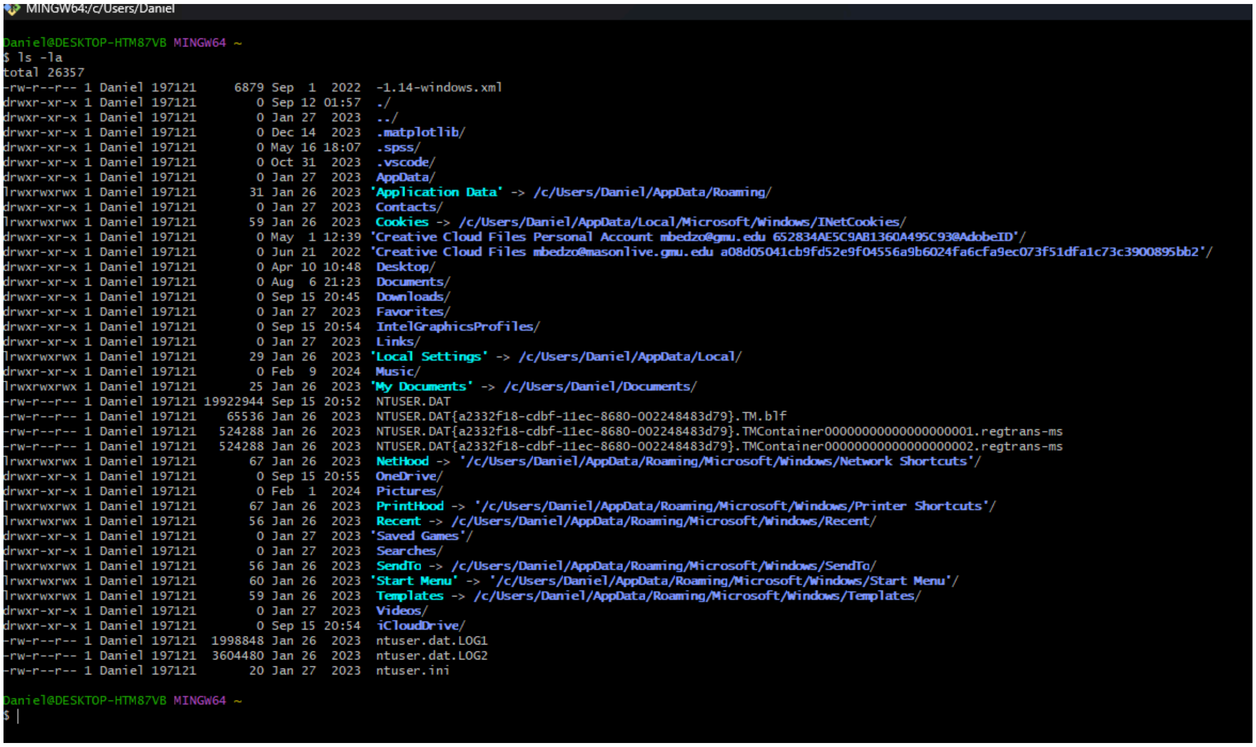
1. What is the difference between shell and bash?

   A shell is an interface that lets its users interact with operating systems by typing commands. Bash, on the other hand, is specific version of a shell, that is most commonly used today and designed as an improved version of Bourne shell.

