Operating System (CS 402)

Lab Report Submitted to

Indian Institute of Information Technology Surat

for

Bachelor of Technology

In

Computer Science and Engineering Department

Submitted by

JITANSHU J. RAUT UI21CS27

Course Faculty

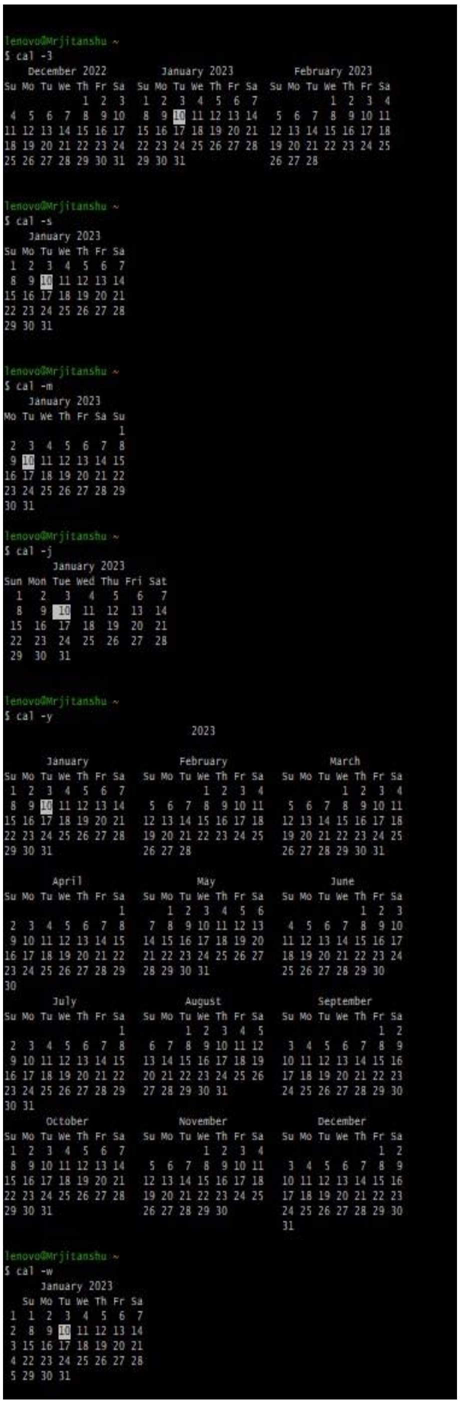
Shraddha Patel

Department of Computer Science and Engineering
Indian Institute of Information Technology Surat
Gujarat-394190, India

2023

Table of Contents

Assignment 1



```
lenovo@Mrjitanshu ~
$ cal -3
   December 2022          January 2023          February 2023
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
             1  2  3   1  2  3  4  5  6  7            1  2  3  4
 4  5  6  7  8  9 10   8  9 10 11 12 13 14   5  6  7  8  9 10 11
11 12 13 14 15 16 17  15 16 17 18 19 20 21  12 13 14 15 16 17 18
18 19 20 21 22 23 24  22 23 24 25 26 27 28  19 20 21 22 23 24 25
25 26 27 28 29 30 31  29 30 31              26 27 28

lenovo@Mrjitanshu ~
$ cal -s
    January 2023
Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7
 8  9 10 11 12 13 14
15 16 17 18 19 20 21
22 23 24 25 26 27 28
29 30 31

lenovo@Mrjitanshu ~
$ cal -m
    January 2023
Mo Tu We Th Fr Sa Su
                   1
 2  3  4  5  6  7  8
 9 10 11 12 13 14 15
16 17 18 19 20 21 22
23 24 25 26 27 28 29
30 31

lenovo@Mrjitanshu ~
$ cal -j
        January 2023
Sun Mon Tue Wed Thu Fri Sat
  1   2   3   4   5   6   7
  8   9  10  11  12  13  14
 15  16  17  18  19  20  21
 22  23  24  25  26  27  28
 29  30  31

lenovo@Mrjitanshu ~
$ cal -y
                              2023

      January               February                March
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4            1  2  3  4
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   5  6  7  8  9 10 11
15 16 17 18 19 20 21  12 13 14 15 16 17 18  12 13 14 15 16 17 18
22 23 24 25 26 27 28  19 20 21 22 23 24 25  19 20 21 22 23 24 25
29 30 31              26 27 28              26 27 28 29 30 31

       April                  May                   June
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1      1  2  3  4  5  6               1  2  3
 2  3  4  5  6  7  8   7  8  9 10 11 12 13   4  5  6  7  8  9 10
 9 10 11 12 13 14 15  14 15 16 17 18 19 20  11 12 13 14 15 16 17
16 17 18 19 20 21 22  21 22 23 24 25 26 27  18 19 20 21 22 23 24
23 24 25 26 27 28 29  28 29 30 31           25 26 27 28 29 30
30
        July                 August              September
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
                   1         1  2  3  4  5                  1  2
 2  3  4  5  6  7  8   6  7  8  9 10 11 12   3  4  5  6  7  8  9
 9 10 11 12 13 14 15  13 14 15 16 17 18 19  10 11 12 13 14 15 16
16 17 18 19 20 21 22  20 21 22 23 24 25 26  17 18 19 20 21 22 23
23 24 25 26 27 28 29  27 28 29 30 31        24 25 26 27 28 29 30
30 31
      October               November              December
Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa  Su Mo Tu We Th Fr Sa
 1  2  3  4  5  6  7            1  2  3  4                  1  2
 8  9 10 11 12 13 14   5  6  7  8  9 10 11   3  4  5  6  7  8  9
15 16 17 18 19 20 21  12 13 14 15 16 17 18  10 11 12 13 14 15 16
22 23 24 25 26 27 28  19 20 21 22 23 24 25  17 18 19 20 21 22 23
29 30 31              26 27 28 29 30        24 25 26 27 28 29 30
                                            31

lenovo@Mrjitanshu ~
$ cal -w
       January 2023
   Su Mo Tu We Th Fr Sa
 1  1  2  3  4  5  6  7
 2  8  9 10 11 12 13 14
 3 15 16 17 18 19 20 21
 4 22 23 24 25 26 27 28
 5 29 30 31
```

CAL

Option Use

-1 Display single (current) month output. (Thisisthe default.)

-3 Display prev/current/next month output

-s Display Sunday as the first day of the week (This is thedefault.)

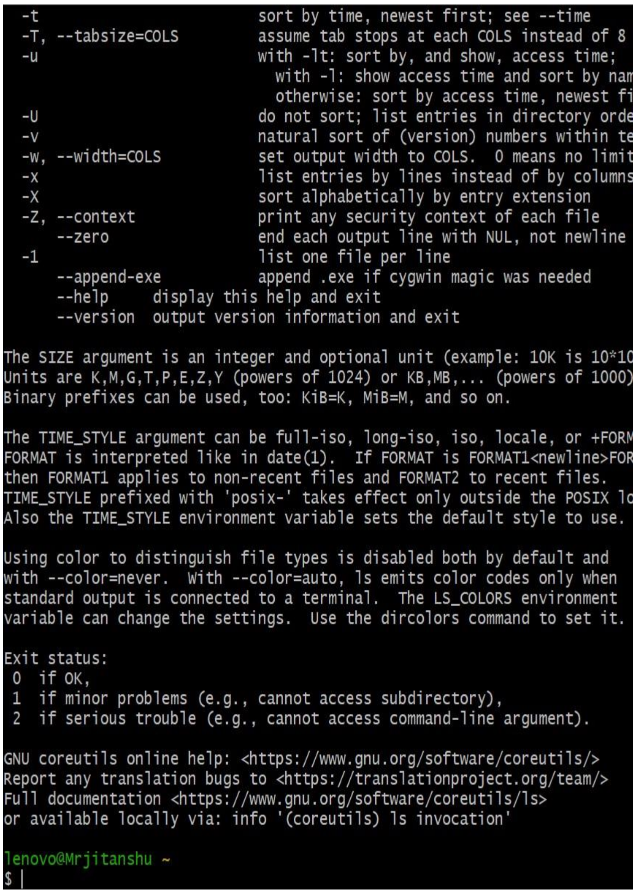
-m Display Monday as the first day of the week

-j Display Julian dates (days one-based, numbered fromJanuary 1)

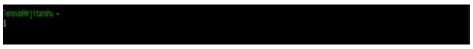
-y Display a calendar for the current year

-w Print the number of the week under each week column

Clear

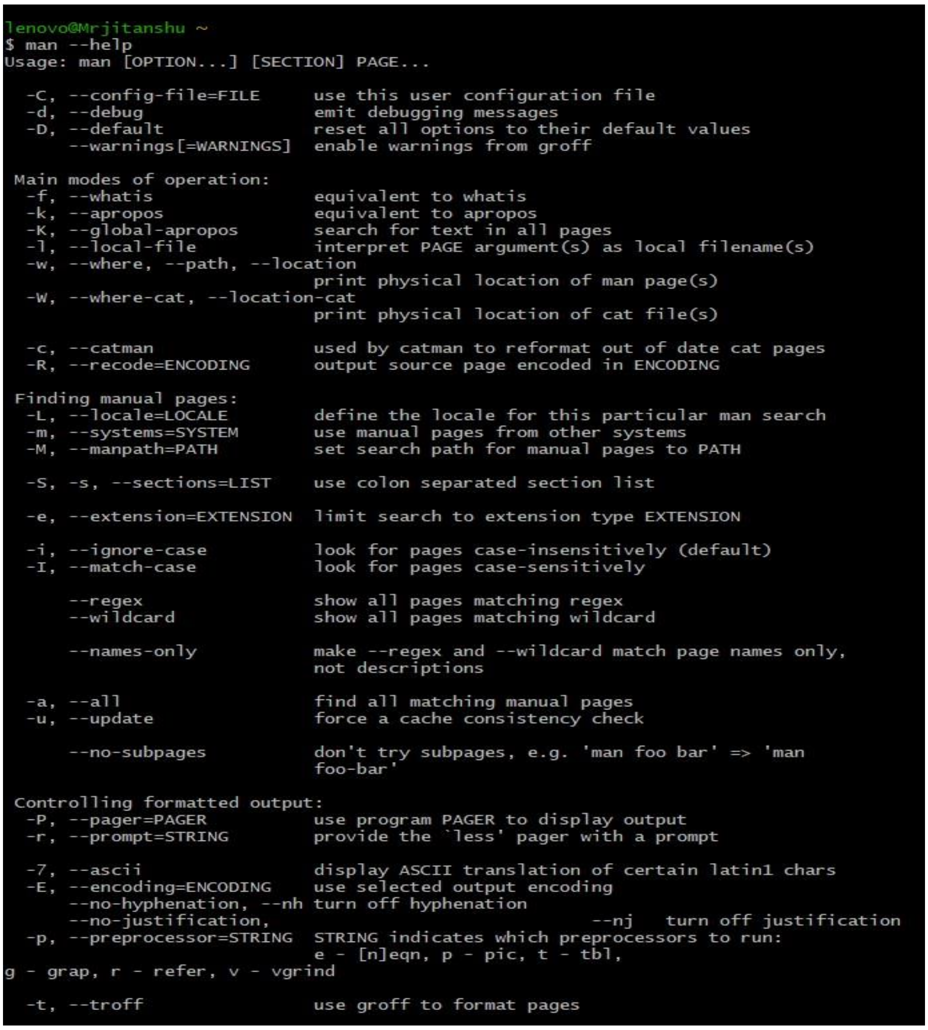


```
  -t                         sort by time, newest first; see --time
  -T, --tabsize=COLS         assume tab stops at each COLS instead of 8
  -u                         with -lt: sort by, and show, access time;
                               with -l: show access time and sort by nam
                               otherwise: sort by access time, newest fi
  -U                         do not sort; list entries in directory orde
  -v                         natural sort of (version) numbers within te
  -w, --width=COLS           set output width to COLS.  0 means no limit
  -x                         list entries by lines instead of by columns
  -X                         sort alphabetically by entry extension
  -Z, --context              print any security context of each file
      --zero                 end each output line with NUL, not newline
  -1                         list one file per line
      --append-exe           append .exe if cygwin magic was needed
      --help     display this help and exit
      --version  output version information and exit

The SIZE argument is an integer and optional unit (example: 10K is 10*10
Units are K,M,G,T,P,E,Z,Y (powers of 1024) or KB,MB,... (powers of 1000)
Binary prefixes can be used, too: KiB=K, MiB=M, and so on.

The TIME_STYLE argument can be full-iso, long-iso, iso, locale, or +FORM
FORMAT is interpreted like in date(1).  If FORMAT is FORMAT1<newline>FOR
then FORMAT1 applies to non-recent files and FORMAT2 to recent files.
TIME_STYLE prefixed with 'posix-' takes effect only outside the POSIX lo
Also the TIME_STYLE environment variable sets the default style to use.

Using color to distinguish file types is disabled both by default and
with --color=never.  With --color=auto, ls emits color codes only when
standard output is connected to a terminal.  The LS_COLORS environment
variable can change the settings.  Use the dircolors command to set it.

Exit status:
 0  if OK,
 1  if minor problems (e.g., cannot access subdirectory),
 2  if serious trouble (e.g., cannot access command-line argument).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/ls>
or available locally via: info '(coreutils) ls invocation'

lenovo@Mrjitanshu ~
$ |
```

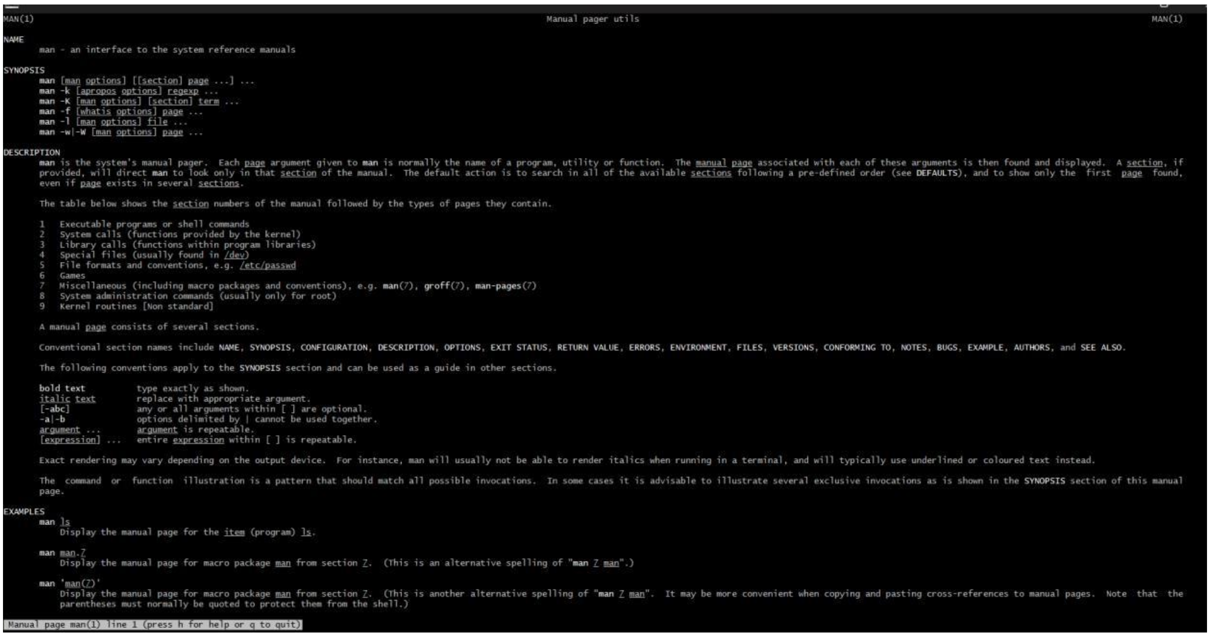


```
lenovo@Mrjitanshu ~
$
```

Man

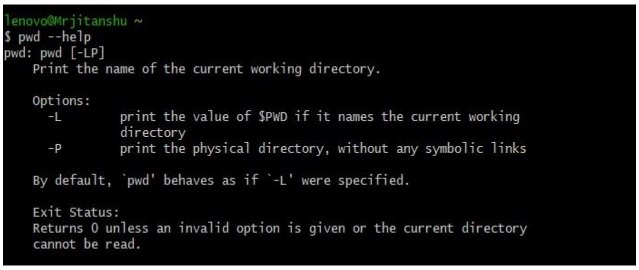


```
lenovo@Mrjitanshu ~
$ man --help
Usage: man [OPTION...] [SECTION] PAGE...

  -C, --config-file=FILE     use this user configuration file
  -d, --debug                emit debugging messages
  -D, --default              reset all options to their default values
      --warnings[=WARNINGS]  enable warnings from groff

 Main modes of operation:
  -f, --whatis               equivalent to whatis
  -k, --apropos              equivalent to apropos
  -K, --global-apropos       search for text in all pages
  -l, --local-file           interpret PAGE argument(s) as local filename(s)
  -w, --where, --path, --location
                             print physical location of man page(s)
  -W, --where-cat, --location-cat
                             print physical location of cat file(s)

  -c, --catman               used by catman to reformat out of date cat pages
  -R, --recode=ENCODING      output source page encoded in ENCODING

 Finding manual pages:
  -L, --locale=LOCALE        define the locale for this particular man search
  -m, --systems=SYSTEM       use manual pages from other systems
  -M, --manpath=PATH         set search path for manual pages to PATH

  -S, -s, --sections=LIST    use colon separated section list

  -e, --extension=EXTENSION  limit search to extension type EXTENSION

  -i, --ignore-case          look for pages case-insensitively (default)
  -I, --match-case           look for pages case-sensitively

      --regex                show all pages matching regex
      --wildcard             show all pages matching wildcard

      --names-only           make --regex and --wildcard match page names only,
                             not descriptions

  -a, --all                  find all matching manual pages
  -u, --update               force a cache consistency check

      --no-subpages          don't try subpages, e.g. 'man foo bar' => 'man
                             foo-bar'

 Controlling formatted output:
  -P, --pager=PAGER          use program PAGER to display output
  -r, --prompt=STRING        provide the `less' pager with a prompt

  -7, --ascii                display ASCII translation of certain latin1 chars
  -E, --encoding=ENCODING    use selected output encoding
      --no-hyphenation, --nh turn off hyphenation
      --no-justification,                          --nj   turn off justification
  -p, --preprocessor=STRING  STRING indicates which preprocessors to run:
                             e - [n]eqn, p - pic, t - tbl,
g - grap, r - refer, v - vgrind

  -t, --troff                use groff to format pages
```

