**cs3423 Proj7 Low Level I/O (20 pts)**

You will use the C low level I/O functions to create our own **`xdir`** command in C. This program must **not** use bash, ls, sed, awk, and/or Perl. It must not use any of the exec functions.

Syntax of xdir:

```
xdir directory
xdir directory switches
```

Your source code must be named **`xdir.c`**

**In each of the cases, we show the directory. The directory entries are then indented four spaces.**

**Without any switches**

- xdir will first print the directory name followed by a colon.
- xdir will not print any files that begin with "."
- xdir will print the filenames (unqualified) in the order they are provided by the readdir function.

```
> xdir Data
Data :
    file5
    file6
    Program4
    Program5
    Program2
    Program3
    file4
```

**With just the `-l` switch**

- xdir will first print the directory name followed by a colon.
- xdir will not print any files that begin with "."
- For each file, xdir will print (on one line):
    - file name (unqualified)
    - file type (F - regular file, D - directory, L - link, P - Pipe)
    - number of 512 byte blocks
    - size in bytes
- xdir will print the files in the order they are provided by the `readdir` function.

```
> xdir Data -l
Data :
    file5 F 8 blks 167 bytes
    file6 F 8 blks 121 bytes
    Program4 D 8 blks 4096 bytes
    Program5 D 8 blks 4096 bytes
    Program2 D 8 blks 4096 bytes
    Program3 D 8 blks 4096 bytes
    file4 F 8 blks 87 bytes
```

**With just the `-a` switch**

- xdir will first print the directory name followed by a colon.
- xdir will print any files in the directory including ones that begin with a "."
- xdir will print the filenames (unqualified) in the order they are provided by the `readdir` function.

```
> xdir Data -a
Data :
    .
    ..
    file5
    file6
    .mydot
    Program4
    Program5
    Program2
    Program3
    file4
```

**With both the `-l` and `-a` switches**

- xdir will first print the directory name followed by a colon.
- xdir will print any files in the directory including ones that begin with a "."
- For each file, xdir will print (on one line):
    - file name (unqualified)
    - file type (F - regular file, D - directory, L - link, P - Pipe)
    - number of 512 byte blocks
    - size in bytes
- xdir will print the files in the order they are provided by the `readdir` function.

```
> xdir Data -l -a
Data :
    . D 8 blks 4096 bytes
    .. D 8 blks 4096 bytes
    file5 F 8 blks 167 bytes
    file6 F 8 blks 121 bytes
    .mydot F 8 blks 25 bytes
    Program4 D 8 blks 4096 bytes
    Program5 D 8 blks 4096 bytes
    Program2 D 8 blks 4096 bytes
    Program3 D 8 blks 4096 bytes
    file4 F 8 blks 87 bytes
```

**Notes**:

1. Your code must be written according to my programming standards.
2. The command **switches can appear in any order**.
3. You may have to construct a file name (when you invoke the `stat` function):
   - You may assume that the length of the combination of the directory name and a file name will be less than 500 characters.
   - Use `snprintf`(szFullFileNm, 500, "`%s/%s`", pszDirectoryNm, *fileNm*);
4. Each directory entry must be indented 4 spaces more than its directory.
5. Larry provided sample data, a makefile, and errExit.c. copy them into your current directory (where your C source code will be located):
   ```
   cp -r /usr/local/courses/clark/cs3423/2017Fa/Proj7/* .
   ```
6. To compile and link your code:
   ```
   make xdir
   ```
   For more information about the **`make`** utility, see
   http://www.cs.utsa.edu/~clark/setup/UnixMakeUtility.pdf
7. Turn in a zip file (named LastnameFirstname.zip using your name). It should contain
   - `xdir.c` - your source code
   - `xdir.h` - (if you created one)
   - `makefile` - your makefile to make the **`xdir`** executable.
   - In the notes in BlackBoard, specify if extra credit was completed.

**Extra Credit (5 pts + 100 / N)**

1. This adds another switch, "**`-r`**", which is used to recurse to subdirectories.
   - The directory entries "." and ".." display as before. These **are not recursively** invoked since that could cause an infinite loop. Only show those when `-a` is specified.
   - When both `-r` and `-l` are provided, directories are displayed differently
     o A directory is shown without the type and size information.
   - A directory entry that is a directory, will show just a colon after it. We then recurse to it and show its entries.
   - Entries in a directory will be indented four spaces. This causes the display to look more like a tree. (see the example)
2. If the `-r` switch is not given, **`xdir`** should behave as previously described.
3. Extra credit is **NOT** given to **late** assignments.
4. All requirements must be met to receive extra credit.
5. N is the number of people to meet all requirements on time.

