Lab 5 – Directory Structures

```
#include <stdlib.h>
#include <unistd.h>
#include <stdio.h>
#include <sys/types.h>
#include <dirent.h>
#include <string.h>

void dumpDir(DIR *dir, int indent, char *base) {
        struct dirent *entry;   // the current directory entry
        char *name;             // the name of the entry
        int type;               // the type of the directory entry
        char *typename;
        int len;

        DIR *newdir;            // directory stream for recursive listing
        char *dirname;          // full name of sub-directory

        entry = readdir(dir);
        while(entry != NULL) {
                name = entry->d_name;
                type = entry->d_type;

                //get type of directory entry
                switch (type) {
                        case 0:
                                typename = "DT_UNKNOWN";
                                break;

                        case 1:
                                typename = "DT_FIFO";
                                break;

                        case 2:
                                typename = "DT_CHR";
                                break;

                        case 4:
                                typename = "DT_DIR";
                                break;

                        case 6:
                                typename = "DT_BLK";
                                break;

                        case 8:
                                typename = "DT_REG";
                                break;

                        case 10:
                                typename = "DT_LNK";
                                break;

                        case 12:
                                typename = "DT_SOCK";
                                break;

                        case 14:
                                typename = "DT_WHT";
                                break;

                        default:
                                break;
                }

                //skip filenames that start with a period
                if(name[0] != '.') {
                        for(int i = 0; i < indent; i++) printf("%s", " ");
                        printf("%ld %s %s\n",entry->d_ino, name, typename);

                        //recursive directory listing
                        if(type == DT_DIR) {
                                len = strlen(base) + strlen(name) + 2;
                                dirname = (char*) malloc(len);

                                strcpy(dirname, base);
                                strcat(dirname, "/");
                                strcat(dirname, name);

                                newdir = opendir(dirname);

                                dumpDir(newdir, indent+2, dirname);
                                closedir(newdir);
                                free(dirname);
                        }
                }

                entry = readdir(dir);
        }
}

int main(int argc, char **argv) {
        DIR *dir;

        if(argc != 2) {
                printf("usage: lab5 directory\n");
                exit(1);
        }

        dir = opendir(argv[1]);
        if(dir == NULL) {
                printf("can't open directory: %s\n", argv[1]);
                exit(1);
        }

        dumpDir(dir,0,argv[1]);
        closedir(dir);
}
```

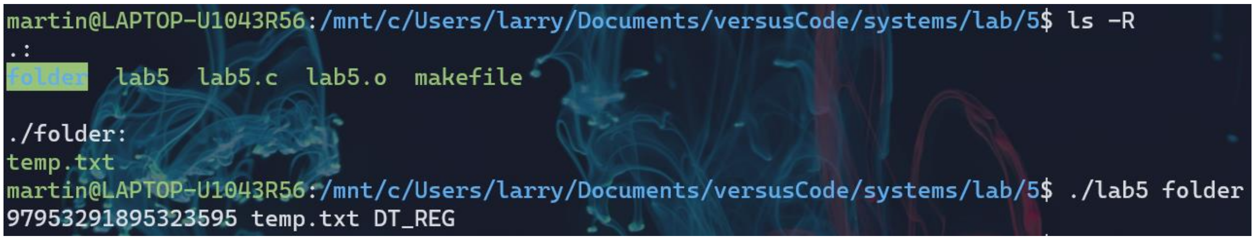
Makefile

```
CFLAGS = -Wall -g

run: lab5.o
	cc -o lab5 lab5.o

clean:
	rm lab5 lab5.o
```

Output



```
martin@LAPTOP-U1043R56:/mnt/c/Users/larry/Documents/versusCode/systems/lab/5$ ls -R
.:
folder  lab5  lab5.c  lab5.o  makefile

./folder:
temp.txt
martin@LAPTOP-U1043R56:/mnt/c/Users/larry/Documents/versusCode/systems/lab/5$ ./lab5 folder
97953291895323595 temp.txt DT_REG
```