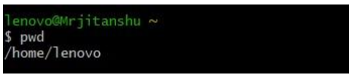


```
MAN(1)                                   Manual pager utils                                   MAN(1)

NAME
       man - an interface to the system reference manuals

SYNOPSIS
       man [man options] [[section] page ...] ...
       man -k [apropos options] regexp ...
       man -K [man options] [section] term ...
       man -f [whatis options] page ...
       man -l [man options] file ...
       man -w|-W [man options] page ...

DESCRIPTION
       man is the system's manual pager.  Each page argument given to man is normally the name of a program, utility or function.  The manual page associated with each of these arguments is then found and displayed.  A section, if
       provided, will direct man to look only in that section of the manual.  The default action is to search in all of the available sections following a pre-defined order (see DEFAULTS), and to show only the  first  page  found,
       even if page exists in several sections.

       The table below shows the section numbers of the manual followed by the types of pages they contain.

       1   Executable programs or shell commands
       2   System calls (functions provided by the kernel)
       3   Library calls (functions within program libraries)
       4   Special files (usually found in /dev)
       5   File formats and conventions, e.g. /etc/passwd
       6   Games
       7   Miscellaneous (including macro packages and conventions), e.g. man(7), groff(7), man-pages(7)
       8   System administration commands (usually only for root)
       9   Kernel routines [Non standard]

       A manual page consists of several sections.

       Conventional section names include NAME, SYNOPSIS, CONFIGURATION, DESCRIPTION, OPTIONS, EXIT STATUS, RETURN VALUE, ERRORS, ENVIRONMENT, FILES, VERSIONS, CONFORMING TO, NOTES, BUGS, EXAMPLE, AUTHORS, and SEE ALSO.

       The following conventions apply to the SYNOPSIS section and can be used as a guide in other sections.

       bold text          type exactly as shown.
       italic text        replace with appropriate argument.
       [-abc]             any or all arguments within [ ] are optional.
       -a|-b              options delimited by | cannot be used together.
       argument ...       argument is repeatable.
       [expression] ...   entire expression within [ ] is repeatable.

       Exact rendering may vary depending on the output device.  For instance, man will usually not be able to render italics when running in a terminal, and will typically use underlined or coloured text instead.

       The  command  or  function  illustration is a pattern that should match all possible invocations.  In some cases it is advisable to illustrate several exclusive invocations as is shown in the SYNOPSIS section of this manual
       page.

EXAMPLES
       man ls
           Display the manual page for the item (program) ls.

       man man.7
           Display the manual page for macro package man from section 7.  (This is an alternative spelling of "man 7 man".)

       man 'man(7)'
           Display the manual page for macro package man from section 7.  (This is another alternative spelling of "man 7 man".  It may be more convenient when copying and pasting cross-references to manual pages.  Note  that  the
           parentheses must normally be quoted to protect them from the shell.)

 Manual page man(1) line 1 (press h for help or q to quit)
```

Pwd

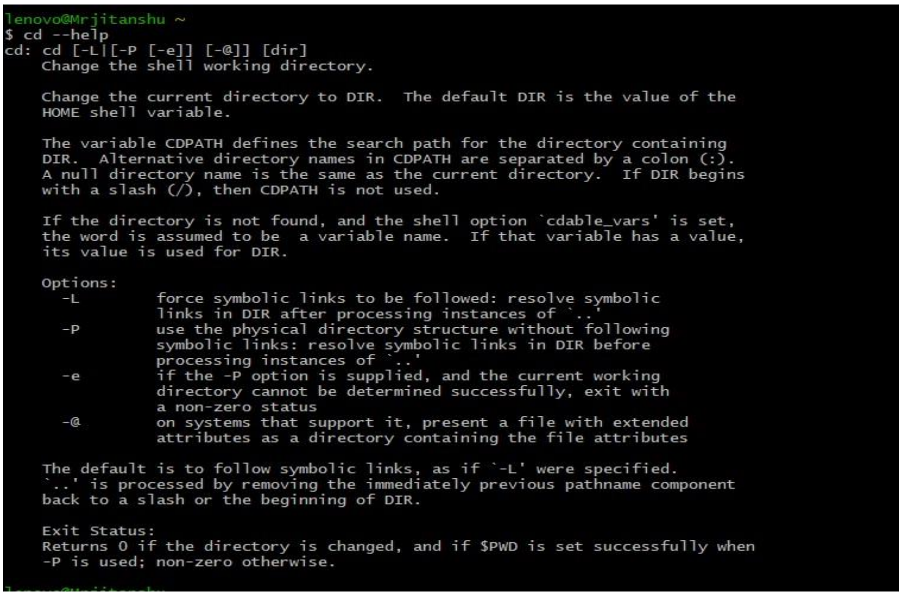


```
lenovo@Mrjitanshu ~
$ pwd --help
pwd: pwd [-LP]
    Print the name of the current working directory.

    Options:
      -L        print the value of $PWD if it names the current working
                directory
      -P        print the physical directory, without any symbolic links

    By default, `pwd' behaves as if `-L' were specified.

    Exit Status:
    Returns 0 unless an invalid option is given or the current directory
    cannot be read.
```

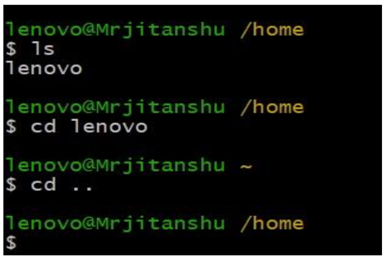


```
lenovo@Mrjitanshu ~
$ pwd
/home/lenovo
```

Cd

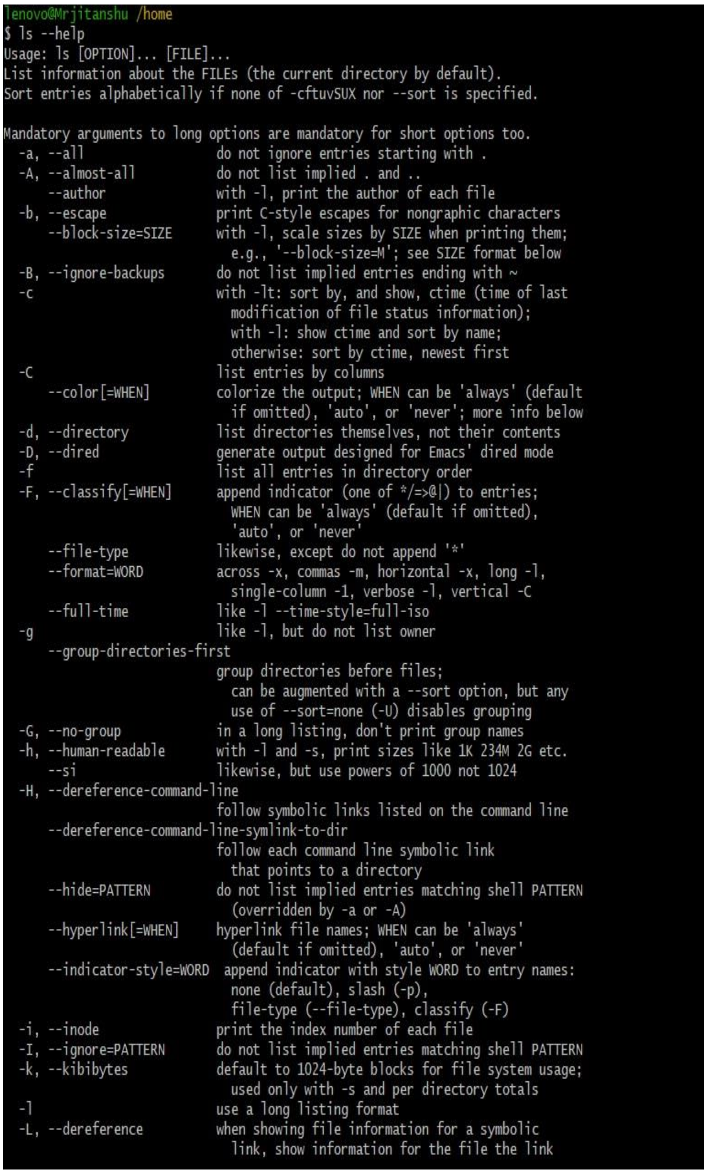


```
lenovo@Mrjitanshu ~
$ cd --help
cd: cd [-L|[-P [-e]] [-@]] [dir]
    Change the shell working directory.

    Change the current directory to DIR.  The default DIR is the value of the
    HOME shell variable.

    The variable CDPATH defines the search path for the directory containing
    DIR.  Alternative directory names in CDPATH are separated by a colon (:).
    A null directory name is the same as the current directory.  If DIR begins
    with a slash (/), then CDPATH is not used.

    If the directory is not found, and the shell option `cdable_vars' is set,
    the word is assumed to be  a variable name.  If that variable has a value,
    its value is used for DIR.

    Options:
      -L        force symbolic links to be followed: resolve symbolic
                links in DIR after processing instances of `..'
      -P        use the physical directory structure without following
                symbolic links: resolve symbolic links in DIR before
                processing instances of `..'
      -e        if the -P option is supplied, and the current working
                directory cannot be determined successfully, exit with
                a non-zero status
      -@        on systems that support it, present a file with extended
                attributes as a directory containing the file attributes

    The default is to follow symbolic links, as if `-L' were specified.
    `..' is processed by removing the immediately previous pathname component
    back to a slash or the beginning of DIR.

    Exit Status:
    Returns 0 if the directory is changed, and if $PWD is set successfully when
    -P is used; non-zero otherwise.
```

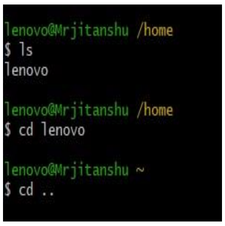


```
lenovo@Mrjitanshu /home
$ ls
lenovo

lenovo@Mrjitanshu /home
$ cd lenovo

lenovo@Mrjitanshu ~
$ cd ..

lenovo@Mrjitanshu /home
$
```

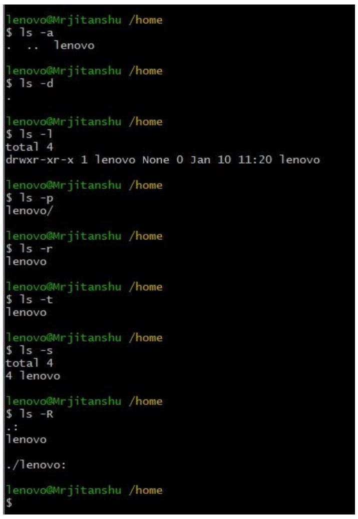
Ls



```
lenovo@Mrjitanshu /home
$ 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                         list all entries in directory order
  -F, --classify[=WHEN]      append indicator (one of */=>@|) to entries;
                               WHEN can be 'always' (default if omitted),
                               'auto', or 'never'
      --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 file system 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
```



```
lenovo@Mrjitanshu /home
$ ls
lenovo

lenovo@Mrjitanshu /home
$ cd lenovo

lenovo@Mrjitanshu ~
$ cd ..
```



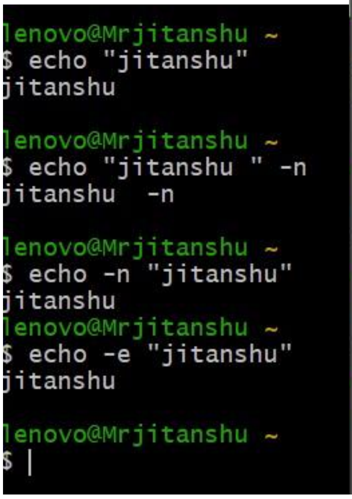
```
lenovo@Mrjitanshu /home
$ ls -a
.  ..  lenovo

lenovo@Mrjitanshu /home
$ ls -d
.

lenovo@Mrjitanshu /home
$ ls -l
total 4
drwxr-xr-x 1 lenovo None 0 Jan 10 11:20 lenovo

lenovo@Mrjitanshu /home
$ ls -p
lenovo/

lenovo@Mrjitanshu /home
$ ls -r
lenovo

lenovo@Mrjitanshu /home
$ ls -t
lenovo

lenovo@Mrjitanshu /home
$ ls -s
total 4
4 lenovo

lenovo@Mrjitanshu /home
$ ls -R
.:
lenovo

./lenovo:

lenovo@Mrjitanshu /home
$
```

Exit

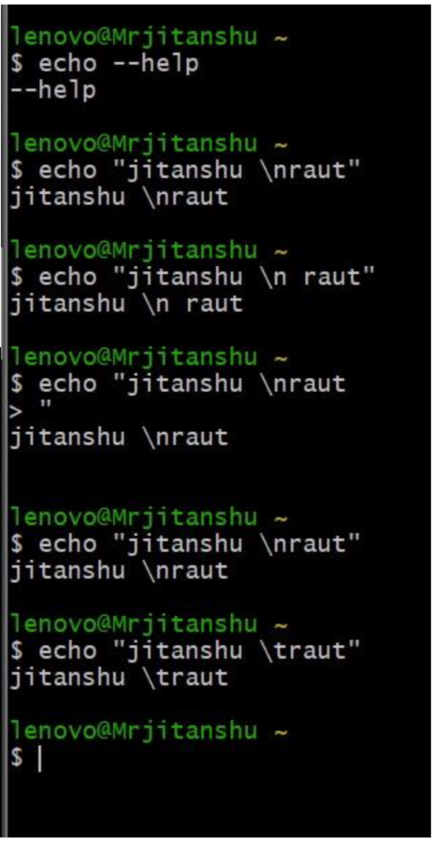
exit :- It is used to terminate a program, shell or log you out of anetwork normally.

Syntax: - exit

Echo

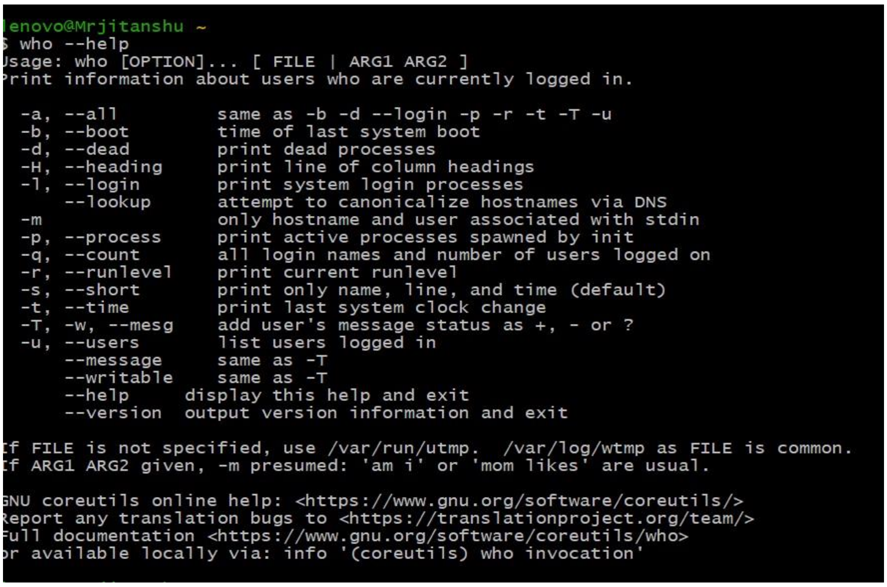


```
lenovo@Mrjitanshu ~
$ echo "jitanshu"
jitanshu

lenovo@Mrjitanshu ~
$ echo "jitanshu " -n
jitanshu  -n

lenovo@Mrjitanshu ~
$ echo -n "jitanshu"
jitanshu
lenovo@Mrjitanshu ~
$ echo -e "jitanshu"
jitanshu

lenovo@Mrjitanshu ~
$ |
```



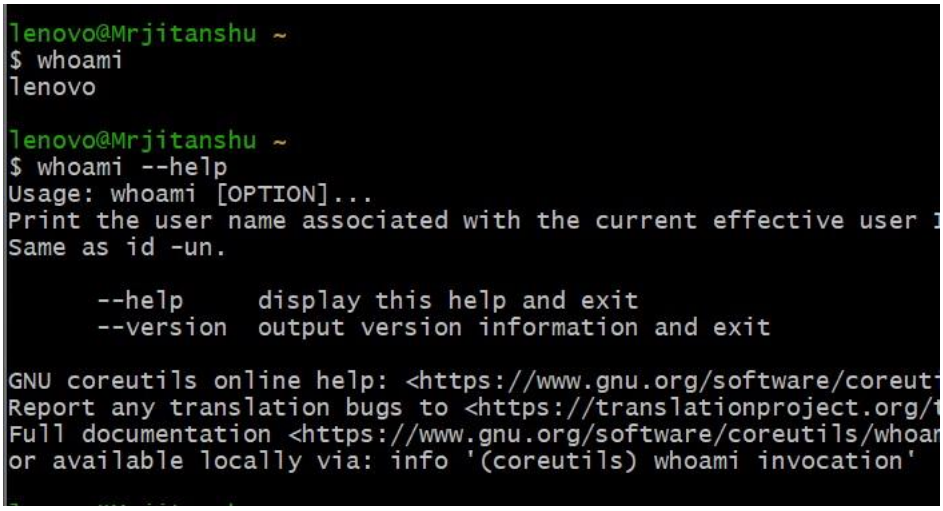
```
lenovo@Mrjitanshu ~
$ echo --help
--help

lenovo@Mrjitanshu ~
$ echo "jitanshu \nraut"
jitanshu \nraut

lenovo@Mrjitanshu ~
$ echo "jitanshu \n raut"
jitanshu \n raut

lenovo@Mrjitanshu ~
$ echo "jitanshu \nraut
> "
jitanshu \nraut


lenovo@Mrjitanshu ~
$ echo "jitanshu \nraut"
jitanshu \nraut

lenovo@Mrjitanshu ~
$ echo "jitanshu \traut"
jitanshu \traut

lenovo@Mrjitanshu ~
$ |
```

Who



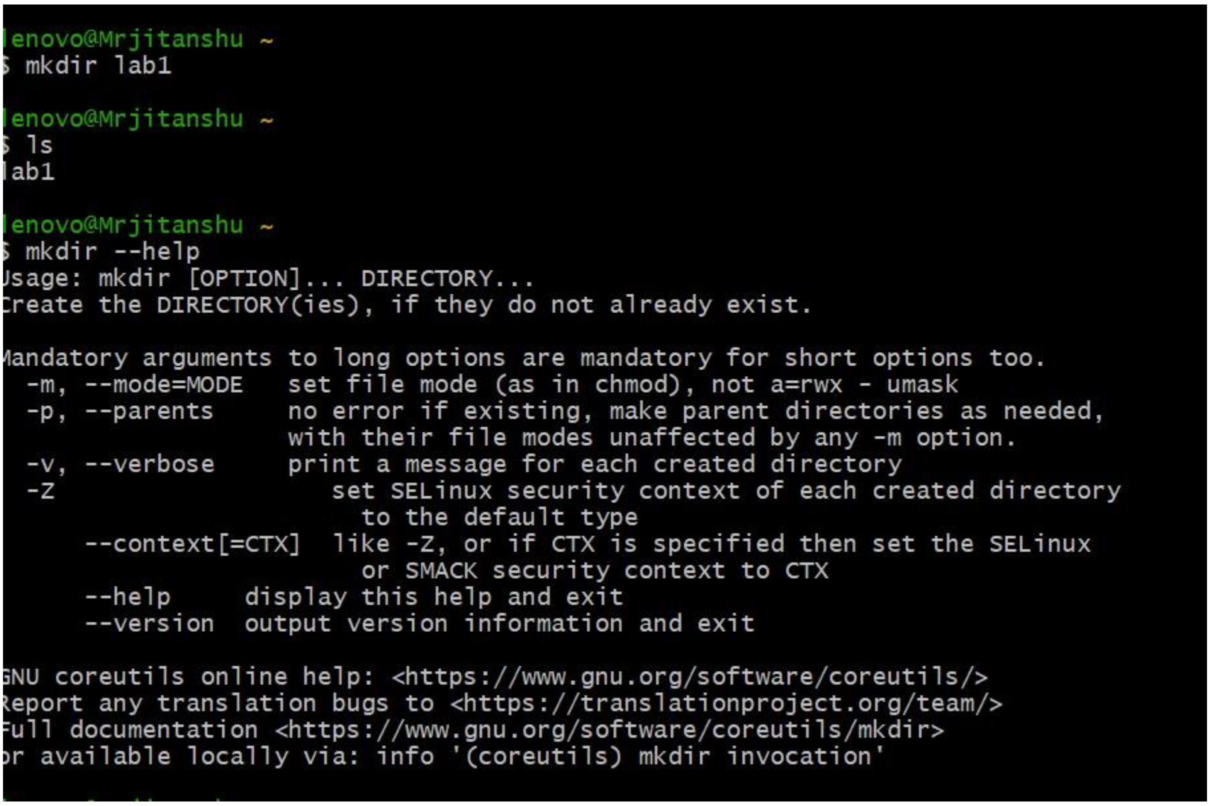
```
lenovo@Mrjitanshu ~
$ who --help
Usage: who [OPTION]... [ FILE | ARG1 ARG2 ]
Print information about users who are currently logged in.

  -a, --all         same as -b -d --login -p -r -t -T -u
  -b, --boot        time of last system boot
  -d, --dead        print dead processes
  -H, --heading     print line of column headings
  -l, --login       print system login processes
      --lookup      attempt to canonicalize hostnames via DNS
  -m                only hostname and user associated with stdin
  -p, --process     print active processes spawned by init
  -q, --count       all login names and number of users logged on
  -r, --runlevel    print current runlevel
  -s, --short       print only name, line, and time (default)
  -t, --time        print last system clock change
  -T, -w, --mesg    add user's message status as +, - or ?
  -u, --users       list users logged in
      --message     same as -T
      --writable    same as -T
      --help     display this help and exit
      --version  output version information and exit

If FILE is not specified, use /var/run/utmp.  /var/log/wtmp as FILE is common.
If ARG1 ARG2 given, -m presumed: 'am i' or 'mom likes' are usual.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/who>
or available locally via: info '(coreutils) who invocation'
```