Extra Credit Examples:

```
> xdir Data -l -r -a
Data :
    . D 8 blks 4096 bytes
    .. D 8 blks 4096 bytes
    file5 F 8 blks 167 bytes
    file6 F 8 blks 121 bytes
    .mydot F 8 blks 25 bytes
    Program4 :
        . D 8 blks 4096 bytes
        .. D 8 blks 4096 bytes
        p4Book.txt F 8 blks 949 bytes
```

```
        cs1713p4.h F 16 blks 5716 bytes
        cs1713p4Driver.c F 48 blks 23503 bytes
        p4Command.txt F 8 blks 1750 bytes
        p4Commandv2.txt F 8 blks 1750 bytes
    Program5 :
        . D 8 blks 4096 bytes
        .. D 8 blks 4096 bytes
        Input5 :
            . D 8 blks 4096 bytes
            .. D 8 blks 4096 bytes
            p5Book.txt F 8 blks 949 bytes
            p5Command.txt F 8 blks 1774 bytes
        Makefile F 8 blks 368 bytes
        cs1713p5Driver.c F 48 blks 23337 bytes
        cs1713p5.h F 16 blks 5880 bytes
    Program2 :
        . D 8 blks 4096 bytes
        .. D 8 blks 4096 bytes
        cs1713p2.h F 16 blks 4570 bytes
        Input :
            . D 8 blks 4096 bytes
            .. D 8 blks 4096 bytes
            p2Book.txt F 8 blks 947 bytes
            p2Customer.txt F 8 blks 876 bytes
        cs1713p2Stuff.c F 16 blks 7960 bytes
        p2Out.txt F 16 blks 5043 bytes
    Program3 :
        . D 8 blks 4096 bytes
        .. D 8 blks 4096 bytes
        p3Book.txt F 8 blks 949 bytes
        cs1713p3.h F 16 blks 5037 bytes
        cs1713p3Driver.c F 40 blks 20430 bytes
        p3Command.txt F 8 blks 1598 bytes
    file4 F 8 blks 87 bytes

> xdir Data -l -r
Data :
    file5 F 8 blks 167 bytes
    file6 F 8 blks 121 bytes
    Program4 :
        p4Book.txt F 8 blks 949 bytes
        cs1713p4.h F 16 blks 5716 bytes
        cs1713p4Driver.c F 48 blks 23503 bytes
        p4Command.txt F 8 blks 1750 bytes
        p4Commandv2.txt F 8 blks 1750 bytes
    Program5 :
        Input5 :
            p5Book.txt F 8 blks 949 bytes
            p5Command.txt F 8 blks 1774 bytes
        Makefile F 8 blks 368 bytes
        cs1713p5Driver.c F 48 blks 23337 bytes
        cs1713p5.h F 16 blks 5880 bytes
    Program2 :
        cs1713p2.h F 16 blks 4570 bytes
        Input :
            p2Book.txt F 8 blks 947 bytes
            p2Customer.txt F 8 blks 876 bytes
        cs1713p2Stuff.c F 16 blks 7960 bytes
        p2Out.txt F 16 blks 5043 bytes
    Program3 :
        p3Book.txt F 8 blks 949 bytes
```

```
        cs1713p3.h F 16 blks 5037 bytes
        cs1713p3Driver.c F 40 blks 20430 bytes
        p3Command.txt F 8 blks 1598 bytes
    file4 F 8 blks 87 bytes
> xdir Data -r
Data :
    file5
    file6
    Program4 :
        p4Book.txt
        cs1713p4.h
        cs1713p4Driver.c
        p4Command.txt
        p4Commandv2.txt
    Program5 :
        Input5 :
            p5Book.txt
            p5Command.txt
        Makefile
        cs1713p5Driver.c
        cs1713p5.h
    Program2 :
        cs1713p2.h
        Input :
            p2Book.txt
            p2Customer.txt
        cs1713p2Stuff.c
        p2Out.txt
    Program3 :
        p3Book.txt
        cs1713p3.h
        cs1713p3Driver.c
        p3Command.txt
    file4
```