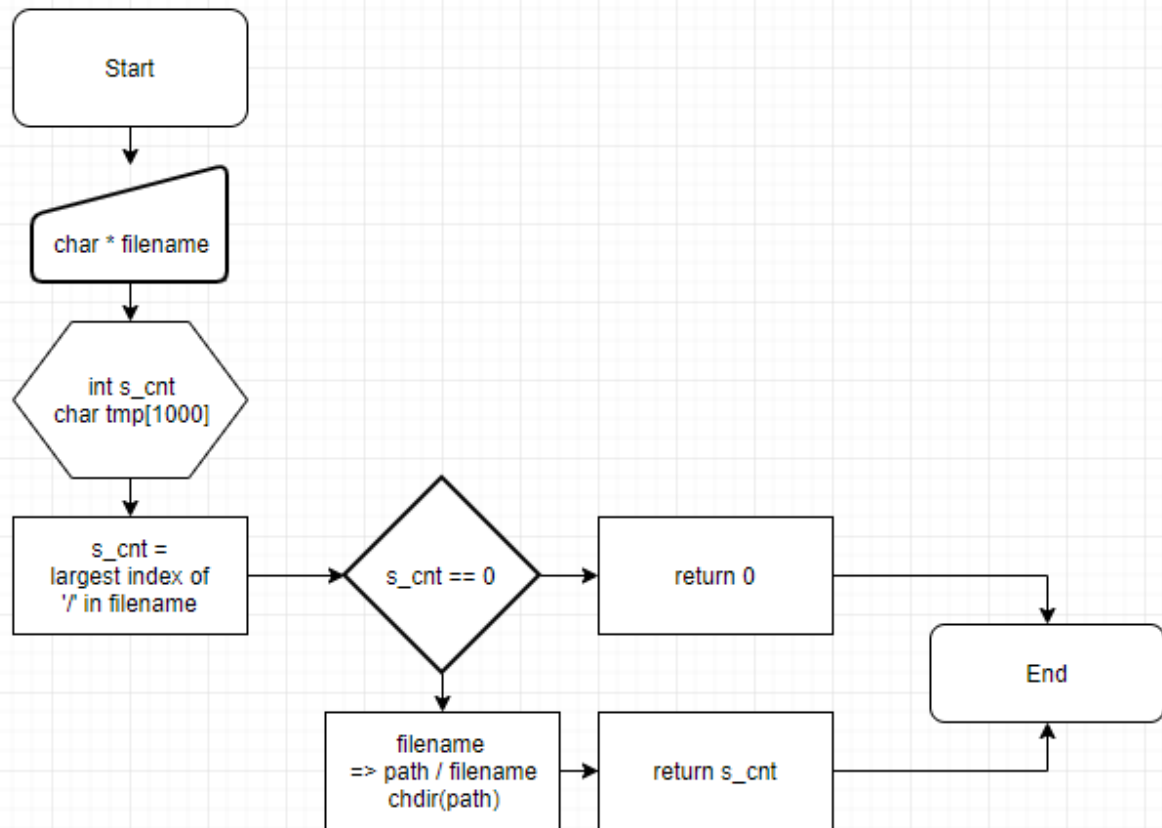
-void sort_list



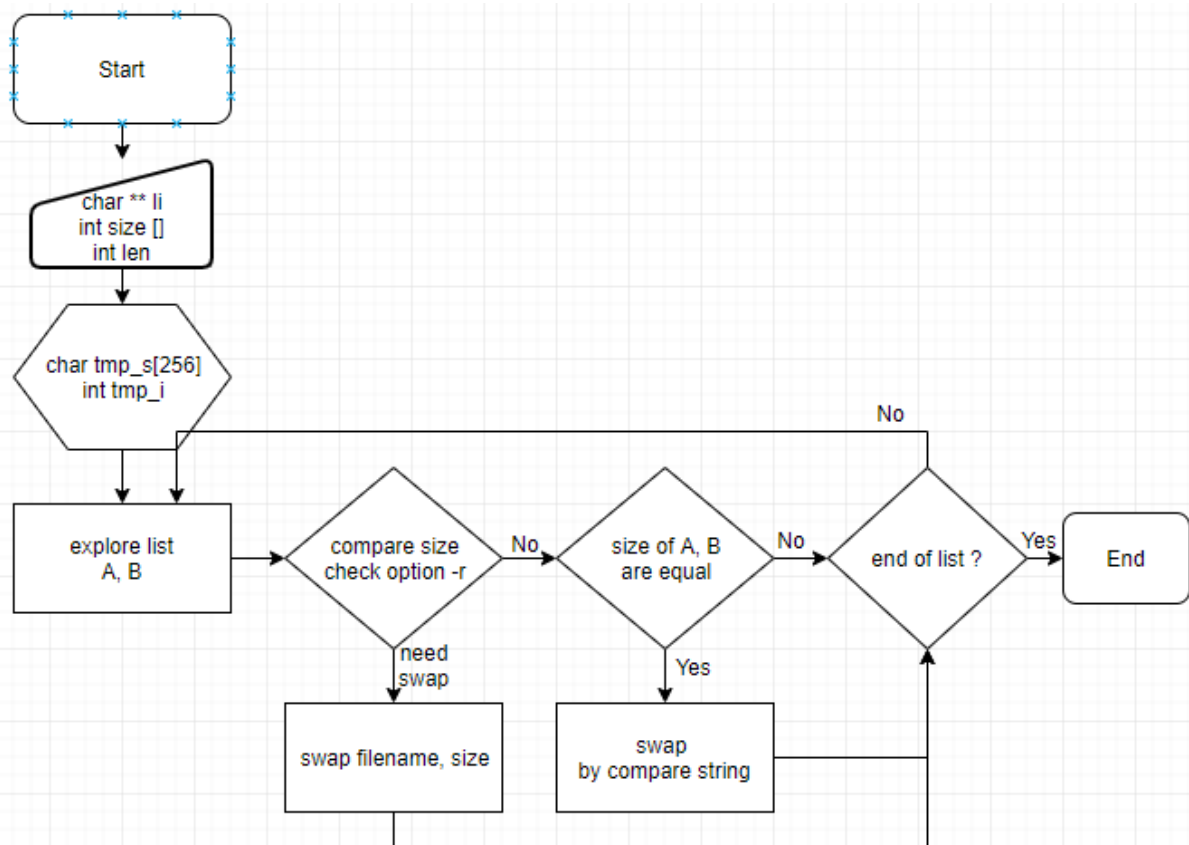
-void view_advanced_list



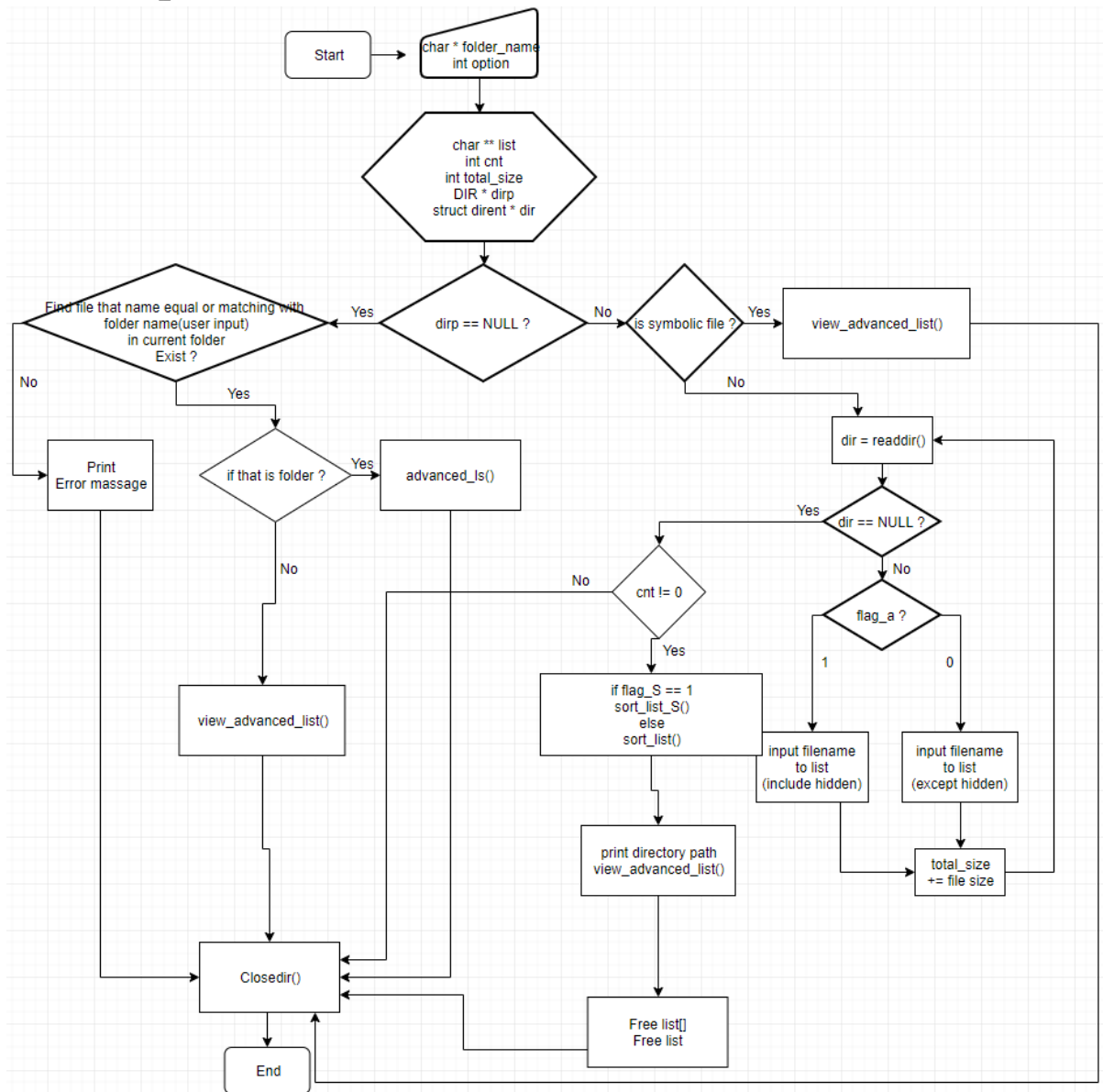
-int check_real_path



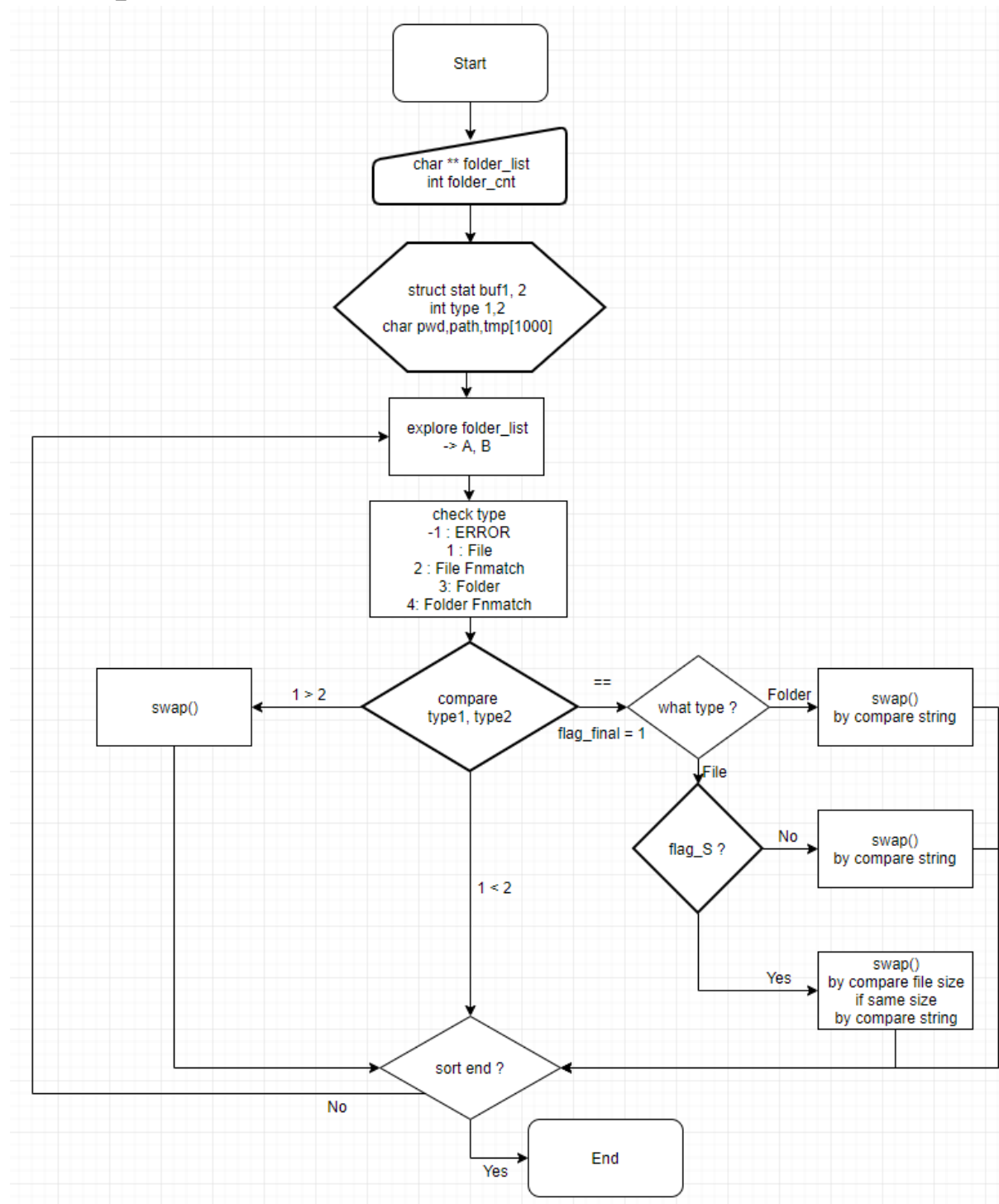
-void sort_list_S



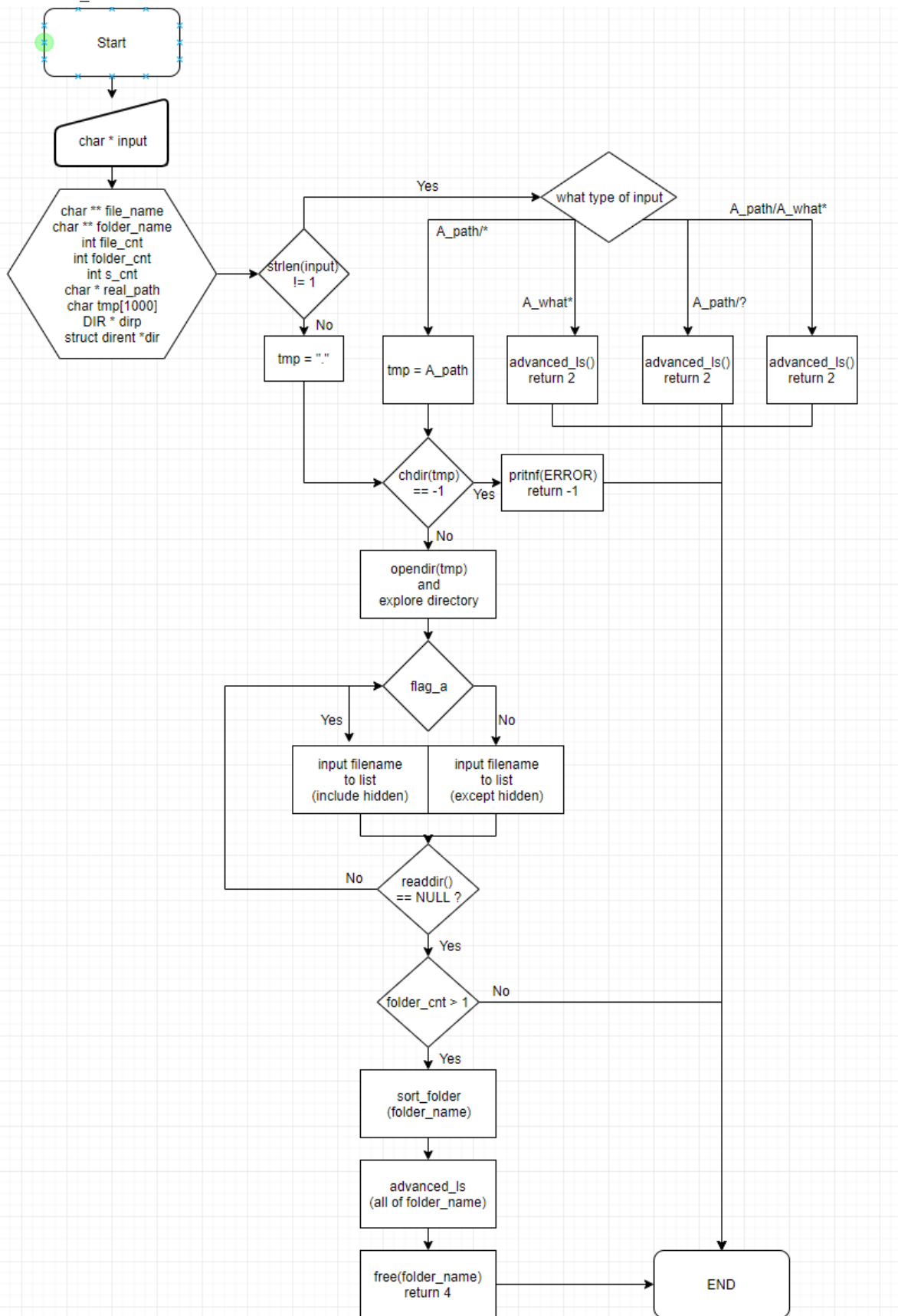
-int advanced_ls



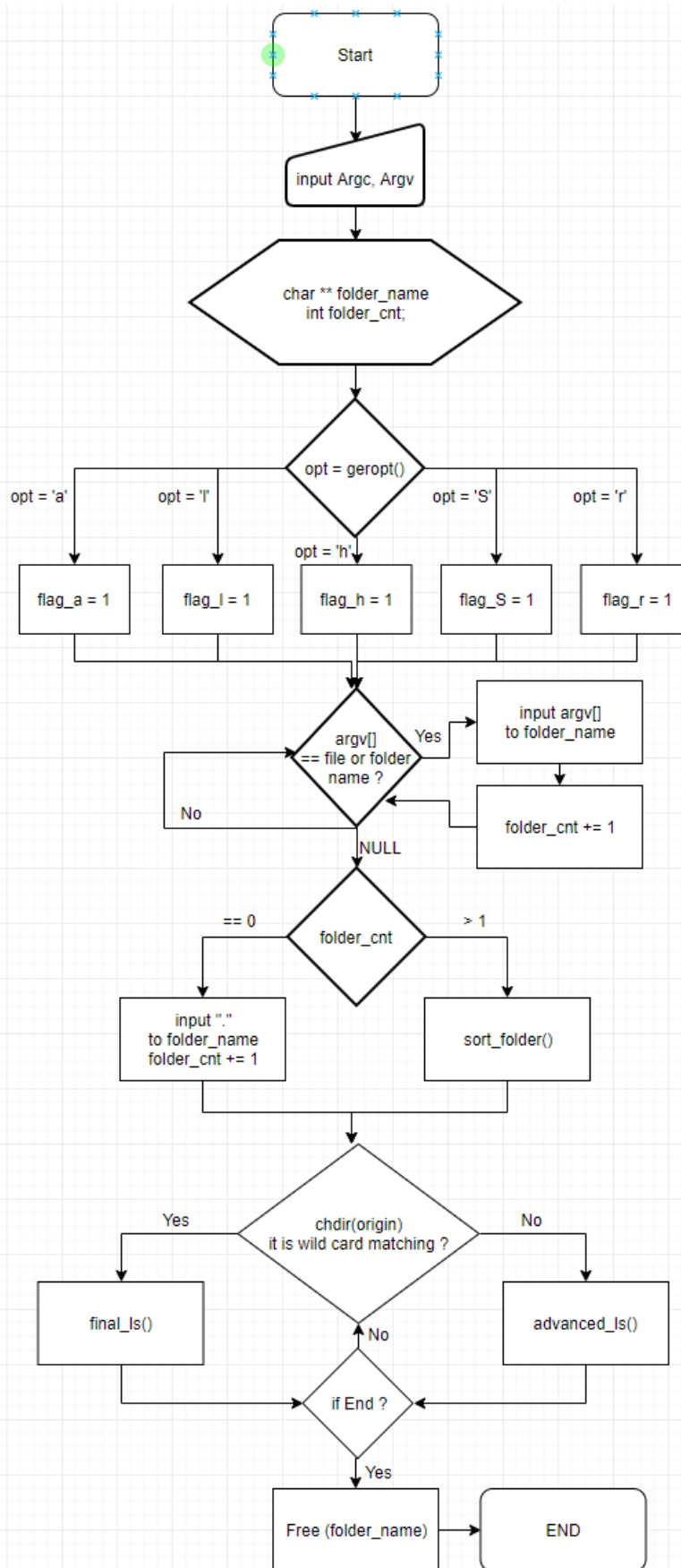
-void sort_folder



-int final_ls



-int main



Pseudo code

```
int check_real_path(char * filename)
{
    for (int i = 0; i < strlen(filename); i++) // check directory.
