PRACTICAL-3

Aim:

Write a program for ‘ls’ command using ‘opendir()’ and ‘readdir()’ system call.

Note 1 : -a and -A , -R should be argument to main(). Create all in one program.

Note 2: Argument of opendir() should automatically take present working directory for which you want to list the content.

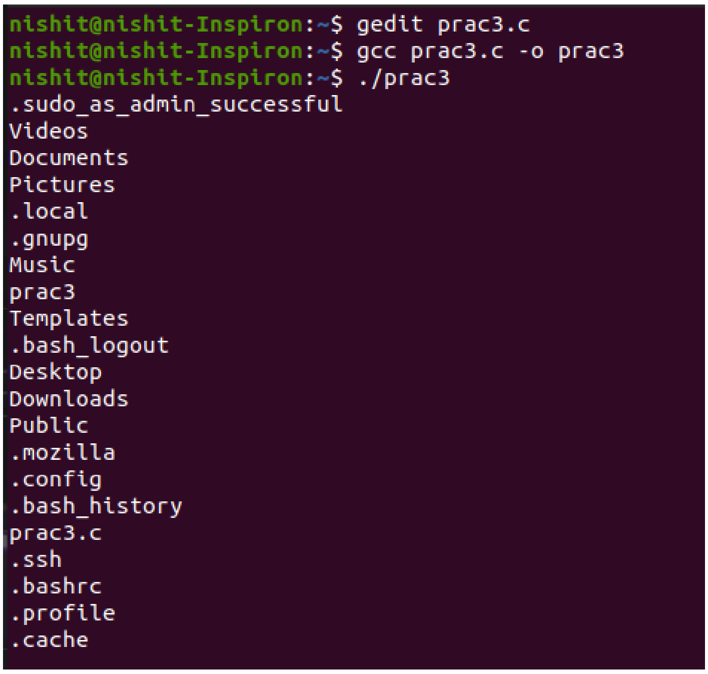
Hint: for ls -R you can use d_type field of struct dirent.

Program Code:

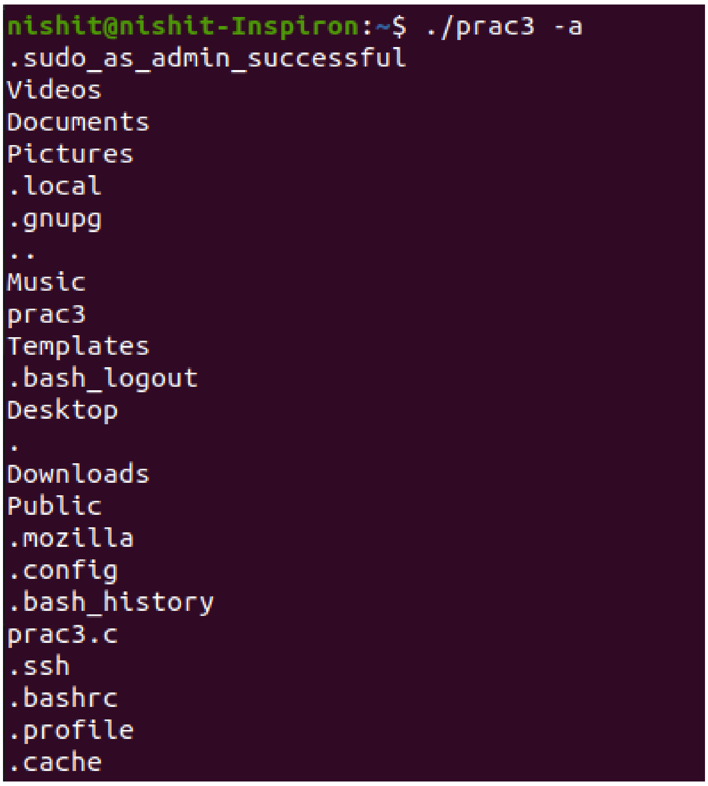
```
#include<sys/types.h>

#include<fcntl.h>

#include<stdio.h>

#include<dirent.h>

#include<stdlib.h>

#include<string.h>

#include<unistd.h>

#include<linux/limits.h>


void listFilesRecursively(char* mainPath) {


    struct dirent *dirPtr;

    DIR *ptr = opendir(mainPath);

    char path[100];

    if(ptr == NULL) {

        printf("\nCannot open the current directory. Please try again.\n");

        exit(0);

    }


    while((dirPtr = readdir(ptr)) != NULL) {

        if(strcmp(dirPtr -> d_name, ".") != 0 && strcmp(dirPtr -> d_name, "..") != 0) {
                printf("%s\n", dirPtr -> d_name);
                if(dirPtr -> d_type == DT_DIR && strcmp(dirPtr -> d_name, ".") != 0 &&
strcmp(dirPtr -> d_name, "..") != 0) {
                        strcpy(path, mainPath);
                        strcat(path, "/");
                        strcat(path, dirPtr -> d_name);
                        listFilesRecursively(path);
                }
        }
    }
    closedir(ptr);
}

void isLowerCaseA() {

    struct dirent *dirPtr;
    char currentDir[256];
    getcwd(currentDir, sizeof(currentDir));
    DIR *ptr = opendir(currentDir);

    if(ptr == NULL) {
        printf("\nCannot open the current directory. Please try again.\n");
        exit(0);
    }

    while((dirPtr = readdir(ptr)) != NULL) {
        printf("%s\n", dirPtr -> d_name);
    }
    closedir(ptr);
}
```

```
void isUpperCaseA() {

    struct dirent *dirPtr;
    char currentDir[256];
    getcwd(currentDir, sizeof(currentDir));
    DIR *ptr = opendir(currentDir);

    if(ptr == NULL) {
        printf("\nCannot open the current directory. Please try again.\n");
        exit(0);
    }

    while((dirPtr = readdir(ptr)) != NULL) {
        if((strcmp(dirPtr -> d_name, ".") == 0) || (strcmp(dirPtr -> d_name, "..") == 0))
            continue;
        printf("%s\n", dirPtr -> d_name);
    }
    closedir(ptr);
}

int main(int argc, char* argv[]) {

    if(argc > 2) {
        printf("Invalid number of Command Line Arguments Passed. Please try again.\n");
        exit(0);
    }
    char currentDir[256];
    getcwd(currentDir, sizeof(currentDir));
    if(argc == 1) {
        isUpperCaseA();
        printf("\n");
        return 0;
    }

    if((strcmp(argv[1], "-a")) == 0) {
        isLowerCaseA();
    }

    else if((strcmp(argv[1], "-A")) == 0) {
        isUpperCaseA();
    }

    else if((strcmp(argv[1], "-R")) == 0) {
        listFilesRecursively(currentDir);
    }

    else {
        printf("Please pass an appropriate Variants of ls Commmand.\n");
    }
    printf("\n");
    return 0;
}
```