Whoami



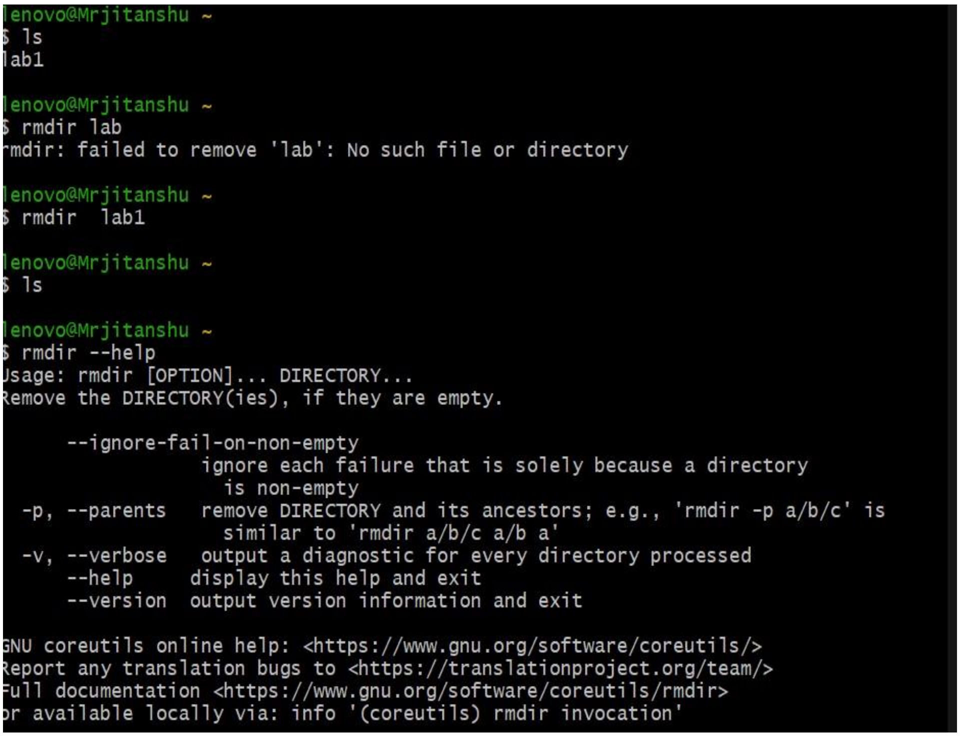
```
lenovo@Mrjitanshu ~
$ whoami
lenovo

lenovo@Mrjitanshu ~
$ whoami --help
Usage: whoami [OPTION]...
Print the user name associated with the current effective user I
Same as id -un.

      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreut
Report any translation bugs to <https://translationproject.org/t
Full documentation <https://www.gnu.org/software/coreutils/whoam
or available locally via: info '(coreutils) whoami invocation'
```

Mkdir



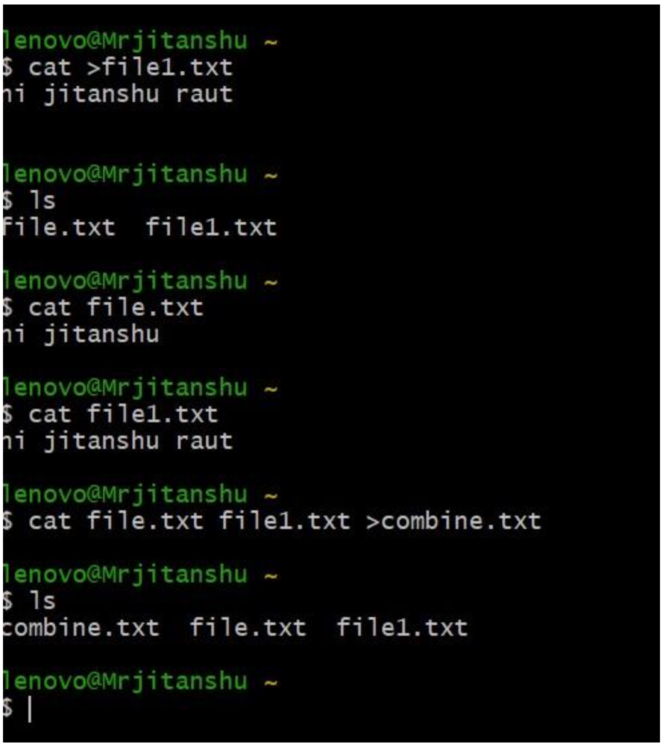
```
enovo@Mrjitanshu ~
 mkdir lab1

enovo@Mrjitanshu ~
 ls
ab1

enovo@Mrjitanshu ~
 mkdir --help
sage: mkdir [OPTION]... DIRECTORY...
reate the DIRECTORY(ies), if they do not already exist.

andatory arguments to long options are mandatory for short options too.
 -m, --mode=MODE   set file mode (as in chmod), not a=rwx - umask
 -p, --parents     no error if existing, make parent directories as needed,
                   with their file modes unaffected by any -m option.
 -v, --verbose     print a message for each created directory
 -Z                   set SELinux security context of each created directory
                        to the default type
     --context[=CTX]  like -Z, or if CTX is specified then set the SELinux
                        or SMACK security context to CTX
     --help     display this help and exit
     --version  output version information and exit

NU coreutils online help: <https://www.gnu.org/software/coreutils/>
eport any translation bugs to <https://translationproject.org/team/>
ull documentation <https://www.gnu.org/software/coreutils/mkdir>
r available locally via: info '(coreutils) mkdir invocation'
```

Rmdir



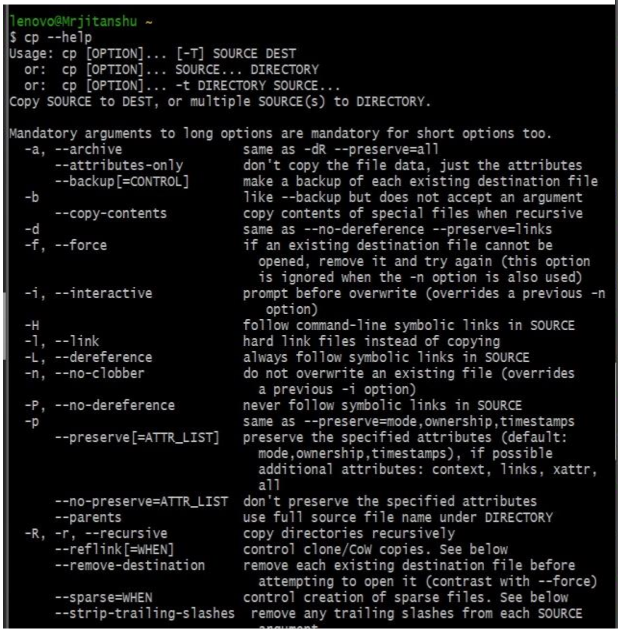
```
lenovo@Mrjitanshu ~
$ ls
lab1

lenovo@Mrjitanshu ~
$ rmdir lab
rmdir: failed to remove 'lab': No such file or directory

lenovo@Mrjitanshu ~
$ rmdir  lab1

lenovo@Mrjitanshu ~
$ ls

lenovo@Mrjitanshu ~
$ rmdir --help
Usage: rmdir [OPTION]... DIRECTORY...
Remove the DIRECTORY(ies), if they are empty.

      --ignore-fail-on-non-empty
                  ignore each failure that is solely because a directory
                    is non-empty
  -p, --parents   remove DIRECTORY and its ancestors; e.g., 'rmdir -p a/b/c' is
                    similar to 'rmdir a/b/c a/b a'
  -v, --verbose   output a diagnostic for every directory processed
      --help     display this help and exit
      --version  output version information and exit

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/rmdir>
or available locally via: info '(coreutils) rmdir invocation'
```

Cat

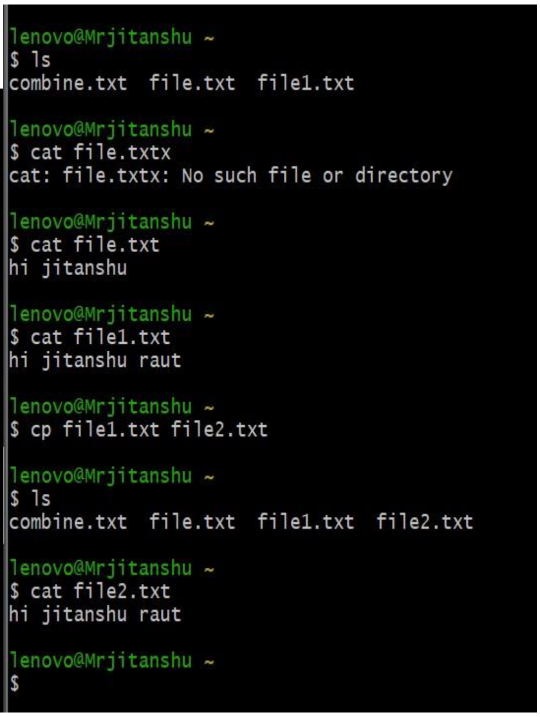


```
lenovo@Mrjitanshu ~
$ cat >file1.txt
hi jitanshu raut


lenovo@Mrjitanshu ~
$ ls
file.txt  file1.txt

lenovo@Mrjitanshu ~
$ cat file.txt
hi jitanshu

lenovo@Mrjitanshu ~
$ cat file1.txt
hi jitanshu raut

lenovo@Mrjitanshu ~
$ cat file.txt file1.txt >combine.txt

lenovo@Mrjitanshu ~
$ ls
combine.txt  file.txt  file1.txt

lenovo@Mrjitanshu ~
$ |
```

Cp

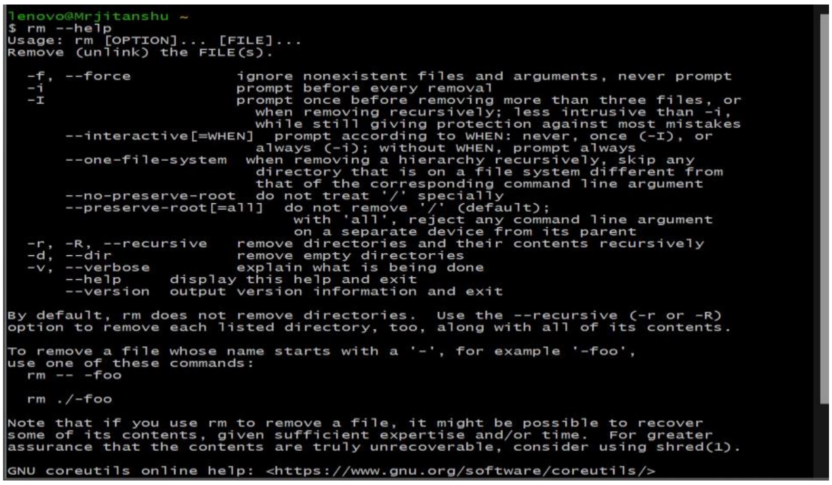


```
lenovo@Mrjitanshu ~
$ cp --help
Usage: cp [OPTION]... [-T] SOURCE DEST
  or:  cp [OPTION]... SOURCE... DIRECTORY
  or:  cp [OPTION]... -t DIRECTORY SOURCE...
Copy SOURCE to DEST, or multiple SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
  -a, --archive                same as -dR --preserve=all
      --attributes-only        don't copy the file data, just the attributes
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
      --copy-contents          copy contents of special files when recursive
  -d                           same as --no-dereference --preserve=links
  -f, --force                  if an existing destination file cannot be
                                 opened, remove it and try again (this option
                                 is ignored when the -n option is also used)
  -i, --interactive            prompt before overwrite (overrides a previous -n
                                  option)
  -H                           follow command-line symbolic links in SOURCE
  -l, --link                   hard link files instead of copying
  -L, --dereference            always follow symbolic links in SOURCE
  -n, --no-clobber             do not overwrite an existing file (overrides
                                 a previous -i option)
  -P, --no-dereference         never follow symbolic links in SOURCE
  -p                           same as --preserve=mode,ownership,timestamps
      --preserve[=ATTR_LIST]   preserve the specified attributes (default:
                                 mode,ownership,timestamps), if possible
                                 additional attributes: context, links, xattr,
                                 all
      --no-preserve=ATTR_LIST  don't preserve the specified attributes
      --parents                use full source file name under DIRECTORY
  -R, -r, --recursive          copy directories recursively
      --reflink[=WHEN]         control clone/CoW copies. See below
      --remove-destination     remove each existing destination file before
                                 attempting to open it (contrast with --force)
      --sparse=WHEN            control creation of sparse files. See below
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
```

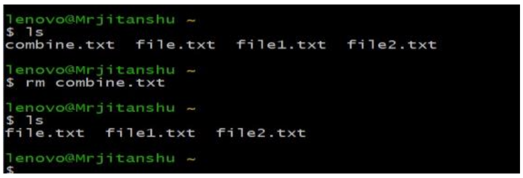


```
lenovo@Mrjitanshu ~
$ ls
combine.txt  file.txt  file1.txt

lenovo@Mrjitanshu ~
$ cat file.txtx
cat: file.txtx: No such file or directory

lenovo@Mrjitanshu ~
$ cat file.txt
hi jitanshu

lenovo@Mrjitanshu ~
$ cat file1.txt
hi jitanshu raut

lenovo@Mrjitanshu ~
$ cp file1.txt file2.txt

lenovo@Mrjitanshu ~
$ ls
combine.txt  file.txt  file1.txt  file2.txt

lenovo@Mrjitanshu ~
$ cat file2.txt
hi jitanshu raut

lenovo@Mrjitanshu ~
$
```

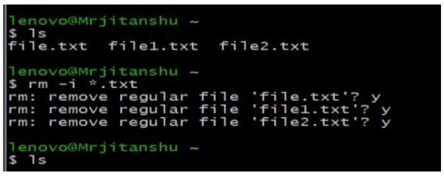
Rm



```
lenovo@Mrjitanshu ~
$ rm --help
Usage: rm [OPTION]... [FILE]...
Remove (unlink) the FILE(s).

  -f, --force           ignore nonexistent files and arguments, never prompt
  -i                    prompt before every removal
  -I                    prompt once before removing more than three files, or
                          when removing recursively; less intrusive than -i,
                          while still giving protection against most mistakes
      --interactive[=WHEN]  prompt according to WHEN: never, once (-I), or
                          always (-i); without WHEN, prompt always
      --one-file-system  when removing a hierarchy recursively, skip any
                          directory that is on a file system different from
                          that of the corresponding command line argument
      --no-preserve-root  do not treat '/' specially
      --preserve-root[=all]  do not remove '/' (default);
                              with 'all', reject any command line argument
                              on a separate device from its parent
  -r, -R, --recursive   remove directories and their contents recursively
  -d, --dir             remove empty directories
  -v, --verbose         explain what is being done
      --help     display this help and exit
      --version  output version information and exit

By default, rm does not remove directories.  Use the --recursive (-r or -R)
option to remove each listed directory, too, along with all of its contents.

To remove a file whose name starts with a '-', for example '-foo',
use one of these commands:
  rm -- -foo

  rm ./-foo

Note that if you use rm to remove a file, it might be possible to recover
some of its contents, given sufficient expertise and/or time.  For greater
assurance that the contents are truly unrecoverable, consider using shred(1).

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
```

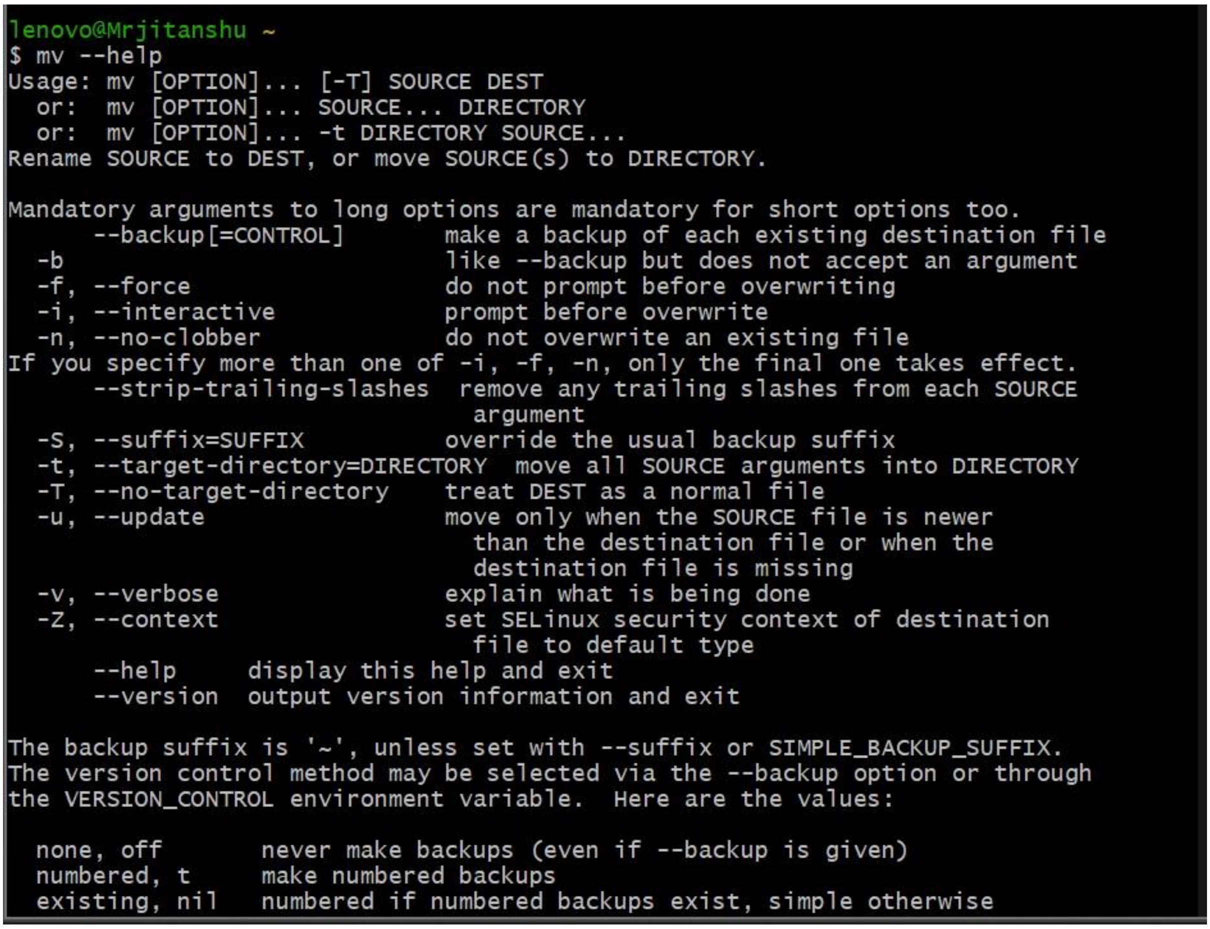


```
lenovo@Mrjitanshu ~
$ ls
combine.txt  file.txt  file1.txt  file2.txt

lenovo@Mrjitanshu ~
$ rm combine.txt

lenovo@Mrjitanshu ~
$ ls
file.txt  file1.txt  file2.txt

lenovo@Mrjitanshu ~
$
```