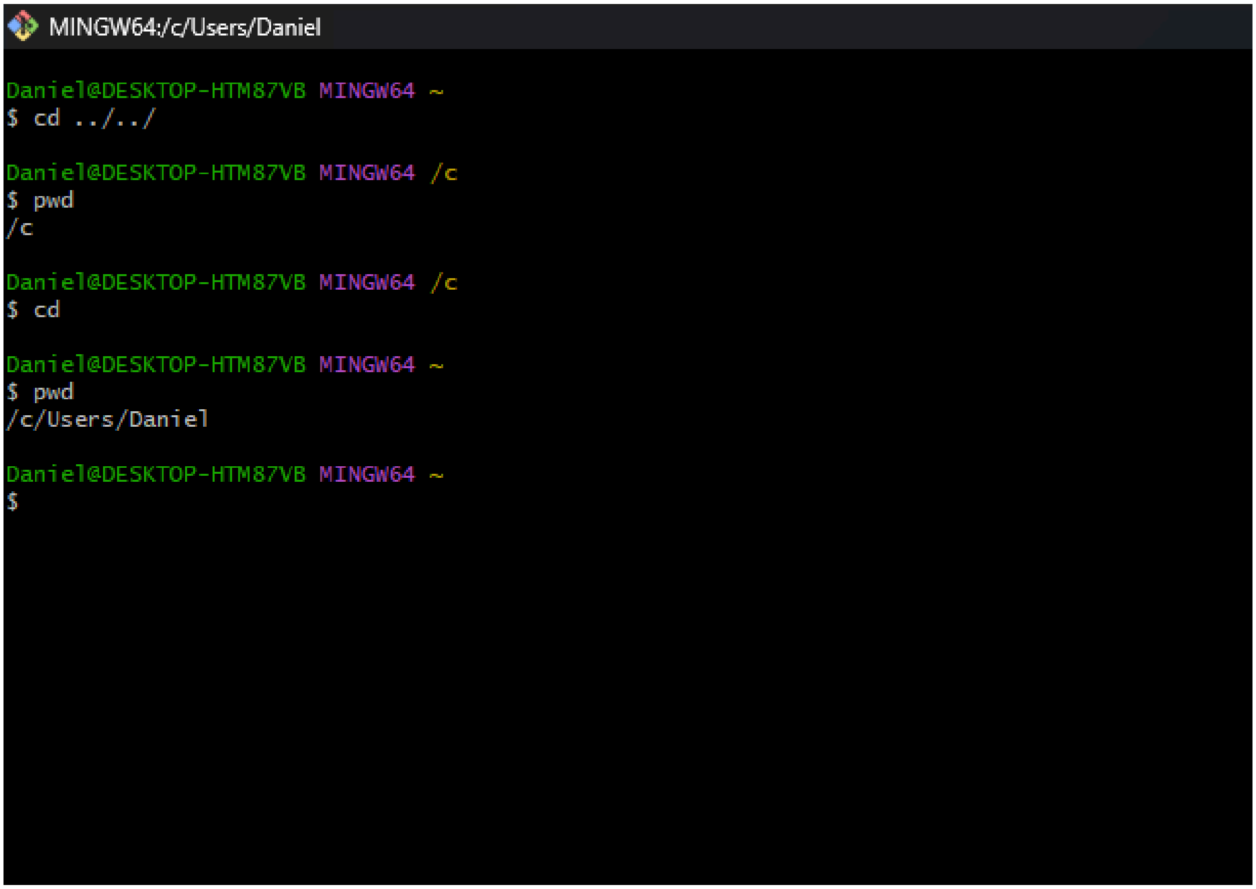
2.To respond to this question, you need to use terminal/Bash and have a screenshot of your terminal/bash.



```
Daniel@DESKTOP-HTM87VB MINGW64 ~
$ ls -la
total 26357
-rw-r--r-- 1 Daniel 197121     6879 Sep  1  2022  -1.14-windows.xml
drwxr-xr-x 1 Daniel 197121        0 Sep 12 01:57  ./
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  ../
drwxr-xr-x 1 Daniel 197121        0 Dec 14  2023  .matplotlib/
drwxr-xr-x 1 Daniel 197121        0 May 16 18:07  .spss/
drwxr-xr-x 1 Daniel 197121        0 Oct 31  2023  .vscode/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  AppData/
lrwxrwxrwx 1 Daniel 197121       31 Jan 26  2023 'Application Data' -> /c/Users/Daniel/AppData/Roaming/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  Contacts/
lrwxrwxrwx 1 Daniel 197121       59 Jan 26  2023  Cookies -> /c/Users/Daniel/AppData/Local/Microsoft/Windows/INetCookies/
drwxr-xr-x 1 Daniel 197121        0 May  1 12:39 'Creative Cloud Files Personal Account mbedzo@gmu.edu 652834AE5C9AB1360A495C93@AdobeID'/
drwxr-xr-x 1 Daniel 197121        0 Jun 21  2022 'Creative Cloud Files mbedzo@masonlive.gmu.edu a08d05041cb9fd52e9f04556a9b6024fa6cfa9ec073f51dfa1c73c3900895bb2'/
drwxr-xr-x 1 Daniel 197121        0 Apr 10 10:48  Desktop/
drwxr-xr-x 1 Daniel 197121        0 Aug  6 21:23  Documents/
drwxr-xr-x 1 Daniel 197121        0 Sep 15 20:45  Downloads/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  Favorites/
drwxr-xr-x 1 Daniel 197121        0 Sep 15 20:54  IntelGraphicsProfiles/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  Links/
lrwxrwxrwx 1 Daniel 197121       29 Jan 26  2023 'Local Settings' -> /c/Users/Daniel/AppData/Local/