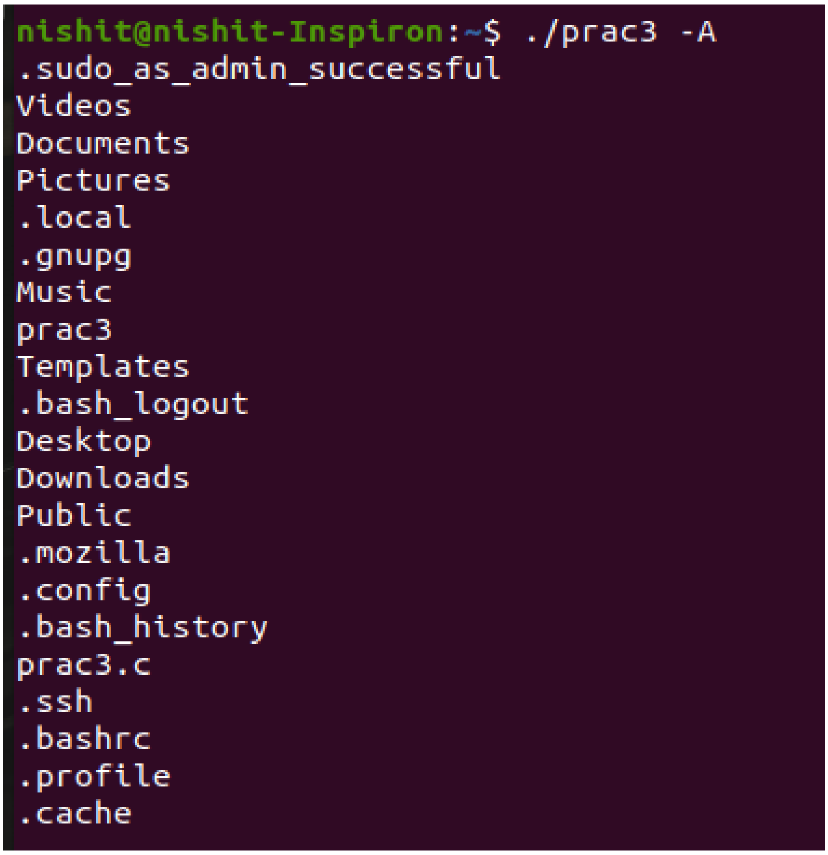
Output:



```
nishit@nishit-Inspiron:~$ gedit prac3.c
nishit@nishit-Inspiron:~$ gcc prac3.c -o prac3
nishit@nishit-Inspiron:~$ ./prac3
.sudo_as_admin_successful
Videos
Documents
Pictures
.local
.gnupg
Music
prac3
Templates
.bash_logout
Desktop
Downloads
Public
.mozilla
.config
.bash_history
prac3.c
.ssh
.bashrc
.profile
.cache
```



```
nishit@nishit-Inspiron:~$ ./prac3 -a
.sudo_as_admin_successful
Videos
Documents
Pictures
.local
.gnupg
..
Music
prac3
Templates
.bash_logout
Desktop
.
Downloads
Public
.mozilla
.config
.bash_history
prac3.c
.ssh
.bashrc
.profile
.cache
```



```
nishit@nishit-Inspiron:~$ ./prac3 -A
.sudo_as_admin_successful
Videos
Documents
Pictures
.local
.gnupg
Music
prac3
Templates
.bash_logout
Desktop
Downloads
Public
.mozilla
.config
.bash_history
prac3.c
.ssh
.bashrc
.profile
.cache
```