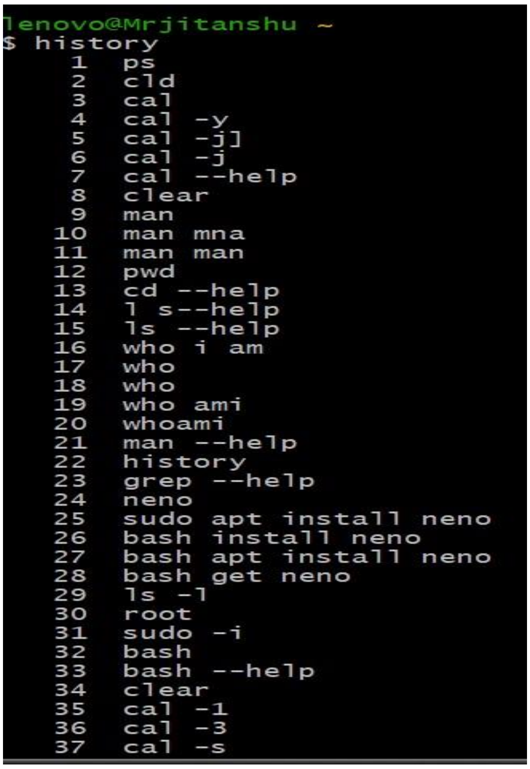
```
lenovo@Mrjitanshu ~
$ ls
file.txt  file1.txt  file2.txt

lenovo@Mrjitanshu ~
$ rm -i *.txt
rm: remove regular file 'file.txt'? y
rm: remove regular file 'file1.txt'? y
rm: remove regular file 'file2.txt'? y

lenovo@Mrjitanshu ~
$ ls
```

Mv



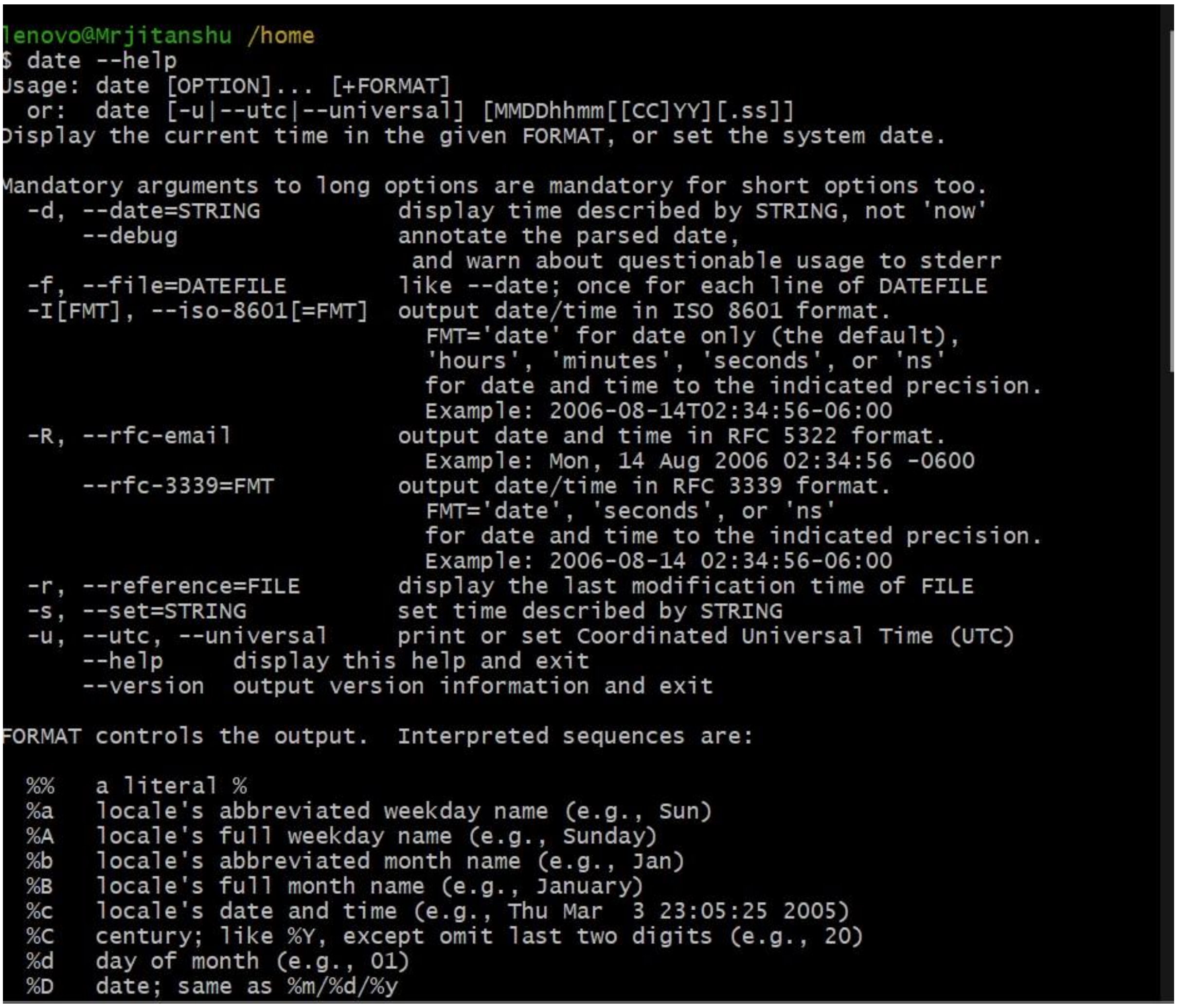
```
lenovo@Mrjitanshu ~
$ mv --help
Usage: mv [OPTION]... [-T] SOURCE DEST
  or:  mv [OPTION]... SOURCE... DIRECTORY
  or:  mv [OPTION]... -t DIRECTORY SOURCE...
Rename SOURCE to DEST, or move SOURCE(s) to DIRECTORY.

Mandatory arguments to long options are mandatory for short options too.
      --backup[=CONTROL]       make a backup of each existing destination file
  -b                           like --backup but does not accept an argument
  -f, --force                  do not prompt before overwriting
  -i, --interactive            prompt before overwrite
  -n, --no-clobber             do not overwrite an existing file
If you specify more than one of -i, -f, -n, only the final one takes effect.
      --strip-trailing-slashes  remove any trailing slashes from each SOURCE
                                 argument
  -S, --suffix=SUFFIX          override the usual backup suffix
  -t, --target-directory=DIRECTORY  move all SOURCE arguments into DIRECTORY
  -T, --no-target-directory    treat DEST as a normal file
  -u, --update                 move only when the SOURCE file is newer
                                 than the destination file or when the
                                 destination file is missing
  -v, --verbose                explain what is being done
  -Z, --context                set SELinux security context of destination
                                 file to default type
      --help     display this help and exit
      --version  output version information and exit

The backup suffix is '~', unless set with --suffix or SIMPLE_BACKUP_SUFFIX.
The version control method may be selected via the --backup option or through
the VERSION_CONTROL environment variable.  Here are the values:

  none, off       never make backups (even if --backup is given)
  numbered, t     make numbered backups
  existing, nil   numbered if numbered backups exist, simple otherwise
```

History



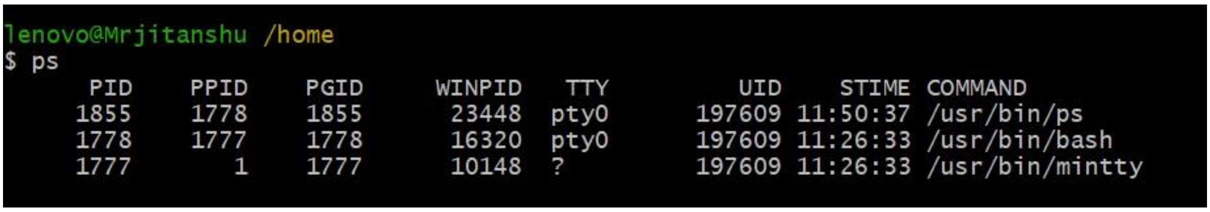
```
lenovo@Mrjitanshu ~
$ history
    1  ps
    2  cld
    3  cal
    4  cal -y
    5  cal -j]
    6  cal -j
    7  cal --help
    8  clear
    9  man
   10  man mna
   11  man man
   12  pwd
   13  cd --help
   14  l s--help
   15  ls --help
   16  who i am
   17  who
   18  who
   19  who ami
   20  whoami
   21  man --help
   22  history
   23  grep --help
   24  neno
   25  sudo apt install neno
   26  bash install neno
   27  bash apt install neno
   28  bash get neno
   29  ls -l
   30  root
   31  sudo -i
   32  bash
   33  bash --help
   34  clear
   35  cal -1
   36  cal -3
   37  cal -s
```

Date



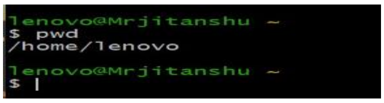
```
lenovo@Mrjitanshu /home
$ date --help
Usage: date [OPTION]... [+FORMAT]
  or:  date [-u|--utc|--universal] [MMDDhhmm[[CC]YY][.ss]]
Display the current time in the given FORMAT, or set the system date.

Mandatory arguments to long options are mandatory for short options too.
  -d, --date=STRING          display time described by STRING, not 'now'
      --debug                annotate the parsed date,
                              and warn about questionable usage to stderr
  -f, --file=DATEFILE        like --date; once for each line of DATEFILE
  -I[FMT], --iso-8601[=FMT]  output date/time in ISO 8601 format.
                               FMT='date' for date only (the default),
                               'hours', 'minutes', 'seconds', or 'ns'
                               for date and time to the indicated precision.
                               Example: 2006-08-14T02:34:56-06:00
  -R, --rfc-email            output date and time in RFC 5322 format.
                               Example: Mon, 14 Aug 2006 02:34:56 -0600
      --rfc-3339=FMT         output date/time in RFC 3339 format.
                               FMT='date', 'seconds', or 'ns'
                               for date and time to the indicated precision.
                               Example: 2006-08-14 02:34:56-06:00
  -r, --reference=FILE       display the last modification time of FILE
  -s, --set=STRING           set time described by STRING
  -u, --utc, --universal     print or set Coordinated Universal Time (UTC)
      --help     display this help and exit
      --version  output version information and exit

FORMAT controls the output.  Interpreted sequences are:

  %%   a literal %
  %a   locale's abbreviated weekday name (e.g., Sun)
  %A   locale's full weekday name (e.g., Sunday)
  %b   locale's abbreviated month name (e.g., Jan)
  %B   locale's full month name (e.g., January)
  %c   locale's date and time (e.g., Thu Mar  3 23:05:25 2005)
  %C   century; like %Y, except omit last two digits (e.g., 20)
  %d   day of month (e.g., 01)
  %D   date; same as %m/%d/%y
```

Ps

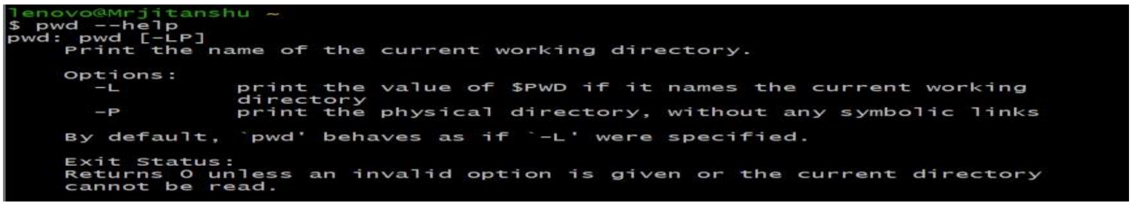


```
lenovo@Mrjitanshu /home
$ ps
      PID    PPID    PGID     WINPID   TTY         UID    STIME COMMAND
     1855    1778    1855      23448  pty0      197609 11:50:37 /usr/bin/ps
     1778    1777    1778      16320  pty0      197609 11:26:33 /usr/bin/bash
     1777       1    1777      10148  ?         197609 11:26:33 /usr/bin/mintty
```

Pwd

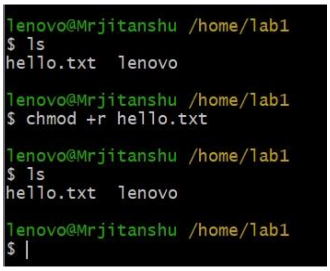


```
lenovo@Mrjitanshu ~
$ pwd
/home/lenovo

lenovo@Mrjitanshu ~
$ |
```

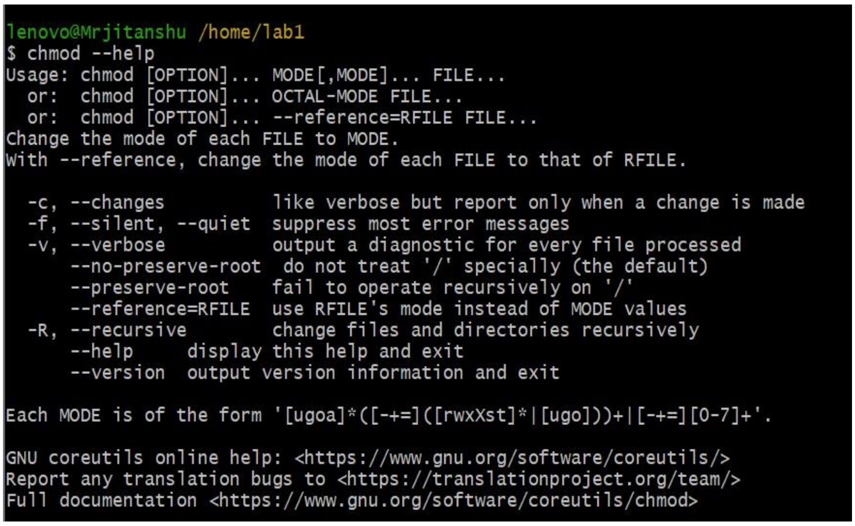


```
lenovo@Mrjitanshu ~
$ pwd --help
pwd: pwd [-LP]
    Print the name of the current working directory.

    Options:
      -L        print the value of $PWD if it names the current working
                directory
      -P        print the physical directory, without any symbolic links

    By default, `pwd' behaves as if `-L' were specified.

    Exit Status:
    Returns 0 unless an invalid option is given or the current directory
    cannot be read.
```

Chmod

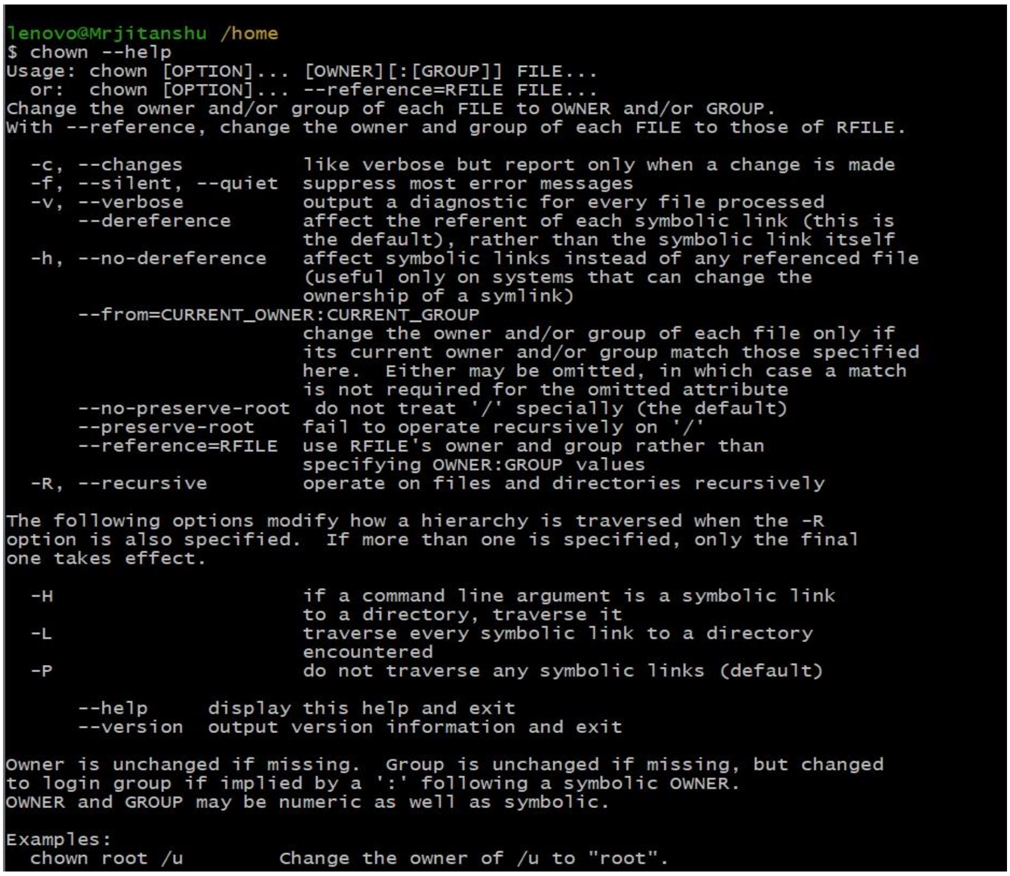


```
lenovo@Mrjitanshu /home/lab1
$ ls
hello.txt  lenovo

lenovo@Mrjitanshu /home/lab1
$ chmod +r hello.txt

lenovo@Mrjitanshu /home/lab1
$ ls
hello.txt  lenovo

lenovo@Mrjitanshu /home/lab1
$ |
```