drwxr-xr-x 1 Daniel 197121        0 Feb  9  2024  Music/
lrwxrwxrwx 1 Daniel 197121       25 Jan 26  2023 'My Documents' -> /c/Users/Daniel/Documents/
-rw-r--r-- 1 Daniel 197121 19922944 Sep 15 20:52  NTUSER.DAT
-rw-r--r-- 1 Daniel 197121    65536 Jan 26  2023  NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf
-rw-r--r-- 1 Daniel 197121   524288 Jan 26  2023  NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000001.regtrans-ms
-rw-r--r-- 1 Daniel 197121   524288 Jan 26  2023  NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000002.regtrans-ms
lrwxrwxrwx 1 Daniel 197121       67 Jan 26  2023  NetHood -> '/c/Users/Daniel/AppData/Roaming/Microsoft/Windows/Network Shortcuts'/
drwxr-xr-x 1 Daniel 197121        0 Sep 15 20:55  OneDrive/
drwxr-xr-x 1 Daniel 197121        0 Feb  1  2024  Pictures/
lrwxrwxrwx 1 Daniel 197121       67 Jan 26  2023  PrintHood -> '/c/Users/Daniel/AppData/Roaming/Microsoft/Windows/Printer Shortcuts'/
lrwxrwxrwx 1 Daniel 197121       56 Jan 26  2023  Recent -> /c/Users/Daniel/AppData/Roaming/Microsoft/Windows/Recent/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023 'Saved Games'/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  Searches/
lrwxrwxrwx 1 Daniel 197121       56 Jan 26  2023  SendTo -> /c/Users/Daniel/AppData/Roaming/Microsoft/Windows/SendTo/
lrwxrwxrwx 1 Daniel 197121       60 Jan 26  2023 'Start Menu' -> '/c/Users/Daniel/AppData/Roaming/Microsoft/Windows/Start Menu'/
lrwxrwxrwx 1 Daniel 197121       59 Jan 26  2023  Templates -> /c/Users/Daniel/AppData/Roaming/Microsoft/Windows/Templates/
drwxr-xr-x 1 Daniel 197121        0 Jan 27  2023  Videos/
drwxr-xr-x 1 Daniel 197121        0 Sep 15 20:54  iCloudDrive/
-rw-r--r-- 1 Daniel 197121  1998848 Jan 26  2023  ntuser.dat.LOG1