    {
        입력된 파일 이름에 '/' 가 있을 경우 해당 index => s_cnt
    }
    if 파일 이름에 '/' 가 있다면 -> real path
    {
        devide < path > / < file >

        <path> 로 작업중인 디렉토리 변경
        s_cnt 값 반환
    }
    else real path 가 아니라면
        0 반환
}
```

```
void swap(char * str1, char * str2)
{
    str1과 str2 swap
}
```

```
int strcmp_i ( char * 비교할 첫 번째 문자열 A, char * 비교할 두 번째 문자열 B )
{
    <예외 처리> : r 옵션이면 거꾸로.
    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 옵션이 -r 이라면
```

```

        if( A > B )      return = -1;
        else if( B > A )  return = 1;
        else             return = 0;

    else

        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 sort_list_S(char ** li, int size[], int len)
{
    // sort like bubble sort
    for (int i = 0; i < len - 1; i++) // total cycle
    {
        if 리스트가 끝났다면
            break;

        for (int j = 0; j < len - 1; j++) // one cycle
        {
            if 파일 사이즈가 다르다면, 정렬 필요 -r 옵션에 따라서.
            {
                swap file name
            }
        }
    }
}

```

```

        swap file size
    }
    else if 파일 사이즈가 같다면, -> 문자열로 정렬
    {
        if 정렬이 필요하다면
        {
            swap file name
        }
    }
}
return;
}

```

```

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->mtime

```

```

    if 옵션이 -h 라면
    size = size / 1024.0, check_h++(단위가 몇번 상승했는지 체크)
    check_h 값에 따라서 단위(K, M, G) 결정

```

```

    // Full Format 출력
    if 파일 type 이 symbolic link 라면
    {
        if | 옵션이 아니라면
            파일 이름만 HTML 파일로 출력 (green)
        else Full Format 출력
            If 옵션이 -h 라면, 사이즈 출력시 단위와 함께 출력
            Symbolic link 파일과 가리키는 경로 파일이 같은 폴더에 있다면 ->
가리키는 파일명만 HTML 파일로 출력(green)
            Symbolic link 파일과 가리키는 경로 파일이 다른 폴더에 있다면 -> 절대
경로 모두 HTML 파일로 출력(green)
    }
    else symbolic link 가 아니라면
    {
        if | 옵션이 아니라면
            directory일 경우 파일 이름만 HTML 파일로 출력 (blue)
            아닐 경우 파일 이름만 HTML 파일로 출력 (red)
        else | 옵션 이라면
            if directory라면 (blue)
                If 옵션이 -h 라면, FullFormat 출력시 단위도 HTML 파일로 출력

