Lab Exercise 2

Due: October 19, 2016 **Points:** 100

This laboratory exercise is designed to teach you about Linux file permissions.

You are to write a program called “access” that will say what rights a user or group has over a file or directory.

Command

The command for your program is the following:

```
access[ -g ] name  file1 ...
```

In this command, *name* is either a name or a non-negative integer. If the `-g` flag is given, *name* is interpreted as a group name or GID, and if that flag is not given, it is interpreted as a user name or UID. It is an error to give a non-existent user or group name, but it is *not* an error to give a UID or GID without an associated name.

Example Output

In these examples, “files” is a file and “dirt” is a directory. Assume “filex” has owner bishop, group owner users, and protection mode 0654, and the directory “dirx” has the same owner and group and the protection mode 0751. Then the command

```
access bishop filex
```

prints

```
The user bishop (UID 7000403) can read and write the file filex
```

The command

```
access sushi filex
```

prints

```
The user sshd (UID 74) can read the file filex
```

and the command

```
access 10 filex
```

prints

```
The user with UID 10 can read the file filex
```

but the command

```
access -g 10 filex
```

prints

```
Members of the group users (GID 10) can read and execute the file filex
```

The difference is that, without the `-g` option, the 10 is interpreted as a UID, and there is no associated name. But with that option, the 10 is interpreted as a GID, and that has the associated name “users”.

If any of the named files are directories, you are to print “list the contents of”, “modify”, or “search” rather than “read”, “write”, and “execute”, respectively. So, the command

```
access bishop dirx
```

prints

```
The user bishop (UID 7000403) can list the contents of, modify, and search the directory
dirx
```

(all on the same line), the command

```
access -g users dirx
```

prints

```
Members of the group users (GID 10) can list the contents of and search the directory dirx
```

and the command

```
access -g cosmos dirx
```

prints

```
Members of the group cosmos (GID 900) can search the directory dirx
```

Checking Your Output

A reference version of this program is available in the directory `bishop/ecs153`; it is the executable file "access". Your output, and especially your error output, is to match that of this program.

Submitting Your Program

You must submit either a tar archive or a compressed tar archive to Canvas, as described in the handout **All About Programs**. Do this as follows:

1. Create a directory called lab2-*yourlastname*, where *yourlastname* is your last name.
2. Copy the source code (*not* the executable!) into that directory.
3. Create a Makefile in that directory. When we test your program, we will change to the directory and type "make". So be sure your Makefile correctly compiles your program on the CSIF!
4. Now create your documentation – for this program, a README saying how to compile it, and what it does, is sufficient.
5. Then create either a tar archive (the archive's name is to end in ".tar") or a compressed tar archive (the compressed archive's name is to end in ".tgz"), and submit that to Canvas.

   That's it!

Regrading

When we grade your program, 40% of the grade will be based on robustness, which includes handling errors and problems gracefully as well as good programming style. If you lose points because of this, we will give you a week to modify your program and resubmit it. We will then regrade *only* the robustness, and add back 75% of the points you regain. So if your score on the robustness part is 20 out of 40, you can get up to 15 of the other 20 points back by fixing your program and resubmitting it.