-rw-r--r-- 1 Daniel 197121  3604480 Jan 26  2023  ntuser.dat.LOG2
-rw-r--r-- 1 Daniel 197121       20 Jan 27  2023  ntuser.ini

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ |
```

A. The home directory is /c/Users/Daniel
B. Files/Folders include Desktop, Documents, Downloads, and Music

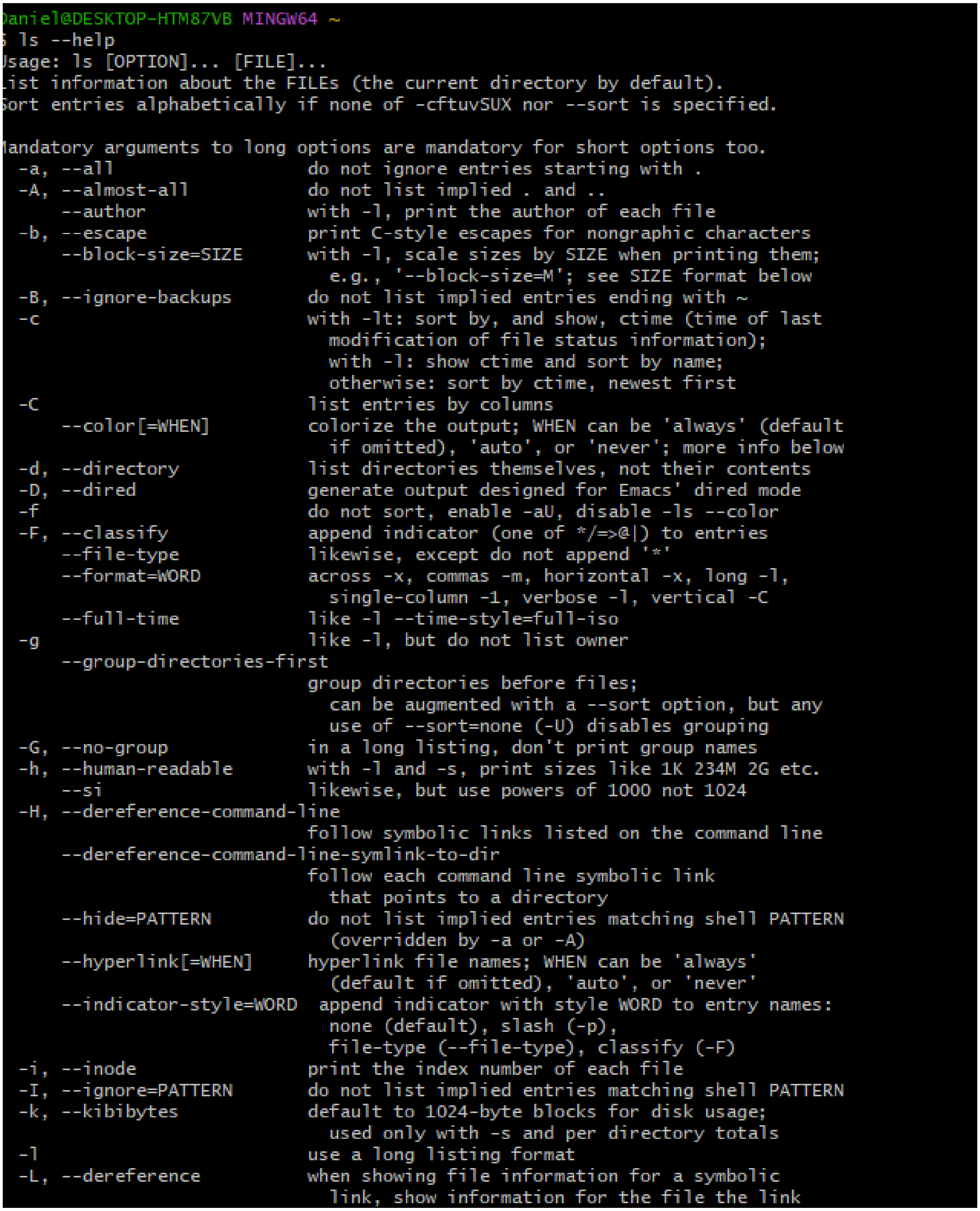
3. To respond to this question, you need to use terminal/Bash and have a screenshot of your terminal/bash.



```
Daniel@DESKTOP-HTM87VB MINGW64 ~
$ cd ../../

Daniel@DESKTOP-HTM87VB MINGW64 /c
$ pwd
/c

Daniel@DESKTOP-HTM87VB MINGW64 /c
$ cd

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ pwd
/c/Users/Daniel

Daniel@DESKTOP-HTM87VB MINGW64 ~
$
```

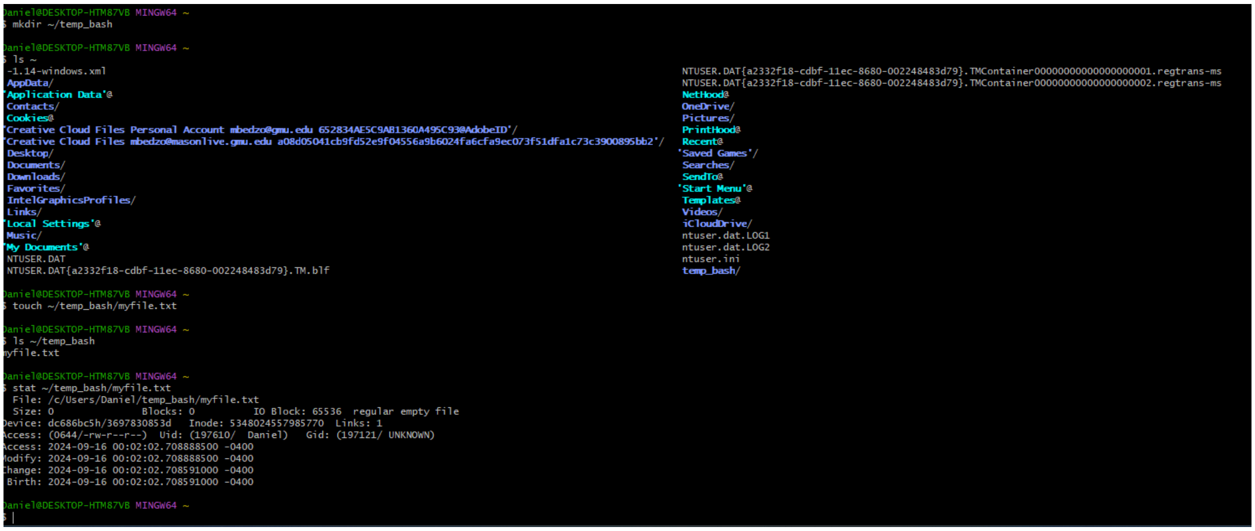
A. '/c', Two levels up from /c/Users/Daniel
B. the command `cd` returns you to the home directory, in this case, /c/Users/Daniel

4. The -a flag (or --all) instructs ls to display all files, including hidden ones that begin with a dot .). The -l: lists files in long format and provides detailed information for them



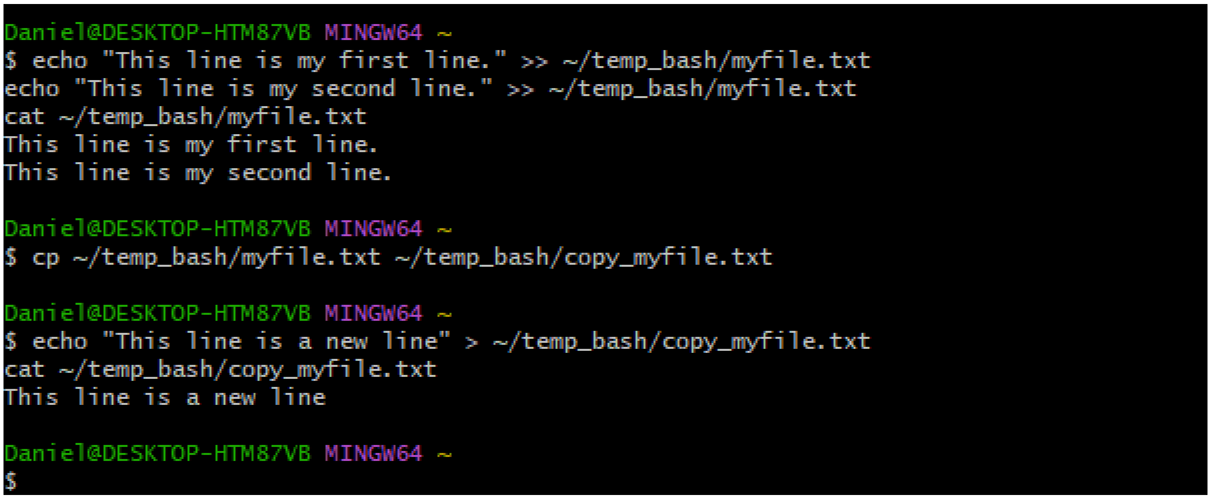
```
Daniel@DESKTOP-HTM87VB MINGW64 ~
$ ls --help
Usage: ls [OPTION]... [FILE]...
List information about the FILEs (the current directory by default).
Sort entries alphabetically if none of -cftuvSUX nor --sort is specified.

Mandatory arguments to long options are mandatory for short options too.
  -a, --all                  do not ignore entries starting with .
  -A, --almost-all           do not list implied . and ..
      --author               with -l, print the author of each file
  -b, --escape               print C-style escapes for nongraphic characters