```

```

        else FullFormat HTML 파일로 출력
    else directory가 아니라면 (red)
        If 옵션이 -h 라면, FullFormat 출력시 단위도 HTML 파일로 출력
        else FullFormat HTML 파일로 출력
}

```

```

int advanced_ls(char * 입력한 폴더)
{
    if 폴더가 아니라면
    {
        혹시 절대경로로 입력된 폴더인지 확인        -> 맞으면 해당 디렉토리로 변경
                                                    -> 아니면 현재 디렉토리

        HTML 파일이면 return
        if table이 만들어져 있지 않고, flag_p가 -1이 아니라면,
            옵션 l 에 따라 테이블 생성 -> flag_table=0;

        while 디렉토리 탐색
        {
            if 파일을 찾았다면
                view_advanced_list()
            If fnmatch 로 매칭이 된다면,
                폴더면 다시 advanced_ls()
                파일이면 view_advanced_list()
        }
        If 출력한 파일이 없다면
            에러 메시지 출력 -> 반환 -1
        else 1 반환
    }
    else // is folder
    {
        p 플래그가 표시되어있다면 return 3 => 폴더
        flag_table == 0 이라면 테이블이 열려있으므로 close
        flag_table = 1 set -> 테이블이 안만들어져있다.

        심볼릭 링크로 연결된 폴더인지 확인
        -> 심볼릭 링크 폴더이고 옵션이 l 이면 심볼릭 링크만 Full Format 출력

        아니면 탐색 시작
        while 입력된 폴더 탐색
        {
            HTML 파일은 처리하지 않음.

            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
        }
    }

    Directory path 출력
    if 파일이 존재한다면
    {
        List 정렬.
        Total size 출력

        if l 옵션이라면 -> Full Format Table 생성
        else -> 파일 이름 Table 생성
        for 파일 개수 만큼 반복
            view_advanced_list()

        테이블 close
        flag_table=1
    }
}
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)
        {
            만약 final_ls 함수에서 정렬이라면 chdir(final_wd)
            아니라면(main에서 정렬) chdir(origin_wd)

            표시 데이터 전부 리셋

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

            If 정렬이 필요하다면,
                폴더 이름으로 정렬
            else if 같은 타입이고 final 에서 정렬하는 경우
            {
                if 둘다 폴더일 경우
                    폴더 이름으로 정렬
                else 둘다 파일인 경우
                    if S 옵션이 있을 경우
                        파일 크기로 정렬 -> 같을 경우 이름으로 정렬
                    else
                        파일 이름으로 정렬
            }
        }
    }
}

```

```

    }

    if 마지막 cycle이고, final에서 호출했다면
        if 파일이고, origin 디렉토리가 아니라면,
            파일 이름앞에 절대 경로 입력
    }
}

```

```

int final_ls(char * input)
{
    if input의 길이가 1이 아닌경우
        if A_what* 같이 입력된 경우
            advanced_ls () 호출 -> 2 반환
        else if A_path / A_What* 같이 입력된 경우
            폴더가 존재하는 지 확인 -> 없으면 에러
            존재하면 advanced_ls () 호출 -> 2 반환
        else if A_path/* 같이 입력된 경우
            존재하는 폴더인지 확인 -> 없으면 에러
        else
            폴더가 존재하는 지 확인 -> 없으면 에러
            존재하면 advanced_ls () 호출 -> 2 반환
    else 길이가 1인 경우
        A_path = 현재 경로

    if A_path 가 폴더가 아닌 경우
        에러 메시지 출력 -> -1 반환
    else A_path가 폴더인 경우
    {
        플래그 p 가 -1 이면 4 반환

        A_path 로 작업중인 디렉토리 변경
        final_wd 에 A_path 저장
        A_path 디렉토리 opendir()

        while 디렉토리 탐색
        {
            if -a 옵션일 경우 모든 파일을 리스트에 저장, ('.', '..' 제외)
            {
                폴더 이름 리스트 동적할당
                파일 이름을 폴더 이름 리스트에 입력
                폴더 개수 카운트 ++
            }
            else if -a 옵션이 아닐 경우 히든 파일 제외한 파일 리스트에 저장.
            {
                폴더 이름 리스트 동적할당
                파일 이름을 폴더 이름 리스트에 입력
                폴더 개수 카운트 ++
            }
        }

        if 파일이 존재할 경우

```



```

    {
        flag_final 를 1 로 변경. -> final_ls 에서 정렬을 시작한다고 알림
        sort_folder() -> 리스트 정렬
    }

    final_wd 출력, 출력할 폴더, 파일들의 상위 폴더. = final_ls가 실행되는 디렉토리
    for (int i = 0; i < folder_cnt; i++) // Print folders !
    {
        final_wd 로 디렉토리 변경
        폴더 혹은 파일 출력 -> advanced_ls()
    }

    for (int a = 0; a < folder_cnt; a++)
    {
        동적 할당 해제
    }
    동적 할당 해제
}

flag_final = 0;
4 반환
}

