

Name : Naman Aggarwal
Roll no. : 2021070

Steps to run shell:

- 1) run command make
- 2) Run ./main
- 3) Shell is running

General:

- 1) assumes that the any file name doesn't contains a space
- 2) doing a blank enter and writing exit on the shell will exit the shell

CAT :

Flags implemented : n and e

Usage : cat [-flag] [file ...]

Errors handled:

- 1) Both flags can be used together and are optional
- 2) It will give error if no parameters are passed
- 3) It will give error if wrong path is entered
- 4) It will give error if a directory is entered
- 5) If one path is wrong then entire statement will not be stopped
- 6) It will check for correct flag

Flags:

-n : number all output lines

-e : display \$ at the end of each line

ex: cat -ne file1 file2 || cat -n file1 file2 || cat -e file1 file2 || cat file1 file2

```
namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> cat
No argument entered
Shell> cat -ne
No file or directory entered
Shell> cat -ne testing.txt
1 hello world $
2 we will be back soon$
3 with a new version of the game$
4 stay tuned
Shell> cat -ne testing2.txt testing.txt lol
1 hello$
2 again
1 hello world $
2 we will be back soon$
3 with a new version of the game$
4 stay tuned
cat: lol: No such file or directory
Shell> █
```

DATE:

Flags implemented : u and r

Usage : date [-u] [-r file]

Errors handled:

- 1) Both flags can be used together and are optional
- 2) It will give error if wrong path is entered.
- 3) It will check if proper syntax is used and arguments are passed
- 4) It will check for correct flag

Flags:

-u : print or set Coordinated Universal Time (UTC)

-r : display the last modification time of file or directory

ex: date -ur file1 || date -r file1 || date -u || date

```
namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> date
Wed Oct 26 22:56:30 2022
Shell> date -u
Wed Oct 26 17:26:35 2022
Shell> date -r
Invalid argument
Shell> date -r ls.c
Wed Oct 26 21:43:16 2022
Shell> date -ur ls.c
Wed Oct 26 16:13:16 2022
Shell>
```

LS:

Flags implemented : a and d

Usage : ls [-a] [-d] [path ...]

Errors handled:

- 1) Both flags can be used together and are optional
- 2) It will give error if wrong path is entered
- 3) It will give error if file is entered
- 4) It will check for correct flag

Flags:

-a : do not ignore entries starting with .

-d : list directory entries instead of contents

ex: ls -ad [path ...] || ls -d [path ...] || ls -a [path ...] || ls [path ...]

MKDIR:

Flags implemented : v and m

Usage : mkdir [-v] [-m mode] [file ...]

Errors handled:

- 1) Both flags can be used together and are optional
- 2) It will give error if directory exists

```

namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> ls
path: .
cat    ls.c    makefile    date    testing.txt    rm.c    cat.c    testing2.
txt    mkdir.c  main.c    mkdir    date.c    ls        main    rm        docs.txt
Shell> ls ../
path: ../
naman  sem1    M3        OS        discrete maths  oops    sem2
Shell> ls -da ../../
path: ../../
.      ..      IIITD    langs    DM        wwdc
Shell> ls -ad ../../ /
path: ../../
.      ..      IIITD    langs    DM        wwdc
path: /
.      ..      home     usr      bin       sbin     etc      var      Library System pr
ivate .vol   Users    Applications  opt      dev      Volumes tmp      cores
Shell> ls na
path: na
Directory not found, could be a file
Shell> pwd
/Users/namanaggarwal/Desktop/IIITD/OS
Shell> ls ../../dp
path: ../../dp
Directory not found, could be a file
Shell>

```

- 3) It will give error for wrong mode entered
- 4) It will give error if wrong path is entered
- 5) It will check for correct flag

Flags:

- v : print a message for each created directory
- m : set file mode (as in chmod)

mode is an octal number where each digit represents the permissions for the owner, group and others and the first digit has to be 0

and other subsequent digits can be sum of 1,2,4 (1 for execute, 2 for write and 4 for read)

ex: mkdir -vm [file1 ...] || mkdir -v [file1 ...] || mkdir -m <mode> [file1 ...] || mkdir [file1 ...]

RM:

flags implemented : d and v

usage : rm [-d] [-v] [file ...]

Errors handled:

- 1) Both flags can be used together and are optional
- 2) It will give error if wrong path is entered
- 3) It will give error if trying to delete a directory without -d flag
- 4) It will give error of trying to delete non empty directory
- 5) It will check for correct flag

Flags:

```

[namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> mkdir
No argument entered
Shell> mkdir -m 0777 naman
Shell> mkdir -m 0777 naman
naman: Directory exists
Shell> mkdir -v hello
mkdir: created directory 'hello'
Shell> mkdir -m 777 lol
Invalid mode
Shell> cd
Shell> mkdir naman/lol
Error in creating directory
Shell> mkdir -pl naman
Invalid flag
Shell> mkdir -mv 0512 lol
mkdir: created directory 'lol'
Shell> rm -d lol
Shell> ls
path: .
Music    Pictures      Desktop  Library  iCloud Drive (Archive)  Public  Movies  A
pplications  Documents      Downloads
Shell>

```

-d : remove empty directories

-v : explain what is being done

ex: `rm -vd [file1 ...] || rm -v [file1 ...] || rm -d [file1 ...] || rm [file1 ...]`

```

[namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> rm
No arguments
Shell> rm test1 test2
Shell> rm naman
rm: cannot remove 'naman': Is a directory
Shell> rm -vs naman hello
Invalid flag
Shell> rm -vd naman hello
removed naman
removed hello
Shell> rm world
rm: cannot remove 'world': No such file or directory
Shell> rm -d l
Directory l not empty
Shell>

```

PWD:

Flags implemented : none

usage : `pwd`

Errors handled:

- 1) It will give error if more than one argument is passed
- 2) It is case sensitive

It will print the current working directory
ex: pwd

```
Shell> pwd
/Users/namanaggarwal/Desktop/IIITD/OS
Shell> pwd ma
Too many arguments
Shell> █
```

CD:

Flags implemented : no flags available
Usage : cd [directory]

Errors handled:

- 1) It will give error if more than one argument is passed
- 2) It will give error if wrong path is entered
- 3) Only cd and cd ~ also works

ex: cd [path] || cd ~ || cd

```
Shell> cd l
Shell> cd naman
No such file or directory
Shell> cd l p
Too many arguments
Shell> cd
Shell> pwd
/Users/namanaggarwal
Shell> cd Desktop
Shell> pwd
/Users/namanaggarwal/Desktop
Shell> cd ~
Shell> pwd
/Users/namanaggarwal
Shell> █
```

ECHO:

Flags implemented : n and help
Usage : echo [-n | -help] [string ...]

Errors handled:

- 1) Both flags can not be used together and are optional
- 2) It will give error if wrong flag is entered
- 3) It will work if no arguments are passed and print empty line

-n : do not output the trailing newline

-help : display help

ex: echo -n [string ...] || echo -help || echo [string ...]

```
namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> echo -p
Invalid flag
Shell> echo naman
naman
Shell> echo -n naman
naman Shell>
Exiting the shell
namanaggarwal@namans-MacBook-Air-3 OS % ./main
Welcome to the shell!
Shell> echo -help
Usage: echo [SHORT-OPTION]... [STRING]...
Echo the STRING(s) to standard output.

Mandatory arguments to long options are mandatory for short options too.
  -n      do not output the trailing newline
  --help  display this help and exit
Shell> █
```