      --block-size=SIZE      with -l, scale sizes by SIZE when printing them;
                               e.g., '--block-size=M'; see SIZE format below
  -B, --ignore-backups       do not list implied entries ending with ~
  -c                         with -lt: sort by, and show, ctime (time of last
                               modification of file status information);
                               with -l: show ctime and sort by name;
                               otherwise: sort by ctime, newest first
  -C                         list entries by columns
      --color[=WHEN]         colorize the output; WHEN can be 'always' (default
                               if omitted), 'auto', or 'never'; more info below
  -d, --directory            list directories themselves, not their contents
  -D, --dired                generate output designed for Emacs' dired mode
  -f                         do not sort, enable -aU, disable -ls --color
  -F, --classify             append indicator (one of */=>@|) to entries
      --file-type            likewise, except do not append '*'
      --format=WORD          across -x, commas -m, horizontal -x, long -l,
                               single-column -1, verbose -l, vertical -C
      --full-time            like -l --time-style=full-iso
  -g                         like -l, but do not list owner
      --group-directories-first
                             group directories before files;
                               can be augmented with a --sort option, but any
                               use of --sort=none (-U) disables grouping
  -G, --no-group             in a long listing, don't print group names
  -h, --human-readable       with -l and -s, print sizes like 1K 234M 2G etc.