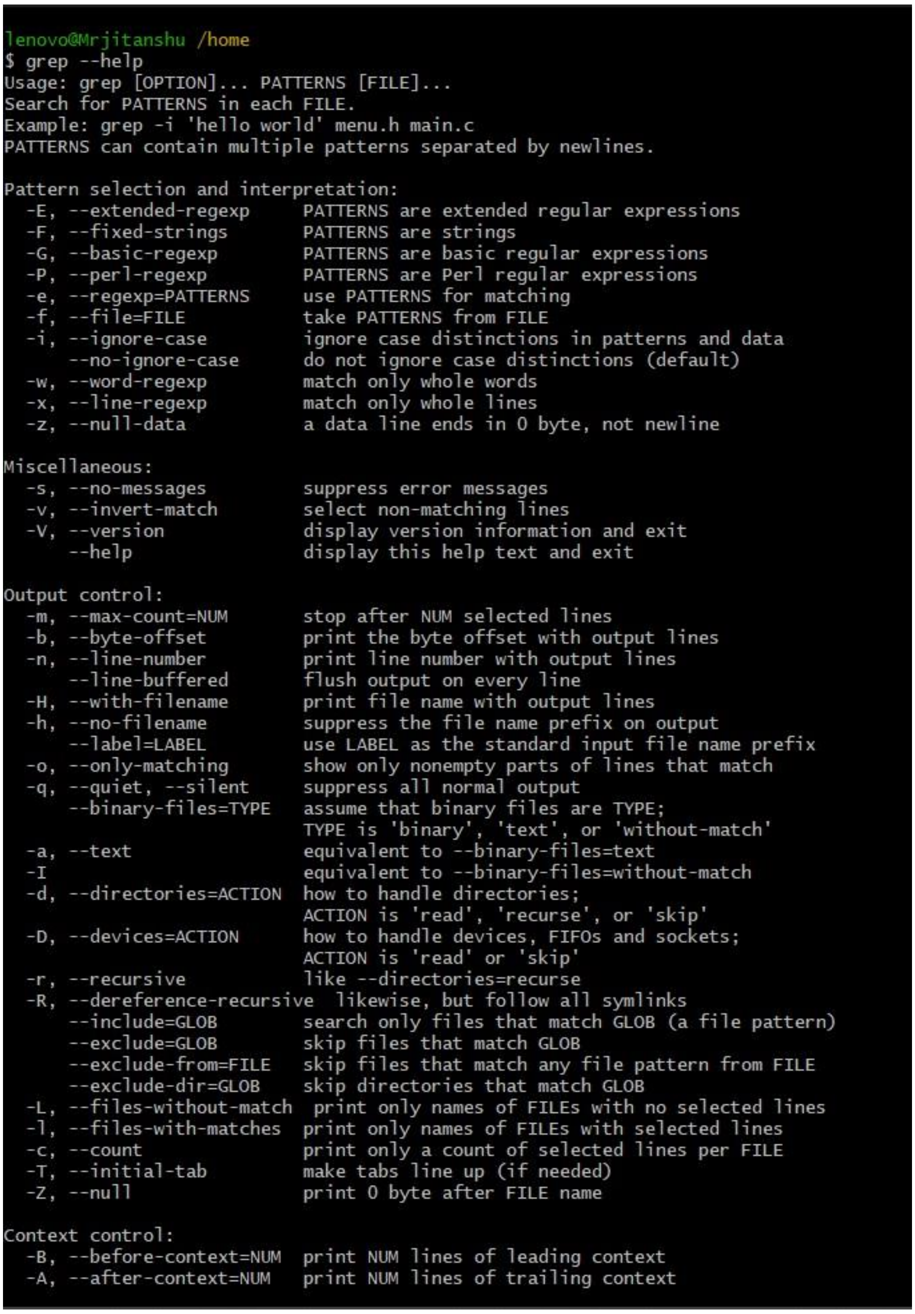
```
lenovo@Mrjitanshu /home/lab1
$ chmod --help
Usage: chmod [OPTION]... MODE[,MODE]... FILE...
  or:  chmod [OPTION]... OCTAL-MODE FILE...
  or:  chmod [OPTION]... --reference=RFILE FILE...
Change the mode of each FILE to MODE.
With --reference, change the mode of each FILE to that of RFILE.

  -c, --changes          like verbose but report only when a change is made
  -f, --silent, --quiet  suppress most error messages
  -v, --verbose          output a diagnostic for every file processed
      --no-preserve-root  do not treat '/' specially (the default)
      --preserve-root    fail to operate recursively on '/'
      --reference=RFILE  use RFILE's mode instead of MODE values
  -R, --recursive        change files and directories recursively
      --help     display this help and exit
      --version  output version information and exit

Each MODE is of the form '[ugoa]*([-+=]([rwxXst]*|[ugo]))+|[-+=][0-7]+'.

GNU coreutils online help: <https://www.gnu.org/software/coreutils/>
Report any translation bugs to <https://translationproject.org/team/>
Full documentation <https://www.gnu.org/software/coreutils/chmod>
```

Chown



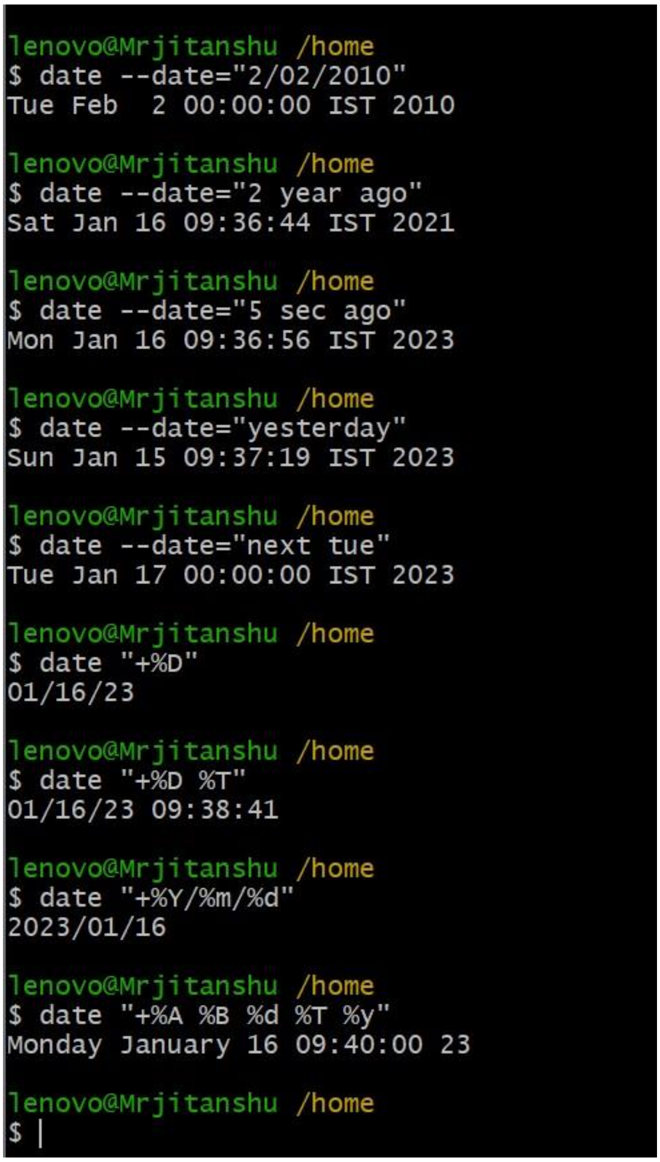
```
lenovo@Mrjitanshu /home
$ chown --help
Usage: chown [OPTION]... [OWNER][:[GROUP]] FILE...
  or:  chown [OPTION]... --reference=RFILE FILE...
Change the owner and/or group of each FILE to OWNER and/or GROUP.
With --reference, change the owner and group of each FILE to those of RFILE.

  -c, --changes          like verbose but report only when a change is made
  -f, --silent, --quiet  suppress most error messages
  -v, --verbose          output a diagnostic for every file processed
      --dereference      affect the referent of each symbolic link (this is
                         the default), rather than the symbolic link itself
  -h, --no-dereference   affect symbolic links instead of any referenced file
                         (useful only on systems that can change the
                         ownership of a symlink)
      --from=CURRENT_OWNER:CURRENT_GROUP
                         change the owner and/or group of each file only if
                         its current owner and/or group match those specified
                         here.  Either may be omitted, in which case a match
                         is not required for the omitted attribute
      --no-preserve-root  do not treat '/' specially (the default)
      --preserve-root    fail to operate recursively on '/'
      --reference=RFILE  use RFILE's owner and group rather than
                         specifying OWNER:GROUP values
  -R, --recursive        operate on files and directories recursively

The following options modify how a hierarchy is traversed when the -R
option is also specified.  If more than one is specified, only the final
one takes effect.

  -H                     if a command line argument is a symbolic link
                         to a directory, traverse it
  -L                     traverse every symbolic link to a directory
                         encountered
  -P                     do not traverse any symbolic links (default)

      --help     display this help and exit
      --version  output version information and exit

Owner is unchanged if missing.  Group is unchanged if missing, but changed
to login group if implied by a ':' following a symbolic OWNER.
OWNER and GROUP may be numeric as well as symbolic.

Examples:
  chown root /u        Change the owner of /u to "root".
```

Grep



```
lenovo@Mrjitanshu /home
$ grep --help
Usage: grep [OPTION]... PATTERNS [FILE]...
Search for PATTERNS in each FILE.
Example: grep -i 'hello world' menu.h main.c
PATTERNS can contain multiple patterns separated by newlines.

Pattern selection and interpretation:
  -E, --extended-regexp     PATTERNS are extended regular expressions
  -F, --fixed-strings       PATTERNS are strings
  -G, --basic-regexp        PATTERNS are basic regular expressions
  -P, --perl-regexp         PATTERNS are Perl regular expressions
  -e, --regexp=PATTERNS     use PATTERNS for matching
  -f, --file=FILE           take PATTERNS from FILE
  -i, --ignore-case         ignore case distinctions in patterns and data
      --no-ignore-case      do not ignore case distinctions (default)
  -w, --word-regexp         match only whole words
  -x, --line-regexp         match only whole lines
  -z, --null-data           a data line ends in 0 byte, not newline

Miscellaneous:
  -s, --no-messages         suppress error messages
  -v, --invert-match        select non-matching lines
  -V, --version             display version information and exit
      --help                display this help text and exit

Output control:
  -m, --max-count=NUM       stop after NUM selected lines
  -b, --byte-offset         print the byte offset with output lines
  -n, --line-number         print line number with output lines
      --line-buffered       flush output on every line
  -H, --with-filename       print file name with output lines
  -h, --no-filename         suppress the file name prefix on output
      --label=LABEL         use LABEL as the standard input file name prefix
  -o, --only-matching       show only nonempty parts of lines that match
  -q, --quiet, --silent     suppress all normal output
      --binary-files=TYPE   assume that binary files are TYPE;
                            TYPE is 'binary', 'text', or 'without-match'
  -a, --text                equivalent to --binary-files=text
  -I                        equivalent to --binary-files=without-match
  -d, --directories=ACTION  how to handle directories;
                            ACTION is 'read', 'recurse', or 'skip'
  -D, --devices=ACTION      how to handle devices, FIFOs and sockets;
                            ACTION is 'read' or 'skip'
  -r, --recursive           like --directories=recurse
  -R, --dereference-recursive  likewise, but follow all symlinks
      --include=GLOB        search only files that match GLOB (a file pattern)
      --exclude=GLOB        skip files that match GLOB
      --exclude-from=FILE   skip files that match any file pattern from FILE
      --exclude-dir=GLOB    skip directories that match GLOB
  -L, --files-without-match  print only names of FILEs with no selected lines
  -l, --files-with-matches  print only names of FILEs with selected lines
  -c, --count               print only a count of selected lines per FILE
  -T, --initial-tab         make tabs line up (if needed)
  -Z, --null                print 0 byte after FILE name

Context control:
  -B, --before-context=NUM  print NUM lines of leading context
  -A, --after-context=NUM   print NUM lines of trailing context
```

Date



```
lenovo@Mrjitanshu /home
$ date --date="2/02/2010"
Tue Feb  2 00:00:00 IST 2010

lenovo@Mrjitanshu /home
$ date --date="2 year ago"
Sat Jan 16 09:36:44 IST 2021

lenovo@Mrjitanshu /home
$ date --date="5 sec ago"
Mon Jan 16 09:36:56 IST 2023

lenovo@Mrjitanshu /home
$ date --date="yesterday"
Sun Jan 15 09:37:19 IST 2023

lenovo@Mrjitanshu /home
$ date --date="next tue"
Tue Jan 17 00:00:00 IST 2023

lenovo@Mrjitanshu /home
$ date "+%D"
01/16/23

lenovo@Mrjitanshu /home
$ date "+%D %T"
01/16/23 09:38:41

lenovo@Mrjitanshu /home
$ date "+%Y/%m/%d"
2023/01/16

lenovo@Mrjitanshu /home
$ date "+%A %B %d %T %y"
Monday January 16 09:40:00 23

lenovo@Mrjitanshu /home
$ |
```

Assignment 2

Question-1

```
┌──(jitanshu㉿kali)-[~/assigments]
└─$ mkdir COMP

┌──(jitanshu㉿kali)-[~/assigments]
└─$ ls
COMP

┌──(jitanshu㉿kali)-[~/assigments]
└─$ cd COMP

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ LS
LS: command not found

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ ls

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ mkdir Comp1 Comp2 Comp3

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ ls
Comp1  Comp2  Comp3

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ cd Comp1

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1]
└─$ mkdir CPU

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1]
└─$ cd CPU

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ mkdir Control ALU

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ cd Control

┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
```

```
└─$ cat> Comp.c
hello
^C

┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ cat>p1.txt
hi
^C

┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ cat >p2.txt
hi hello
^C

┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ ls
Comp.c  p1.txt  p2.txt

┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ cd /home/jitanshu/assigments/COMP/Comp2

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ ls

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ mkdir In_out

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ cd ..

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ cd Comp3

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3]
└─$ mkdir Memory

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3]
└─$ cd Memory

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ mkdir RAM ROM

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ cd RAM

┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/RAM]
└─$ cat>play.wml
hi
^C

┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/RAM]
└─$ cat>play2.wml
hello
```

```
^C

┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/RAM]
└─$ cd ..

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ cd ROM

┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/ROM]
└─$ ls
pict1.jpeg  pict2.jpeg

┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/ROM]
└─$ cd /home/jitanshu/assigments/COMP
┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       └── Control
│           ├── Comp.c
│           ├── p1.txt
│           └── p2.txt
├── Comp2
│   └── In_out
└── Comp3
    └── Memory
        ├── RAM
        │   ├── play2.wml
        │   └── play.wml
        └── ROM
            ├── pict1.jpeg
            └── pict2.jpeg

10 directories, 7 files
```

Question-2

```
┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2/In_out]
└─$ cp /home/jitanshu/assigments/COMP/Comp1/CPU/Control/Comp.c
/home/jitanshu/assigments/COMP/Comp1/CPU/Control/Computer.c


┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           │
│           ├── p1.txt
│           └── p2.txt
├── Comp2
│   └── In_out
└── Comp3
    └── Memory
        ├── RAM
        │   ├── play2.wml
        │   └── play.wml
        └── ROM
            ├── pict1.jpeg
            └── pict2.jpeg

10 directories, 8 files
```

Question-3

```
┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ cp /home/jitanshu/assigments/COMP/Comp1/CPU/Control/p2.txt
/home/jitanshu/assigments/COMP/Comp1/CPU/Control/p3.txt

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ tree
.
├── ALU
└── Control
    ├── Comp.c
    ├── Computer.c
    ├── p1.txt
    ├── p2.txt
    └── p3.txt

2 directories, 5 files


┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ cp /home/jitanshu/assigments/COMP/Comp1/CPU/Control/p2.txt
/home/jitanshu/assigments/COMP/Comp1/CPU/ALU/p3.txt

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp1/CPU]
└─$ tree
.
├── ALU
│   └── p3.txt
└── Control
    ├── Comp.c
    ├── Computer.c
    ├── p1.txt
    ├── p2.txt
    └── p3.txt

2 directories, 6 files
```

Question-4

```
─(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ cp /home/jitanshu/assigments/COMP/Comp1/CPU/Control/p1.txt
/home/jitanshu/assigments/COMP/Comp3/Memory/ROM/p3.txt

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ tree
.
├── RAM
│   ├── play2.wml
│   └── play.wml
└── ROM
    ├── p3.txt
    ├── pict1.jpeg
    └── pict2.jpeg

2 directories, 5 files
```

Question-5

```
(jitanshu㉿kali)-[~/assigments/COMP]
└─$ cp /home/jitanshu/assigments/COMP/Comp3/Memory/ROM/pict1.jpeg
/home/jitanshu/assigments/COMP/Comp1/CPU/ALU/pict1.jpeg


┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       │   ├── p3.txt
│       │   └── pict1.jpeg
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           ├── p1.txt
│           ├── p2.txt
│           └── p3.txt
├── Comp2
│   └── In_out
└── Comp3
    └── Memory
        ├── RAM
        │   ├── play2.wml
        │   └── play.wml
        └── ROM
            ├── p3.txt
            ├── pict1.jpeg
            └── pict2.jpeg
```

Question-6

```
┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ mv /home/jitanshu/assigments/COMP/Comp3/Memory/RAM/play.wml
/home/jitanshu/assigments/COMP/Comp3/Memory/RAM/playlist.wml

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ tree
.
└── In_out

1 directory, 0 files

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ cd ..

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       │   ├── p3.txt
│       │   └── pict1.jpeg
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           ├── p1.txt
│           ├── p2.txt
│           └── p3.txt
├── Comp2
│   └── In_out
└── Comp3
    └── Memory
        ├── RAM
        │   ├── play2.wml
        │   └── playlist.wml
        └── ROM
            ├── p3.txt
            ├── pict1.jpeg
            └── pict2.jpeg

10 directories, 12 files
```

Question-7

```
┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ find -iname  '*.txt'
./Comp3/Memory/RAM/p3.txt
./Comp1/CPU/Control/p1.txt
./Comp1/CPU/Control/p3.txt
./Comp1/CPU/Control/p2.txt
```

```
./Comp1/CPU/ALU/p3.txt
./Comp2/In_out/p3.txt
```

Question-8

```
─(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/ALU]
└─$ cp /home/jitanshu/assigments/COMP/Comp1/CPU/ALU/*
/home/jitanshu/assigments/COMP/Comp2/In_out/


┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       │   ├── p3.txt
│       │   └── pict1.jpeg
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           ├── p1.txt
│           ├── p2.txt
│           └── p3.txt
├── Comp2
│   └── In_out
│       ├── p3.txt
│       └── pict1.jpeg
└── Comp3
    └── Memory
        ├── RAM
        │   ├── play2.wml
        │   └── playlist.wml
        └── ROM
            ├── p3.txt
            ├── pict1.jpeg
            └── pict2.jpeg

10 directories, 14 files
```