```

```

int main(int 입력한 인자 개수, char * 입력한 인자 문자열)
{
    HTML 파일 오픈
    HTML 파일 기본 포맷 입력, 타이틀에 현재 디렉토리 입력.
    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;
            case 'h': // include option h : change format of print file size
                -h 옵션이 있다면 l 옵션 flag ++
                break;
            case 'r': // include option r : reverse sort
                -r 옵션이 있다면 l 옵션 flag ++
                break;
            case 'S': // include option S : sort by file size
                -S 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 리스트 동적 할당
        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()

heading에 h1 굵기로 입력된 명령어 출력
for (폴더 개수 만큼 반복)// Print folders !
{
    origin_wd 로 작업중인 디렉토리 변경

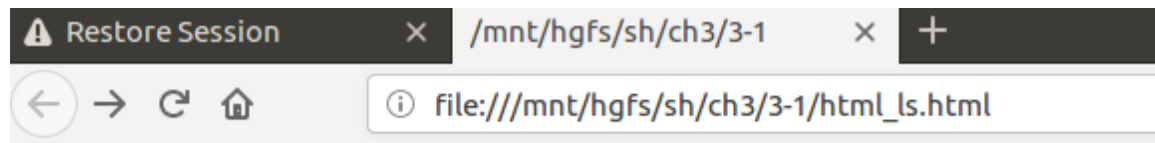
    -> '*' 출력일 경우 -> final_ls()
    -> 아니라면 -> advanced_ls()
}
테이블 close
html 파일 포맷 close
open한 파일 close

폴더 리스트 동적 할당 해제

프로그램 종료
}

```

Result



./html_ls -al . ../../*

Directory path: /mnt/hgfs/sh/ch3/3-1
total : 113

| File Name | Permission | Link | Owner | Group | Size | Last Modified |
|-----------|------------|------|-------|-------|-------|---------------|
| . | drwxrwxrwx | 1 | root | root | 4096 | May 1 22:59 |
| .. | drwxrwxrwx | 1 | root | root | 0 | Apr 30 13:13 |
| html_ls | -rwxrwxrwx | 1 | root | root | 39224 | May 1 22:59 |
| Makefile | -rwxrwxrwx | 1 | root | root | 43 | May 1 22:59 |
| spls | -rwxrwxrwx | 1 | root | root | 39272 | May 1 16:08 |
| spls.c | -rwxrwxrwx | 1 | root | root | 34184 | May 1 22:58 |
| text1.txt | -rwxrwxrwx | 1 | root | root | 0 | Apr 12 10:20 |
| text4.txt | -rwxrwxrwx | 1 | root | root | 0 | Apr 12 10:20 |
| text9.txt | -rwxrwxrwx | 1 | root | root | 0 | Apr 12 10:20 |

-> title 에 현재 경로가 출력되었고, heading 으로 h1 굵기로 명령어를 출력한 화면이다.
directory 파일은 blue, other 파일은 red 색상으로 출력하도록 하였다.

Directory path: /mnt/hgfs/sh

| File Name | Permission | Link | Owner | Group | Size | Last Modified |
|--|------------|------|-------|-------|------|---------------|
| /mnt/hgfs/sh/desktop.ini | -rwxrwxrwx | 1 | root | root | 46 | Mar 29 16:58 |

Directory path: /mnt/hgfs/sh/ch2

total : 12

| File Name | Permission | Link | Owner | Group | Size | Last Modified |
|---------------------|------------|------|-------|-------|------|---------------|
| . | drwxrwxrwx | 1 | root | root | 0 | Apr 30 13:13 |
| .. | drwxrwxrwx | 1 | root | root | 4096 | Apr 30 13:13 |
| 2-1 | drwxrwxrwx | 1 | root | root | 0 | Apr 5 19:39 |
| 2-2 | drwxrwxrwx | 1 | root | root | 4096 | Apr 12 15:21 |
| 2-3 | drwxrwxrwx | 1 | root | root | 4096 | Apr 27 00:12 |

