TERMINAL DOCUMENTATION

ASSUMPTIONS:

1. The maximum length of the command line can be 1000 characters.
2. Only one independent tag is handled for every common line.
3. The directory length shouldn't be more than 1000 characters.
4. If you want to achieve the functionality of the shell using pthread, then you should add '&t' at the end of the command line.
   Eg: mkdir a b c &t

Commands Available:

1.Internal Commands:
   a. pwd
   b. echo
   c. cd

2. External Commands:
   a. mkdir
   b. ls
   c. cat
   d. date
   e. rm

Internal Commands:

1. Pwd: print the name of the current/working directory. Using the function getcwd().

   a. Without any tags: print the name of the current working directory.

   Command line: `pwd`

   b. Using the '-P' tag: physical, avoid all symlinks.

   Command line: `pwd -P`

   c. Using the '-L' tag: logical uses PWD from the environment, even if it contains symlinks.

   Command line: `pwd -L`

   Error handling:

   a. Alerts if you entered an invalid tag.

   b. Alerts if you entered anything after the pwd command apart from the tags

2. Cd: change the current working directory. Using the function chdir().

   a. Without any tags: change current working directory.

   Command line: `cd [Directory Name]`

   b. Using the '-P' tag: physical, avoid all symlinks.

   Command line: `cd -P [Directory Name]`

   c. Using the '-L' tag: logical use PWD from the environment, even if it contains symlinks.

   Command line: `cd -L [Directory Name]`

   Note: You can also use '..' instead of [Directory Name], which will change the current directory to the parent directory.

   Error handling:

   a. Alerts if didn't enter anything after the cd command.cd

   b. Alerts if its unable to change the directory.

3. Echo: prints the entered string to the command line. Using the function printf().

   a. Without any tags: prints the contents of the command line after the echo word.

   Command line: `echo [String stdin]`

   b. Using the '-n' tag: does not output the trailing newline.

Command line: `echo -n [String stdin]`

c. Using the '-e' tag: enable the interpretation of backslash escapes

Command line: `echo -e [String stdin]`

Error handling:

c. Alerts if didn't enter anything after the echo command

d. Alerts if didn't enter anything after the echo comman.

External Commands:

1. Mkdir: Used to make directories. Using the mkdir() function.
   a. Without any tags: prints the contents of the command line after the echo word.
      Command line: mkdir [Directory Name..1] [Directory Name..2]… [Directory Name..n]
   b. Using the '-p' tags: no error if existing, make parent directories as needed
      Command line: mkdir -p [Directory Name..1]/[Directory Name..2]/[Directory Name..3]
   c. Using the '-v' tags: print a message for each created directory
      Command line: mkdir -v [Directory Name..1] [Directory Name..2] [Directory Name..3]

   Error handling:
   c. Alerts if you are trying to create a directory if it already exists in the current directory.
   d. Alerts if you have not mentioned the name of the directory you want to create,

2. Cat: Used to concatenate files and print on the standard output.
   a. Without any tags: prints the contents of the file on the standard output
      Command line: cat [File Name..1] [File Name..2]… [File Name..n]
   b. Using the '-E' tags: prints a '$' to mark the beginning of the line.
      Command line: cat -E [File Name..1] [File Name..2]… [File Name..n]
   c. Using the '-T tags: displays tab characters as ^I
      Comcatmand line: cat -T [File Name..1] [File Name..2]… [File Name..n]

   Error handling:
   e. Alerts if the file you want to concatenate doesn't open.
   f. Alerts if you havent entered anything after the cat command with or without tag.

3. date: used to get the current date and time. Using the gmttime() function and formatting using ctime() and strftime() function..

      d. Without any tags: prints the local time and date to the std output
         Command line: `date`

      e. Using the '-u' tags: prints the UTC time and date to the std output
         Command line: `date -u`

      f. Using the '-R' tags: print a local data and time in RFC 5322 format
         Command line: `date -R`

Error handling:

      g. Alerts if you have entered anything other than specified tags in the command line.

      h. Alerts if you have entered anything after the specified tags in the command line.

4. ls: used to list the contents of the directory. Using the opendir(), readdir() and struct dirent imported from <dirent.h> library in c.

      g. Without any tags: prints the list of the contents of the directory.
         Command line: `ls`

      h. Using the '-u' tags: prints the list of the contents of the directory but does not list implied . and ..
         Command line: `ls -A`

      i. Using the '-1' tags: prints the list of the contents of the directory one file per line
         Command line: `ls -1`

Error handling:

      i. Alerts if you have entered anything other than specified tags in the command line.

      j. Alerts if you have entered anything after the specified tags in the command line.

5. rm: used to remove the files or directories. Using the rmdir() and remove() function.

    j.   Without any tags: removes the files entered by you

       Command line: `rm [file name1] …`

    k.   Using the '-r' tags: used to remove directories.

       Command line: `rm -r [directory name1] …`

    l.   Using the '-v' tags: prints what you have done, alerts status of deletion of file

       Command line: `rm -v [file name1] …`

Error handling:

    k.   Alerts if you have entered anything other than specified tags in the command line.

    l.   Alerts if the file you are trying to delete a directory without using -r tag..