Question-9

```
┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2/In_out]
└─$ mv /home/jitanshu/assigments/COMP/Comp3/Memory/ROM/*
/home/jitanshu/assigments/COMP/Comp3/Memory/RAM


┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
```

```
│   └── CPU
│       ├── ALU
│       │   ├── p3.txt
│       │   └── pict1.jpeg
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           ├── p1.txt
│           ├── p2.txt
│           └── p3.txt
├── Comp2
│   └── In_out
│       ├── p3.txt
│       └── pict1.jpeg
└── Comp3
    └── Memory
        ├── RAM
        │   ├── p3.txt
        │   ├── pict1.jpeg
        │   ├── pict2.jpeg
        │   ├── play2.wml
        │   └── playlist.wml
        └── ROM

10 directories, 14 files
```

Question-10

```
┌──(jitanshu㉿kali)-[~/…/COMP/Comp3/Memory/RAM]
└─$ ls [a-z]*[a-z]*[a-z]*[y]*
play2.wml  playlist.wml
```

Question-11

```
┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2/In_out]
└─$ cp [p]* /home/jitanshu/assigments/COMP/Comp2/

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2/In_out]
└─$ cd ..

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp2]
└─$ ls
In_out  p3.txt  pict1.jpeg
```

Question-12

```
──(jitanshu㉿kali)-[~/assigments/COMP/Comp3]
└─$ cd Memory

┌──(jitanshu㉿kali)-[~/assigments/COMP/Comp3/Memory]
└─$ rm /home/jitanshu/assigments/COMP/Comp1/CPU/ALU/*
zsh: sure you want to delete all 2 files in
/home/jitanshu/assigments/COMP/Comp1/CPU/ALU [yn]? Y
```

Question-13

```
┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ rm /home/jitanshu/assigments/COMP/Comp2/In_out/*.log

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ tree
.
├── Comp1
│   └── CPU
│       ├── ALU
│       └── Control
│           ├── Comp.c
│           ├── Computer.c
│           ├── p1.txt
│           ├── p2.txt
│           └── p3.txt
├── Comp2
│   ├── In_out
│   │   ├── p3.txt
│   │   └── pict1.jpeg
│   ├── p3.txt
│   └── pict1.jpeg
└── Comp3
    └── Memory
        ├── RAM
        │   ├── p3.txt
        │   ├── pict1.jpeg
        │   ├── pict2.jpeg
        │   ├── play2.wml
        │   └── playlist.wml
        └── ROM

10 directories, 14 files
```

Question-14

```
┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ ls
Comp.c  Computer.c  p1.txt  p2.txt  p3.txt
```

```
┌──(jitanshu㉿kali)-[~/…/COMP/Comp1/CPU/Control]
└─$ ls -la
total 28
drwxr-xr-x 2 jitanshu jitanshu 4096 Jan 18 19:13 .
drwxr-xr-x 4 jitanshu jitanshu 4096 Jan 18 19:00 ..
-rw-r--r-- 1 jitanshu jitanshu    6 Jan 18 19:00 Comp.c
-rw-r--r-- 1 jitanshu jitanshu    6 Jan 18 19:10 Computer.c
-rw-r--r-- 1 jitanshu jitanshu    3 Jan 18 19:00 p1.txt
-rw-r--r-- 1 jitanshu jitanshu    9 Jan 18 19:00 p2.txt
-rw-r--r-- 1 jitanshu jitanshu    9 Jan 18 19:13 p3.txt
```

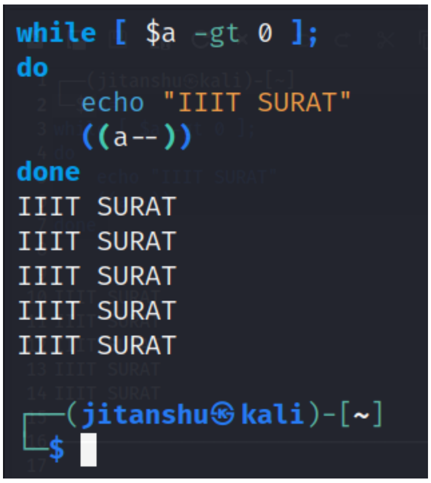
Question-15

```
┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ who
jitanshu tty7         2023-01-23 09:25 (:0)

┌──(jitanshu㉿kali)-[~/assigments/COMP]
└─$ whoami
jitanshu
```

Assignment 3

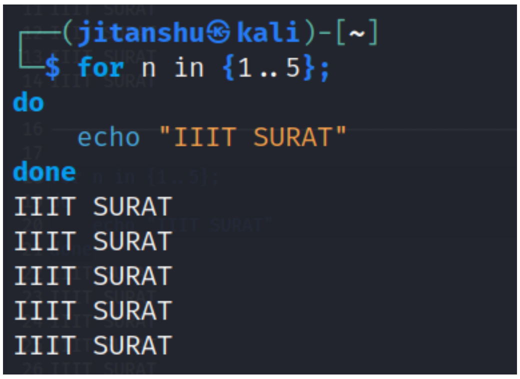
1. Print "IIIT Surat" message 5 times using for, while and do--while loop

```
┌──(jitanshu㉿kali)-[~]
└─$ a=5
while [ $a -gt 0 ];
do
    echo "IIIT SURAT"
    ((a--))
done
```



```
while [ $a -gt 0 ];
do
    echo "IIIT SURAT"
    ((a--))
done
IIIT SURAT
IIIT SURAT
IIIT SURAT
IIIT SURAT
IIIT SURAT

┌──(jitanshu㉿kali)-[~]
└─$ 
```

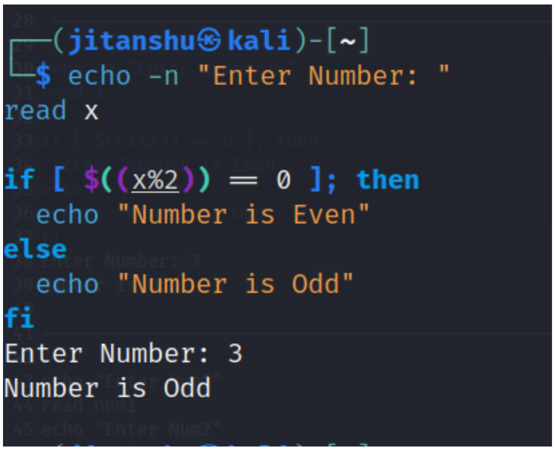
```
for n in {1..5};
do
    echo "IIIT SURAT"
done
```



```
┌──(jitanshu㉿kali)-[~]
└─$ for n in {1..5};
do
    echo "IIIT SURAT"
done
IIIT SURAT
IIIT SURAT
IIIT SURAT
IIIT SURAT
IIIT SURAT
```

2 Check entered number is even or odd

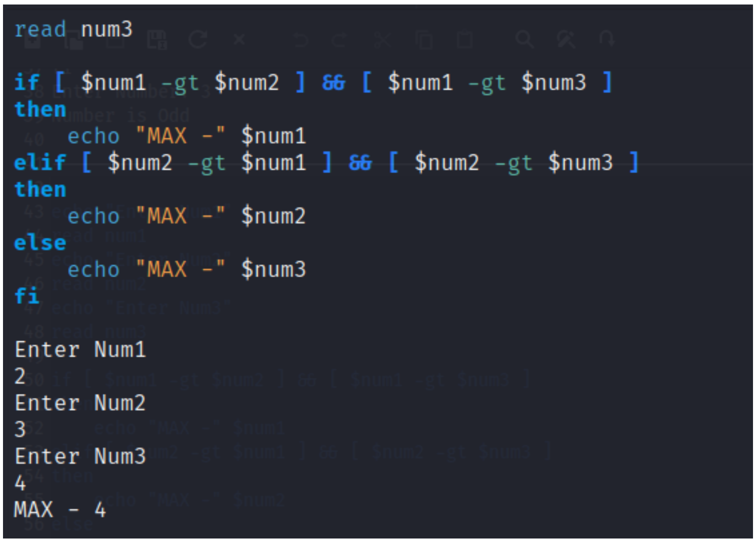
```
 echo -n "Enter Number: "
read x

if [ $((x%2)) == 0 ]; then
  echo "Number is Even"
else
  echo "Number is Odd"
fi
Enter Number: 3
Number is Odd
```



```
┌──(jitanshu㉿kali)-[~]
└─$ echo -n "Enter Number: "
read x

if [ $((x%2)) == 0 ]; then
  echo "Number is Even"
else
  echo "Number is Odd"
fi
Enter Number: 3
Number is Odd
```

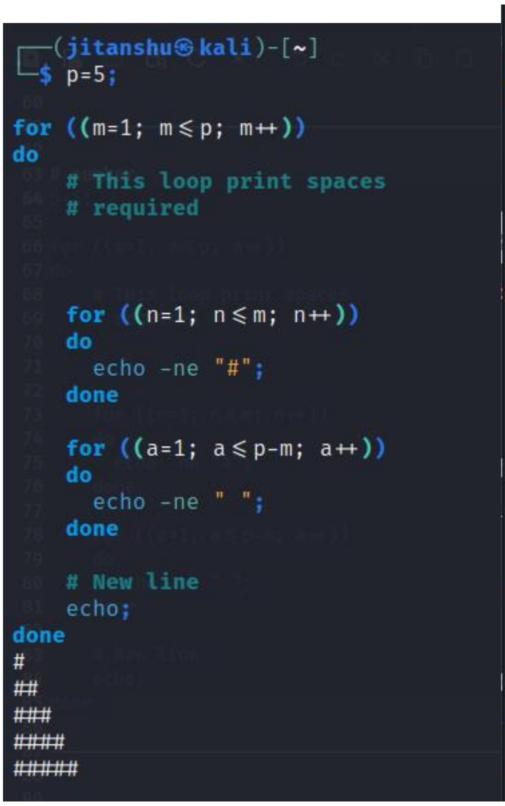
3 Check greatest among three integers on user input

```
echo "Enter Num1"
read num1
echo "Enter Num2"
read num2
echo "Enter Num3"
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo "MAX -" $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo "MAX -" $num2
else
    echo "MAX -" $num3
fi
```

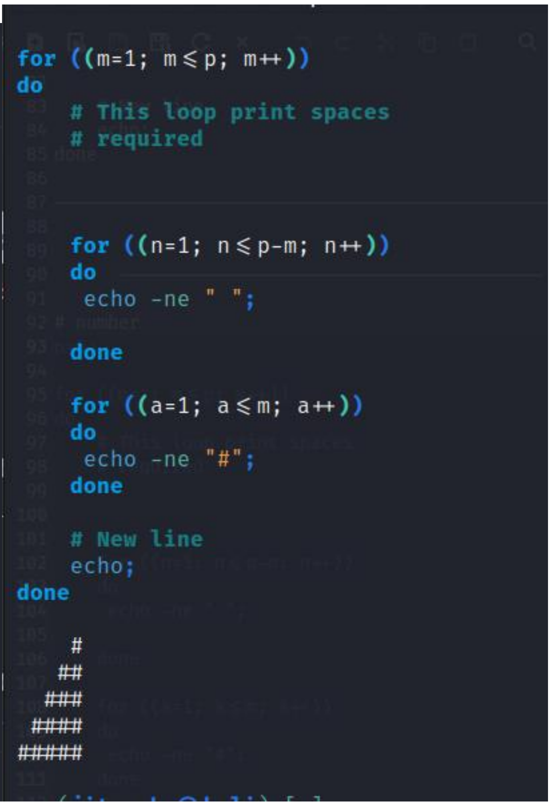


```
read num3

if [ $num1 -gt $num2 ] && [ $num1 -gt $num3 ]
then
    echo "MAX -" $num1
elif [ $num2 -gt $num1 ] && [ $num2 -gt $num3 ]
then
    echo "MAX -" $num2
else
    echo "MAX -" $num3
fi

Enter Num1
2
Enter Num2
3
Enter Num3
4
MAX - 4
```

5 . Print the following pattern



```
┌──(jitanshu㉿kali)-[~]
└─$ p=5;

for ((m=1; m<=p; m++))
do

    # This loop print spaces
    # required


    for ((n=1; n<=m; n++))
    do
      echo -ne "#";
    done

    for ((a=1; a<=p-m; a++))
    do
      echo -ne " ";
    done

    # New line
    echo;
done
#
##
###
####
#####
```



```
for ((m=1; m<=p; m++))
do

    # This loop print spaces
    # required



    for ((n=1; n<=p-m; n++))
    do
     echo -ne " ";

    done

    for ((a=1; a<=m; a++))
    do
     echo -ne "#";
    done

    # New line
    echo;
done
    #
   ##
  ###
 ####
#####
```

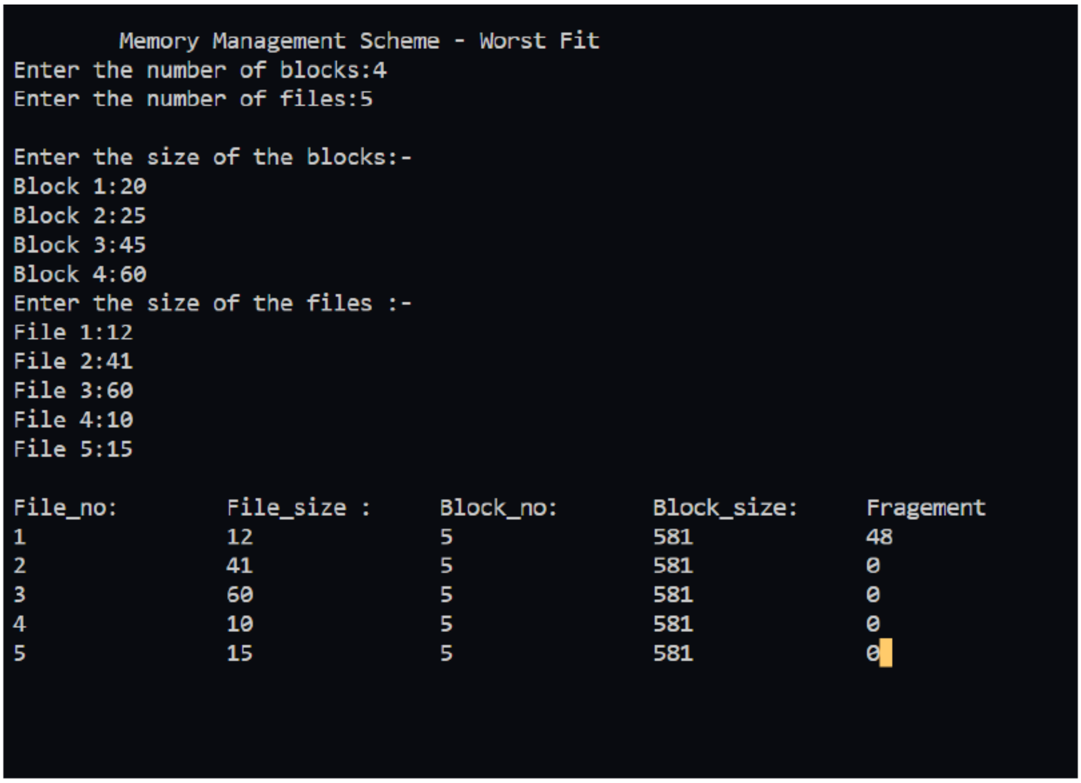
Assignment 4

Implement a program for memory allocation strategies using switch case or functions as follows:
1. Best fit
2. First fit
3. Worst fit

Worst fit

```
#include<stdio.h>
#include<conio.h>
#define max 25
void main()
{
int frag[max],b[max],f[max],i,j,nb,nf,temp,highest=0;
static int bf[max],ff[max];
printf("\n\tMemory Management Scheme - Worst Fit");
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of files:");
scanf("%d",&nf);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block %d:",i);
scanf("%d",&b[i]);
}
printf("Enter the size of the files :-\n");
for(i=1;i<=nf;i++)
{
printf("File %d:",i);
scanf("%d",&f[i]);
}
for(i=1;i<=nf;i++)
{
for(j=1;j<=nb;j++)
{
if(bf[j]!=1) //if bf[j] is not allocated
{
temp=b[j]-f[i];
if(temp>=0)
if(highest<temp)
{
ff[i]=j;
highest=temp;
}
}
frag[i]=highest;
bf[ff[i]]=1;
highest=0;
}
ff[i]=j;
highest=temp;
}
printf("\nFile_no:\tFile_size :\tBlock_no:\tBlock_size:\tFragement");
for(i=1;i<=nf;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,f[i],ff[i],b[ff[i]],frag[i]);
getch();
}
```

OUTPUT-----

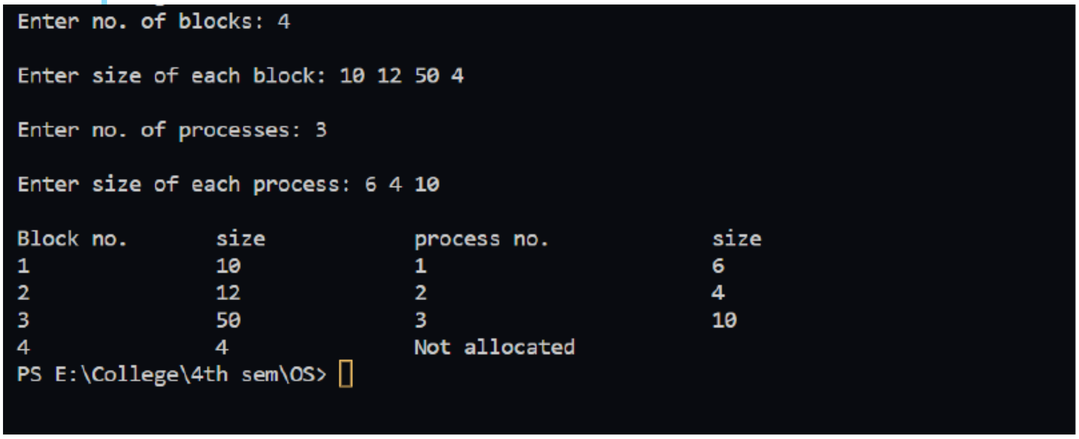


```
        Memory Management Scheme - Worst Fit
Enter the number of blocks:4
Enter the number of files:5

Enter the size of the blocks:-
Block 1:20
Block 2:25
Block 3:45
Block 4:60
Enter the size of the files :-
File 1:12
File 2:41
File 3:60
File 4:10
File 5:15

File_no:        File_size :     Block_no:       Block_size:     Fragement
1               12              5               581             48
2               41              5               581             0
3               60              5               581             0
4               10              5               581             0
5               15              5               581             0
```

First Fit:

```
#include<stdio.h>
void main()
{
    int bsize[10], psize[10], bno, pno, flags[10], allocation[10], i, j;
    for(i = 0; i < 10; i++)
    {
        flags[i] = 0;
        allocation[i] = -1;
    }
    printf("Enter no. of blocks: ");
    scanf("%d", &bno);
    printf("\nEnter size of each block: ");
    for(i = 0; i < bno; i++)
        scanf("%d", &bsize[i]);
    printf("\nEnter no. of processes: ");
    scanf("%d", &pno);
    printf("\nEnter size of each process: ");
    for(i = 0; i < pno; i++)
        scanf("%d", &psize[i]);
    for(i = 0; i < pno; i++)         //allocation as per first fit
        for(j = 0; j < bno; j++)
            if(flags[j] == 0 && bsize[j] >= psize[i])
            {
                allocation[j] = i;
                flags[j] = 1;
                break;
            }
    //display allocation details
    printf("\nBlock no.\tsize\t\tprocess no.\t\tsize");
    for(i = 0; i < bno; i++)
    {
        printf("\n%d\t\t%d\t\t", i+1, bsize[i]);
        if(flags[i] == 1)
            printf("%d\t\t\t%d",allocation[i]+1,psize[allocation[i]]);
        else
            printf("Not allocated");
    }
}
```