Directory path: /mnt/hgfs/sh/ch3

total : 8

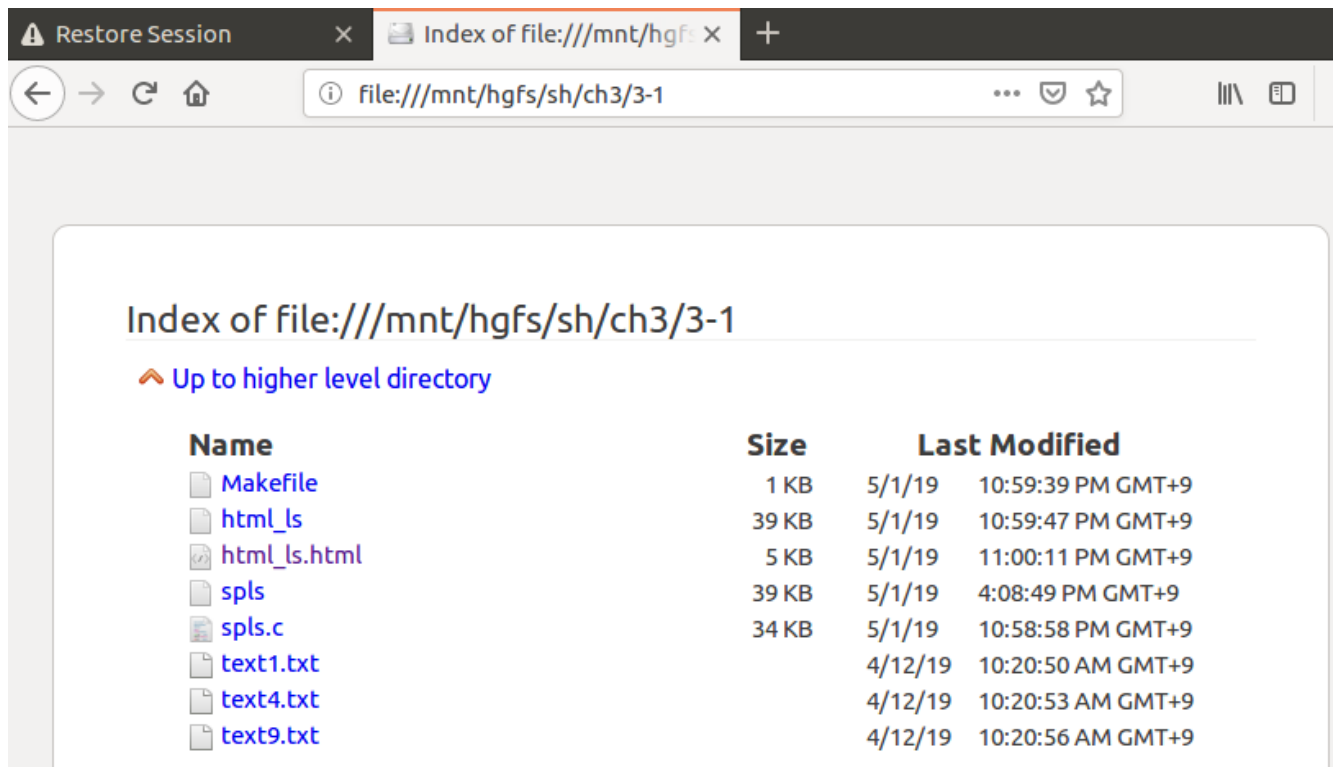
| File Name | Permission | Link | Owner | Group | Size | Last Modified |
|---------------------|------------|------|-------|-------|------|---------------|
| . | drwxrwxrwx | 1 | root | root | 0 | Apr 30 13:13 |
| .. | drwxrwxrwx | 1 | root | root | 4096 | Apr 30 13:13 |
| 3-1 | drwxrwxrwx | 1 | root | root | 4096 | May 1 22:59 |

Directory path: /mnt/hgfs/sh/test

total : 72

| File Name | Permission | Link | Owner | Group | Size | Last Modified |
|--------------------------|------------|------|-------|-------|-------|---------------|
| . | drwxrwxrwx | 1 | root | root | 4096 | Apr 27 00:47 |
| .. | drwxrwxrwx | 1 | root | root | 4096 | Apr 30 13:13 |
| ls | -rwxrwxrwx | 1 | root | root | 34184 | Apr 27 00:47 |
| Makefile | -rwxrwxrwx | 1 | root | root | 33 | Apr 27 00:47 |
| test.c | -rwxrwxrwx | 1 | root | root | 31697 | Apr 27 00:47 |

-> 이후 fnmatch 를 통해 디렉토리 내의 파일을 출력하고 모든 directory 를 open 하여 모든 파일을 출력한 모습이다. 역시 파일 타입에 따라 색상을 다르게하여 출력하였다.



-> 모든 파일에 해당 파일의 링크를 hyperlink 로 연결하고, 클릭했을때의 화면이다.

Conclusion

이번 과제는 처음으로 내가 구현한 프로그램의 출력 결과를 MFC 나 shell 이 아닌 브라우저로 확인해야하는 과제라서 처음에는 감이 잡히지를 않았다. 어떻게 html 로 출력을 할 수 있을까 생각만하다가 강의자료에 나와있는 html 파일의 포맷을 참고하여 프로그램이 파일에 출력을 포맷대로 하게 하였더니 결과를 브라우저에서 확인할 수 있게 되었다.

이후로는 코드의 출력부분만 HTML 포맷에 따라서 출력하게 해주었더니 모두 정상적으로 출력되었다. 어려웠던 부분은 table 을 생성하고 close 하는 부분에서 제대로 되지 않아 결과가 예상과는 다르게 나왔었는데 디버깅을 하면서 테이블을 닫기전에 테이블을 다시 생성하는 오류를 찾아 해결하였다.

항상 해보기 전에는 어려워서 어떻게 할까 생각하지만, 막상 시도해보면서 공부하면 해결할 수 있다는 생각이 드는 과제였다.