      --si                   likewise, but use powers of 1000 not 1024
  -H, --dereference-command-line
                             follow symbolic links listed on the command line
      --dereference-command-line-symlink-to-dir
                             follow each command line symbolic link
                               that points to a directory
      --hide=PATTERN         do not list implied entries matching shell PATTERN
                               (overridden by -a or -A)
      --hyperlink[=WHEN]     hyperlink file names; WHEN can be 'always'
                               (default if omitted), 'auto', or 'never'
      --indicator-style=WORD  append indicator with style WORD to entry names:
                               none (default), slash (-p),
                               file-type (--file-type), classify (-F)
  -i, --inode                print the index number of each file
  -I, --ignore=PATTERN       do not list implied entries matching shell PATTERN
  -k, --kibibytes            default to 1024-byte blocks for disk usage;
                               used only with -s and per directory totals
  -l                         use a long listing format
  -L, --dereference          when showing file information for a symbolic
                               link, show information for the file the link
```

5. Question 5: (A and B each have 5 points, and C has 10 points. The total is 20 points.) To respond to this question, you need to use terminal/Bash and have a screenshot of your terminal/bash.



```
Daniel@DESKTOP-HTM87VB MINGW64 ~
$ mkdir ~/temp_bash

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ ls ~
 -1.14-windows.xml                                                                                                  NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000001.regtrans-ms
 AppData/                                                                                                           NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TMContainer00000000000000000002.regtrans-ms
'Application Data'@                                                                                                 NetHood@
 Contacts/                                                                                                          OneDrive/
 Cookies@                                                                                                           Pictures/
'Creative Cloud Files Personal Account mbedzo@gmu.edu 652834AE5C9AB1360A495C93@AdobeID'/                            PrintHood@
'Creative Cloud Files mbedzo@masonlive.gmu.edu a08d05041cb9fd52e9f04556a9b6024fa6cfa9ec073f51dfa1c73c3900895bb2'/   Recent@
 Desktop/                                                                                                          'Saved Games'/
 Documents/                                                                                                         Searches/
 Downloads/                                                                                                         SendTo@
 Favorites/                                                                                                        'Start Menu'@
 IntelGraphicsProfiles/                                                                                             Templates@
 Links/                                                                                                             Videos/
'Local Settings'@                                                                                                   iCloudDrive/
 Music/                                                                                                             ntuser.dat.LOG1
'My Documents'@                                                                                                     ntuser.dat.LOG2
 NTUSER.DAT                                                                                                         ntuser.ini
 NTUSER.DAT{a2332f18-cdbf-11ec-8680-002248483d79}.TM.blf                                                            temp_bash/

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ touch ~/temp_bash/myfile.txt

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ ls ~/temp_bash
myfile.txt

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ stat ~/temp_bash/myfile.txt
  File: /c/Users/Daniel/temp_bash/myfile.txt
  Size: 0               Blocks: 0          IO Block: 65536  regular empty file
Device: dc686bc5h/3697830853d   Inode: 5348024557985770  Links: 1
Access: (0644/-rw-r--r--)  Uid: (197610/  Daniel)   Gid: (197121/ UNKNOWN)
Access: 2024-09-16 00:02:02.708888500 -0400
Modify: 2024-09-16 00:02:02.708888500 -0400
Change: 2024-09-16 00:02:02.708591000 -0400
 Birth: 2024-09-16 00:02:02.708591000 -0400

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ |
```

Question 6: (40 Points) To respond to this question, you need to use terminal/Bash and have a screenshot of your terminal/bash.



```
Daniel@DESKTOP-HTM87VB MINGW64 ~
$ echo "This line is my first line." >> ~/temp_bash/myfile.txt
echo "This line is my second line." >> ~/temp_bash/myfile.txt
cat ~/temp_bash/myfile.txt
This line is my first line.
This line is my second line.

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ cp ~/temp_bash/myfile.txt ~/temp_bash/copy_myfile.txt

Daniel@DESKTOP-HTM87VB MINGW64 ~
$ echo "This line is a new line" > ~/temp_bash/copy_myfile.txt
cat ~/temp_bash/copy_myfile.txt
This line is a new line

Daniel@DESKTOP-HTM87VB MINGW64 ~
$
```

Answer: With ‘>>’ (Append Redirection), it adds (or appends) the output to the end of the file, and if the file doesn’t exist, it will be created. Additionally, the previous contents of the file remain intact.
In contrast, ‘>’ (Overwrite Redirection), overwrites the file with the new content. Put simply, if the file already exists, it must be cleared before the new content is written.Furthermore, if said file doesn’t exist, it will be created.