Output----

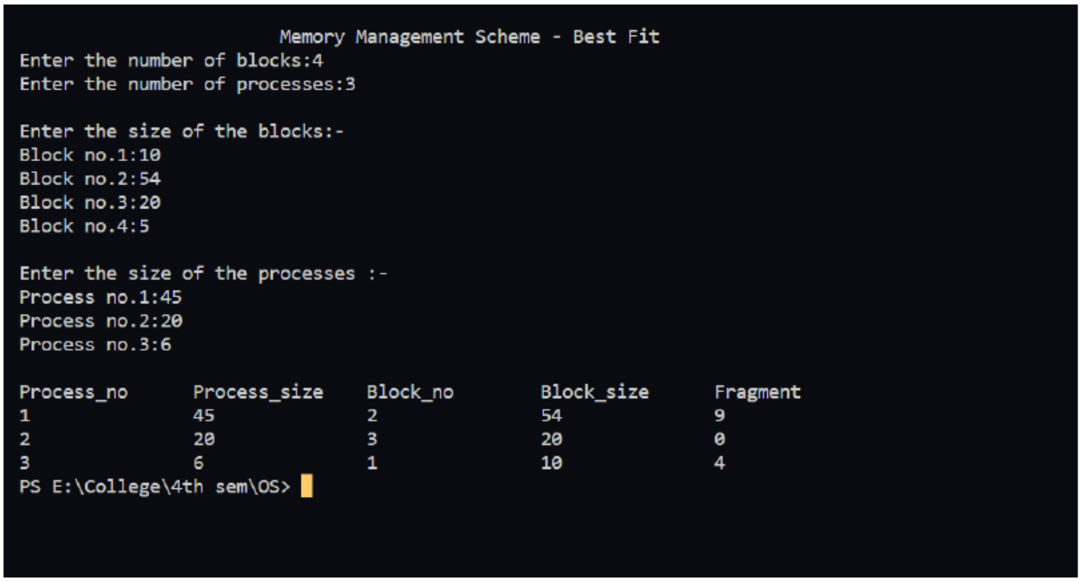


```
Enter no. of blocks: 4

Enter size of each block: 10 12 50 4

Enter no. of processes: 3

Enter size of each process: 6 4 10

Block no.       size            process no.             size
1               10              1                       6
2               12              2                       4
3               50              3                       10
4               4               Not allocated
PS E:\College\4th sem\OS> 
```

Best Fit:

```
#include<stdio.h>
void main()
{
int fragment[20],b[20],p[20],i,j,nb,np,temp,lowest=9999;
static int barray[20],parray[20];
printf("\n\t\t\tMemory Management Scheme - Best Fit");
printf("\nEnter the number of blocks:");
scanf("%d",&nb);
printf("Enter the number of processes:");
scanf("%d",&np);
printf("\nEnter the size of the blocks:-\n");
for(i=1;i<=nb;i++)
{
printf("Block no.%d:",i);
scanf("%d",&b[i]);
}
printf("\nEnter the size of the processes :-\n");
for(i=1;i<=np;i++)
{
printf("Process no.%d:",i);
scanf("%d",&p[i]);
}
for(i=1;i<=np;i++)
{
for(j=1;j<=nb;j++)
{
if(barray[j]!=1)
{
temp=b[j]-p[i];
if(temp>=0)
if(lowest>temp)
{
parray[i]=j;
lowest=temp;
}
}
}
fragment[i]=lowest;
barray[parray[i]]=1;
lowest=10000;
}
printf("\nProcess_no\tProcess_size\tBlock_no\tBlock_size\tFragment");
for(i=1;i<=np && parray[i]!=0;i++)
printf("\n%d\t\t%d\t\t%d\t\t%d\t\t%d",i,p[i],parray[i],b[parray[i]],fragm
e
nt[i]);
}
```

Output----



```
                        Memory Management Scheme - Best Fit
Enter the number of blocks:4
Enter the number of processes:3

Enter the size of the blocks:-
Block no.1:10
Block no.2:54
Block no.3:20
Block no.4:5

Enter the size of the processes :-
Process no.1:45
Process no.2:20
Process no.3:6

Process_no      Process_size    Block_no        Block_size      Fragment
1               45              2               54              9
2               20              3               20              0
3               6               1               10              4
PS E:\College\4th sem\OS>
```

Assignment 5

Implement a CPU scheduling algorithms using switch case (in single program file)

1. FCFS
2. SJF
3. SRTF
4. Round Robin
5. Priority

Code:

```
#include <bits/stdc++.h>
using namespace std;
void print(vector<int> v)
{
string space(1, ' ');
for (auto i : v)
cout << i << space << endl;
}
bool sortcol(vector<int> a, vector<int> b)
{
return a[1] < b[1];
}
void fcfs(vector<vector<int>> v)
{
int n = v.size();
vector<int> ct(n), turn(n), wait(n);
sort(v.begin(), v.end(), sortcol);
int c = v[0][0];
for (int i = 0; i < n; i++)
{
c += v[i][1];
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
}
cout << endl;
cout << "Process No.\t"
<< "AT\t\t"
<< "BT\t\t"
<< "CT\t\t"
<< "TAT\t\t"
<< "WT\t\t" << endl;
for (int i = 0; i < n; i++)
{
cout << v[i][2] << "\t\t" << v[i][0] << "\t\t" << v[i][1] << "\t\t" <<
ct[i] << "\t\t" << turn[i] << "\t\t" << wait[i] << endl;
}
}
void sjf(vector<vector<int>> v)
{

int n = v.size();
vector<int> ct(n), turn(n), wait(n);
```

```
int c = 0;
int i;
sort(v.begin(), v.end());
priority_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int,
int>>> pq;
pq.push({v[0][1], 0});
map<int, int> m;
while (!pq.empty())
{
i = pq.top().second;
// cout<<i<<endl;
m[i] = 1;
c += v[i][1];
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
pq.pop();
if (m.size() != n)
for (int j = i + 1; j < n; j++)
if (v[j][0] <= c && !m[j])
pq.push({v[j][1], j});
else
break;

}
cout << "Process No.\t"
<< "AT\t\t"
<< "BT\t\t"
<< "CT\t\t"
<< "TAT\t\t"
<< "WT\t\t" << endl;
for (int i = 0; i < n; i++)
{
cout << v[i][2] << "\t\t" << v[i][0] << "\t\t" << v[i][1] << "\t\t" <<
ct[i] << "\t\t" << turn[i] << "\t\t" << wait[i] << endl;
}
}
void srtf(vector<vector<int>> v)
{
int n = v.size();
vector<int> ct(n), turn(n), wait(n), rem_time(n);
int i, c = 0, count = 0;
sort(v.begin(), v.end());
for (int i = 0; i < n; i++)
rem_time[i] = v[i][1];
priority_queue<pair<int, int>, vector<pair<int, int>>, greater<pair<int,
int>>> pq;

while (count != n)
{
for (int j = 0; j < n; j++)
if (v[j][0] <= c && rem_time[j] > 0)
pq.push({rem_time[j], j});
i = pq.top().second;
pq.pop();
if ((rem_time[i] - 1) < 0)
continue;
c++;
rem_time[i]--;
if (!rem_time[i])
```

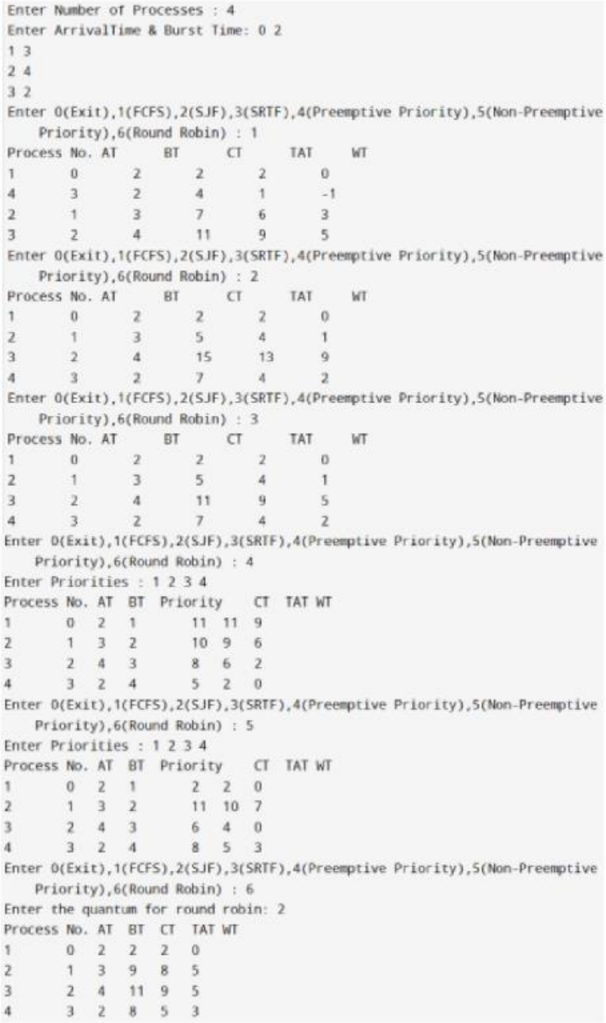
```
{
count++;
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
}
}
cout << "Process No.\t"
<< "AT\t\t"
<< "BT\t\t"
<< "CT\t\t"
<< "TAT\t\t"
<< "WT\t\t" << endl;
for (int i = 0; i < n; i++)
cout << v[i][2] << "\t\t" << v[i][0] << "\t\t" << v[i][1] << "\t\t" <<
ct[i] << "\t\t" << turn[i] << "\t\t" << wait[i] << endl;
}
void prnp(vector<vector<int>> v)
{
int n = v.size();
vector<int> ct(n), turn(n), wait(n);
int c = 0;
int i;
cout << "Enter Priorities : ";
for (int i = 0; i < n; i++)
cin >> v[i][3];
sort(v.begin(), v.end());
priority_queue<pair<int, int>, vector<pair<int, int>>,
function<bool(pair<int, int>, pair<int, int>)>> pq =
priority_queue<pair<int, int>, vector<pair<int, int>>,

function<bool(pair<int, int>, pair<int, int>)>>(

[](pair<int, int> a, pair<int, int> b)
{
if (a.first != b.first)
{
return a.first < b.first;

}
else
{
return a.second > b.second;
}
});
// priority_queue<pii>pq;
vector<pair<int, int>> vv;
// Stores BT and index of vec have<int>g that BT after Sorting
pq.push({v[0][3], 0});
map<int, int> m, vis;
while (!pq.empty())
{
i = pq.top().second;
// cout<<i+1<<endl;
m[i] = 1;
c += v[i][1];
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
pq.pop();
```

```
if (vis.size() != n)
{
for (int j = i + 1; j < n; j++)
if (v[j][0] <= c && !vis[j])
{
pq.push({v[j][3], j});
vis[j] = 1;
}
else
break;

}
}
cout << "Process No.\t"
<< "AT\t"
<< "BT\t"
<< "Priority\t"
<< "CT\t"
<< "TAT\t"
<< "WT\t" << endl;
for (int i = 0; i < n; i++)
{
cout << v[i][2] << "\t\t" << v[i][0] << "\t" << v[i][1] << "\t" <<
v[i][3] << "\t\t" << ct[i] << "\t" << turn[i] << "\
t" << wait[i] << endl;
}
}
void prp(vector<vector<int>> v)

{
int n = v.size();
vector<int> ct(n), turn(n), wait(n), rem_time(n);
int c = 0;
int i;
cout << "Enter Priorities : ";
for (int i = 0; i < n; i++)
cin >> v[i][3];
sort(v.begin(), v.end());
for (int i = 0; i < n; i++)
rem_time[i] = v[i][1];
priority_queue<pair<int, int>, vector<pair<int, int>>,
function<bool(pair<int, int>, pair<int, int>)>> pq =
priority_queue<pair<int, int>, vector<pair<int, int>>,

function<bool(pair<int, int>, pair<int, int>)>>(

[](pair<int, int> a, pair<int, int> b)
{
if (a.first != b.first)
{
return a.first < b.first;
}
else
{
return a.second > b.second;
}
});
// pq.push({v[0][3],0});
map<int, int> m, vis;
int count = 0;
```

```
while (count != n)
{
for (int j = 0; j < n; j++)
if (v[j][0] <= c && rem_time[j] > 0)
pq.push({v[j][3], j});
i = pq.top().second;
pq.pop();
if ((rem_time[i] - 1) < 0)
continue;
c++;
rem_time[i]--;
if (!rem_time[i])
{
count++;
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
}
}

cout << "Process No.\t"
<< "AT\t"
<< "BT\t"
<< "Priority\t"
<< "CT\t"
<< "TAT\t"
<< "WT\t" << endl;
for (int i = 0; i < n; i++)
{
cout << v[i][2] << "\t\t" << v[i][0] << "\t" << v[i][1] << "\t" <<
v[i][3] << "\t\t" << ct[i] << "\t" << turn[i] << "\
t" << wait[i] << endl;
}
}
void rr(vector<vector<int>> v)
{
int n = v.size();
vector<int> ct(n), turn(n), wait(n), rem_time(n);
int i, quan, c = 0;
cout << "Enter the quantum for round robin: ";
cin >> quan;
sort(v.begin(), v.end());
for (int i = 0; i < n; i++)
rem_time[i] = v[i][1];
queue<pair<int, int>> pq;
map<int, int> vis;
int count = 0;
pq.push({rem_time[0], 0});
vis[0] = 1;
while (count != n)
{
i = pq.front().second;
pq.pop();
if (rem_time[i] > quan)
{
c += quan;
rem_time[i] -= quan;
}
else
{
```

```
c += rem_time[i];
rem_time[i] = 0;
}
for (int j = 0; j < n; j++)
if (v[j][0] <= c && rem_time[j] > 0 && !vis[j])
{
pq.push({rem_time[j], j});
vis[j] = 1;

}
if (rem_time[i])
pq.push({rem_time[i], i});
queue<pair<int, int>> qq = pq;
if (!rem_time[i])
{
count++;
ct[i] = c;
turn[i] = ct[i] - v[i][0];
wait[i] = turn[i] - v[i][1];
}
}
cout << "Process No.\t"
<< "AT\t"
<< "BT\t"
<< "CT\t"
<< "TAT\t"
<< "WT\t" << endl;
for (int i = 0; i < n; i++)
{
cout << v[i][2] << "\t\t" << v[i][0] << "\t" << v[i][1] << "\t" <<
ct[i] << "\t" << turn[i] << "\t" << wait[i] << endl;
}
}
int main()
{
cout << " Enter Number of Processes : ";
int n;
cin >> n;
vector<vector<int>> ab(n, vector<int>(4));
cout << "Enter ArrivalTime & Burst Time: ";
for (int i = 0; i < n; i++)
{
cin >> ab[i][0] >> ab[i][1];
ab[i][2] = i + 1;
}
while (1)
{

cout<<"Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-
Preemptive Priority),6(Round Robin) : ";

int whichalgo;
cin >> whichalgo;
switch (whichalgo)
{
case 1:
{
fcfs(ab);
break;
}
case 2:
{
sjf(ab);
break;
}
case 3:
{
srtf(ab);
break;
}
case 4:
{
prp(ab);
break;
}
case 5:
{
prnp(ab);
break;
}
case 6:
{
rr(ab);
break;
}
case 0:
{
exit(0);
break;
}
default:
{
cout << "Enter Valid Number for Algorithm";
}
}
}
}
```

Output:



```
Enter Number of Processes : 4
Enter ArrivalTime & Burst Time: 0 2
1 3
2 4
3 2
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 1
Process No. AT      BT      CT      TAT     WT
1       0       2       2       2       0
4       3       2       4       1       -1
2       1       3       7       6       3
3       2       4       11      9       5
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 2
Process No. AT      BT      CT      TAT     WT
1       0       2       2       2       0
2       1       3       5       4       1
3       2       4       15      13      9
4       3       2       7       4       2
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 3
Process No. AT      BT      CT      TAT     WT
1       0       2       2       2       0
2       1       3       5       4       1
3       2       4       11      9       5
4       3       2       7       4       2
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 4
Enter Priorities : 1 2 3 4
Process No. AT  BT  Priority    CT  TAT WT
1       0   2   1       11  11  9
2       1   3   2       10  9   6
3       2   4   3       8   6   2
4       3   2   4       5   2   0
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 5
Enter Priorities : 1 2 3 4
Process No. AT  BT  Priority    CT  TAT WT
1       0   2   1       2   2   0
2       1   3   2       11  10  7
3       2   4   3       6   4   0
4       3   2   4       8   5   3
Enter 0(Exit),1(FCFS),2(SJF),3(SRTF),4(Preemptive Priority),5(Non-Preemptive
    Priority),6(Round Robin) : 6
Enter the quantum for round robin: 2
Process No. AT  BT  CT  TAT WT
1       0   2   2   2   0
2       1   3   9   8   5
3       2   4   11  9   5
4       3   2   8   5   3
```

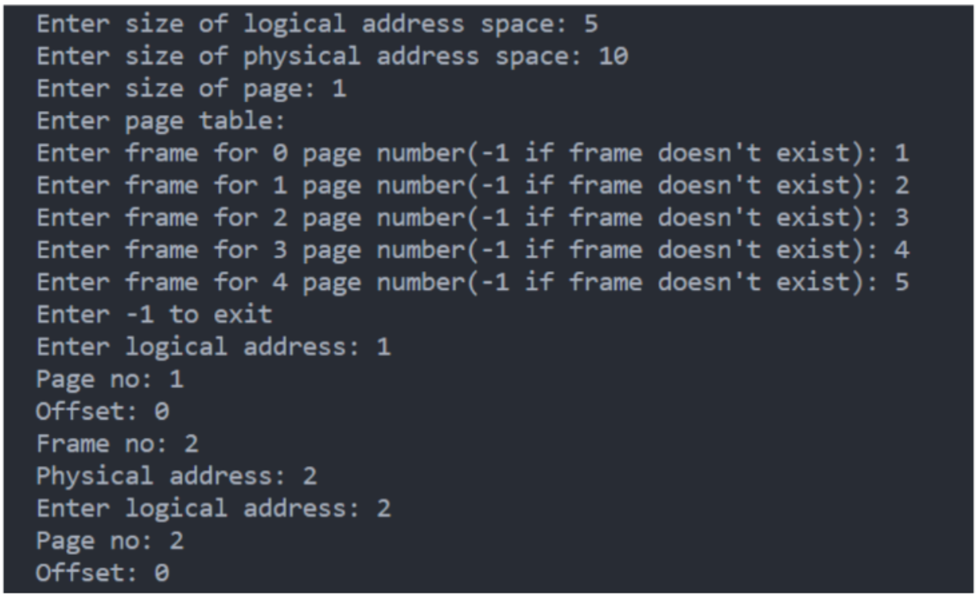
Assignment 6

Implement a program for memory management technique PAGING

Code:

```
#include "stdio.h"
#include "stdbool.h"
#include "stdlib.h"
struct page{
int page_no;
int frame;
};
int main()
{
int size_logical_address,size_physical_address,i,size_of_page,j;
printf("Enter size of logical address space: ");
scanf("%d",&size_logical_address);
printf("Enter size of physical address space: ");
scanf("%d",&size_physical_address);
printf("Enter size of page: ");
scanf("%d",&size_of_page);
int number_of_frames = size_physical_address/size_of_page;
int number_of_pages = size_logical_address/size_of_page;
struct page pageTable[number_of_pages];
printf("Enter page table: \n");
for(i=0;i<number_of_pages;i++)
{
pageTable[i].frame = -1;
}
for(i=0;i<number_of_pages;i++)
{
int frame;
bool replica = false;
pageTable[i].page_no = i;
printf("Enter frame for %d page number(-1 if frame doesn't exist):
",i);
scanf("%d",&frame);
for(j=0;j<number_of_pages;j++)
{
if(frame!= -1 && pageTable[j].frame == frame)
{
replica = true;
printf("Frame number already stored\n");
}
}
if(frame > number_of_frames)

{
replica = true;
printf("Cannot exceed frame size\n");
}
if(replica == false)
{
pageTable[i].frame = frame;
}
}
int logical_address;
printf("Enter -1 to exit\n");
while(1)
{
printf("Enter logical address: ");
scanf("%d",&logical_address);
if(logical_address == -1)
return 0;
int page_no = logical_address/size_of_page;
int offset = logical_address%size_of_page;
if(pageTable[page_no].frame == -1)
{
printf("No such logical address exist\n");
}
else
{
printf("Page no: %d \nOffset: %d\nFrame no: %d\nPhysical address:

%d\n",page_no, offset,
pageTable[page_no].frame,pageTable[page_no].frame*size_of_page + offset );
}
}
}
```

Output:



```
Enter size of logical address space: 5
Enter size of physical address space: 10
Enter size of page: 1
Enter page table:
Enter frame for 0 page number(-1 if frame doesn't exist): 1
Enter frame for 1 page number(-1 if frame doesn't exist): 2
Enter frame for 2 page number(-1 if frame doesn't exist): 3
Enter frame for 3 page number(-1 if frame doesn't exist): 4
Enter frame for 4 page number(-1 if frame doesn't exist): 5
Enter -1 to exit
Enter logical address: 1
Page no: 1
Offset: 0
Frame no: 2
Physical address: 2
Enter logical address: 2
Page no: 2
Offset: 0
```

Assignment 7

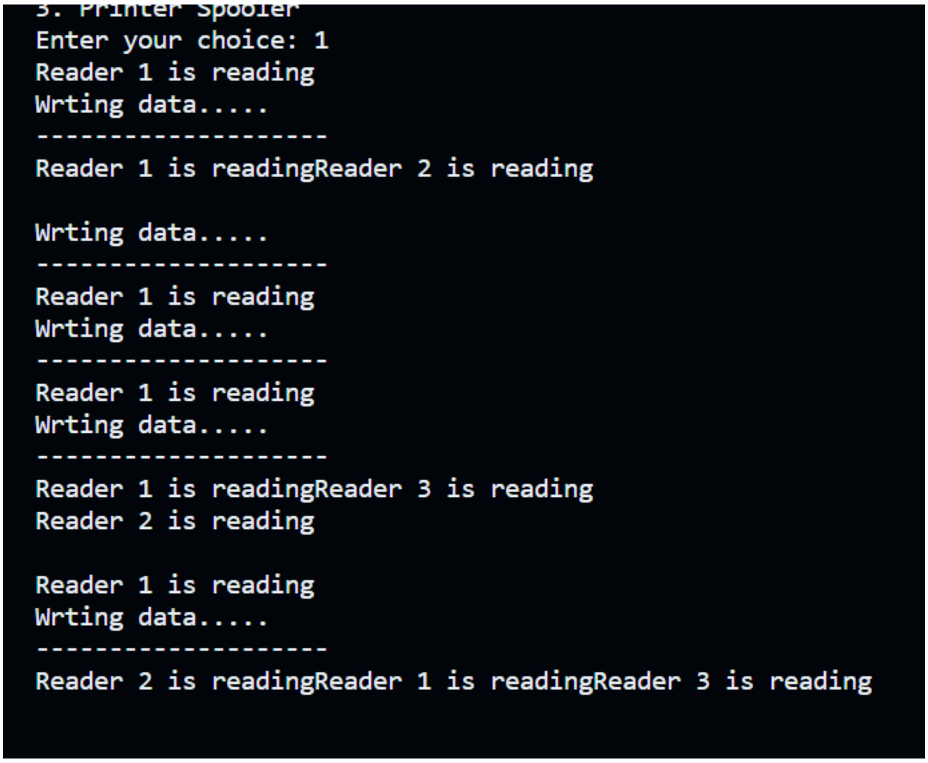
Implement a program for readers writers, producer consumer and printer spooler using semaphore (use switch case or function)

CODE:

```
import random
import threading
import time
import sys

print("0. Exit")
print("1. Reader Writer")
print("2. Producer Consumer")
print("3. Printer Spooler")
choice = int(input("Enter your choice: "))

if choice == 0:
    sys.exit()
elif choice == 1:
    class ReaderWriter():
        def __init__(self):
            # initializing semaphores using Semaphore class in threading
module for reading and wrting
            self.rd = threading.Semaphore()
            self.wrt = threading.Semaphore()

            self.readCount = 0  # initializing number of reader present

        def reader(self):
            while True:
                self.rd.acquire()  # wait on read semaphore

                self.readCount += 1  # increase count for reader by 1

                if self.readCount == 1:  # since reader is present, prevent
writing on data
                    self.wrt.acquire()  # wait on write semaphore

                self.rd.release()  # sinal on read semaphore

                print(f"Reader {self.readCount} is reading")

                self.rd.acquire()  # wait on read semaphore

                self.readCount -= 1  # reading performed by reader hence
decrementing readercount

                if self.readCount == 0:  # if no reader is present allow
writer to write the data
                    self.wrt.release()  # signal on write semphore, now
writer can write
```

```
                self.rd.release()  # sinal on read semaphore

                time.sleep(3)

        def writer(self):
            while True:
                self.wrt.acquire()  # wait on write semaphore

                print("Wrting data.....")  # write the data
                print("-"*20)

                self.wrt.release()  # sinal on write semaphore

                time.sleep(3)

        def main(self):
            # calling mutliple readers and writers
            t1 = threading.Thread(target=self.reader)
            t1.start()
            t2 = threading.Thread(target=self.writer)
            t2.start()
            t3 = threading.Thread(target=self.reader)
            t3.start()
            t4 = threading.Thread(target=self.reader)
            t4.start()
            t6 = threading.Thread(target=self.writer)
            t6.start()
            t5 = threading.Thread(target=self.reader)
            t5.start()

    if __name__ == "__main__":
        c = ReaderWriter()
        c.main()

elif choice == 2:
    # Shared Memory variables
    CAPACITY = 10
    buffer = [-1 for i in range(CAPACITY)]
    in_index = 0
    out_index = 0

    # Declaring Semaphores
    mutex = threading.Semaphore()
    empty = threading.Semaphore(CAPACITY)
    full = threading.Semaphore(0)

    # Producer Thread Class


    class Producer(threading.Thread):
        def run(self):

            global CAPACITY, buffer, in_index, out_index
            global mutex, empty, full

            items_produced = 0
            counter = 0

            while items_produced < 20:
                empty.acquire()
                mutex.acquire()

                counter += 1
                buffer[in_index] = counter
                in_index = (in_index + 1) % CAPACITY
                print("Producer produced : ", counter)

                mutex.release()
                full.release()

                time.sleep(1)

                items_produced += 1

    # Consumer Thread Class

    class Consumer(threading.Thread):
        def run(self):

            global CAPACITY, buffer, in_index, out_index, counter
            global mutex, empty, full

            items_consumed = 0

            while items_consumed < 20:
                full.acquire()
                mutex.acquire()

                item = buffer[out_index]
                out_index = (out_index + 1) % CAPACITY
                print("Consumer consumed item : ", item)

                mutex.release()
                empty.release()

                time.sleep(2.5)

                items_consumed += 1


    # Creating Threads
    producer = Producer()
    consumer = Consumer()

    # Starting Threads
    consumer.start()
    producer.start()

    # Waiting for threads to complete
    producer.join()
    consumer.join()

elif choice==3:
    # Define the maximum number of jobs that can be queued
    MAX_JOBS = 5

    # Define the shared job queue
    job_queue = []
    # Define a semaphore for controlling access to the job queue
    job_queue_mutex = threading.Semaphore(1)
    # Define a semaphore for indicating that the job queue is not empty
    job_queue_not_empty = threading.Semaphore(0)
    # Define a semaphore for indicating that the job queue is not full
    job_queue_not_full = threading.Semaphore(MAX_JOBS)

    # Define the printer thread function


    def printer_thread():
        while True:
            # Acquire the job queue not empty semaphore to wait for a job to
be added to the queue
            job_queue_not_empty.acquire()
            # Acquire the job queue mutex to remove a job from the queue
            job_queue_mutex.acquire()
            # Remove the first job from the queue
            job = job_queue.pop(0)
            # Release the job queue mutex
            job_queue_mutex.release()
            # Release the job queue not full semaphore to signal that there
is now room for another job
            job_queue_not_full.release()

            # Print the job
            print(f"Printing job {job}")
            time.sleep(1)  # Simulate the time it takes to print the job

    # Define the user thread function


    def user_thread():
        global job_queue
        # Generate a job ID (in this case, the current time as an integer)
        job_id = random.randint(1, 50)
        # Acquire the job queue not full semaphore to wait for space to add
the job to the queue
        job_queue_not_full.acquire()
        # Acquire the job queue mutex to add the job to the queue
        job_queue_mutex.acquire()
        # Add the job ID to the end of the queue
        job_queue.append(job_id)
        # Release the job queue mutex
        job_queue_mutex.release()
        # Release the job queue not empty semaphore to signal that there is
now a job in the queue
        job_queue_not_empty.release()

        print(f"Added job {job_id} to the queue.")


    # Create a printer thread and start it
    printer = threading.Thread(target=printer_thread, name="Printer")
    printer.start()

    # Create some user threads to add jobs to the queue
    for i in range(10):
        threading.Thread(target=user_thread, name=f"User {i+1}").start()
```

OUTPUT:



```
3. Printer Spooler
Enter your choice: 1
Reader 1 is reading
Wrting data.....
--------------------
Reader 1 is readingReader 2 is reading

Wrting data.....
--------------------
Reader 1 is reading
Wrting data.....
--------------------
Reader 1 is reading
Wrting data.....
--------------------
Reader 1 is readingReader 3 is reading
Reader 2 is reading

Reader 1 is reading
Wrting data.....
--------------------
Reader 2 is readingReader 1 is readingReader 3 is reading
```

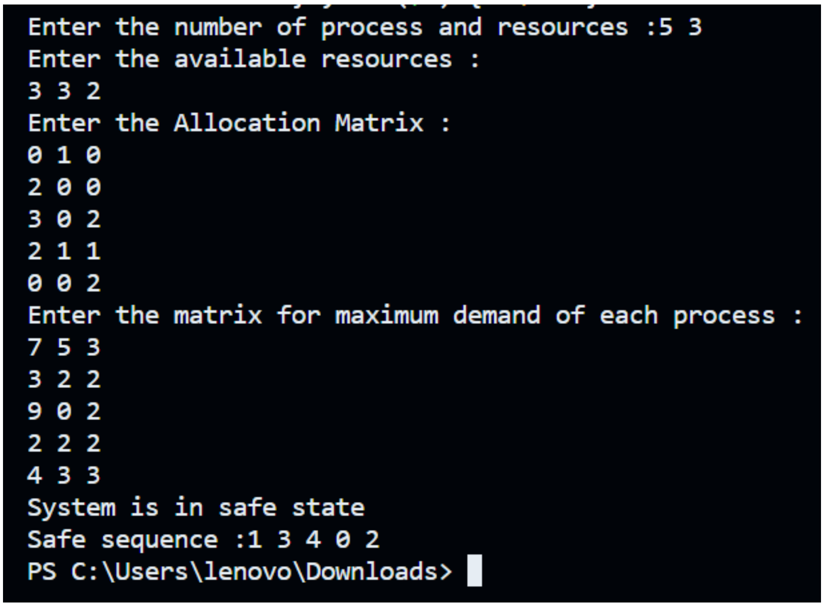
Assignment 8

Implement a code for Banker's Algorithm for Deadlock Avoidance and check for user input system is deadlock free or not.

CODE:

```
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>

int need[100][100],allot[100][100],max[100][100],available[100];
bool isFinished[100];
int sequence[100];

void isSafe(int N,int M)
{
        int i,j,work[100],count=0;
        for(i=0;i<M;i++)
            work[i]=available[i];
        for(i=0;i<100;i++)
            isFinished[i]=false;
        while(count<N)
        {
                bool canAllot=false;
                for(i=0;i<N;i++)
                {
                   if(isFinished[i]==false)
                   {

                        for(j=0;j<M;j++)
                        {
                            if(work[j]<need[i][j])
                            {
                                break;
                            }
                        }
                        if(j==M)
                        {
                            for(j=0;j<M;j++)
                            {
                                work[j]+=allot[i][j];
                            }

                            sequence[count++]=i;
                            isFinished[i]=true;
                            canAllot=true;
                        }
                   }
                }
                if(canAllot==false)
                {
                    printf("System Is  not safe\n");
                    return ;
                }
        }

        printf("System is in safe state\n");

        printf("Safe sequence :");
        for(i=0;i<N;i++)
            printf("%d ",sequence[i]);
        printf("\n");
}

int main()
{
        int i,j,N,M;
        printf("Enter the number of process and resources :");
        scanf("%d %d",&N,&M);

        printf("Enter the available resources :\n");

        for(i=0;i<M;i++)
            scanf("%d",&available[i]);

        printf("Enter the Allocation Matrix :\n");

        for(i=0;i<N;i++)
            for(j=0;j<M;j++)
                scanf("%d",&allot[i][j]);

        printf("Enter the matrix for maximum demand of each process :\n");

        for(i=0;i<N;i++)
            for(j=0;j<M;j++)
                scanf("%d",&max[i][j]);

        //calculation of need matrix
        for(i=0;i<N;i++)
            for(j=0;j<M;j++)
                need[i][j]=max[i][j]-allot[i][j];
        isSafe(N,M);
}
```

OUTPUT:



```
Enter the number of process and resources :5 3
Enter the available resources :
3 3 2
Enter the Allocation Matrix :
0 1 0
2 0 0
3 0 2
2 1 1
0 0 2
Enter the matrix for maximum demand of each process :
7 5 3
3 2 2
9 0 2
2 2 2
4 3 3
System is in safe state
Safe sequence :1 3 4 0 2
PS C:\Users\lenovo\Downloads>
```

Assignment 10

Implement a program for disk scheduling algorithm and state which is better for number of head movements

1. FCFS
2. SSTF
3. SCAN
4. C-SCAN
5. LOOK
6. C-LOOK

Code:

```
/*
    Disk Scheduling
    C-Look
*/
#include "stdio.h"
#include "stdlib.h"
#include "stdbool.h"

struct request
{
    int request_track_number;
    bool visited;
};

int clook()
{
    int i, no_of_requests, initial_head, limit, j, choice, previous_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    struct request req[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &req[i].request_track_number);
        req[i].visited = false;
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);

    printf("Enter the previous position of R/W head: ");
    scanf("%d", &previous_head);

    printf("Enter the cylinder size: ");
    scanf("%d", &limit);

    if (previous_head - initial_head > 0)
    {
        choice = 2;
    }
    else
        choice = 1;
    // scanf("%d",&choice);
    int seek_time = 0;
    printf("%d -> ", initial_head);
    int cp_initial_head = initial_head;
    if (choice == 1)
    {
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        initial_head = 0;
        for (i = 0; i < cp_initial_head; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    else if (choice == 2)
    {
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        initial_head = limit - 1;
        for (i = limit; i > cp_initial_head; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    printf("Seek Time: %d\n", seek_time);
}

int sstf()
{
    int i, no_of_requests, initial_head, limit, j, choice, previous_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    struct request req[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &req[i].request_track_number);
        req[i].visited = false;
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);

    int seek_time = 0;
    printf("%d -> ", initial_head);
    int n = no_of_requests;
    while (n)
    {
        int min = 1e9;
        int min_track_number, position;
        for (i = 0; i < no_of_requests; i++)
        {
            if (abs(initial_head - req[i].request_track_number) < min &&
req[i].visited == false)
            {
                min = abs(initial_head - req[i].request_track_number);
                min_track_number = req[i].request_track_number;
                position = i;
            }
        }
        initial_head = req[position].request_track_number;
        req[position].visited = true;
        printf("%d ->", min_track_number);
        seek_time += min;
        n--;
    }

    printf("\nSeek Time: %d\n", seek_time);
}

int scan()
{
    int i, no_of_requests, initial_head, limit, j, choice, previous_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    struct request req[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &req[i].request_track_number);
        req[i].visited = false;
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);

    printf("Enter the previous position of R/W head: ");
    scanf("%d", &previous_head);

    printf("Enter the cylinder size: ");
    scanf("%d", &limit);

    if (previous_head - initial_head > 0)
    {
        choice = 2;
    }
    else
        choice = 1;
    // scanf("%d",&choice);
    int seek_time = 0;
    printf("%d -> ", initial_head);
    if (choice == 1)
    {
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("%d -> ", limit - 1);
        seek_time += abs(limit - 1 - initial_head);
        initial_head = limit - 1;
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        seek_time += abs(initial_head - 0);
        printf("0 \n");
    }
    else if (choice == 2)
    {
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("%d -> ", 0);
        seek_time += abs(0 - initial_head);
        initial_head = 0;
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        seek_time += abs(limit - 1 - initial_head);
        printf("%d \n", limit - 1);
    }
    printf("Seek Time: %d\n", seek_time);
}

int cscan()
{
    int i, no_of_requests, initial_head, limit, j, choice, previous_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    struct request req[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &req[i].request_track_number);
        req[i].visited = false;
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);
    printf("Enter the previous position of R/W head: ");
    scanf("%d", &previous_head);

    printf("Enter the cylinder size: ");
    scanf("%d", &limit);

    if (previous_head - initial_head > 0)
    {
        choice = 2;
    }
    else
        choice = 1;
    // scanf("%d",&choice);
    int seek_time = 0;
    printf("%d -> ", initial_head);
    int cp_initial_head = initial_head;
    if (choice == 1)
    {
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("%d -> \n", limit - 1);
        seek_time += abs(limit - 1 - initial_head);
        initial_head = 0;
        for (i = 0; i < cp_initial_head; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    else if (choice == 2)
    {
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("%d -> ", 0);
        seek_time += abs(initial_head - 0);
        initial_head = limit - 1;
        for (i = limit; i > cp_initial_head; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    printf("Seek Time: %d\n", seek_time);
}

int fcfs()
{
    int i, no_of_requests, initial_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    int request[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &request[i]);
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);
    int seek_time = 0;
    printf("%d -> ", initial_head);
    for (i = 0; i < no_of_requests; i++)
    {
        if (i == no_of_requests - 1)
            printf("%d\n", request[i]);
        else
            printf("%d -> ", request[i]);
        seek_time += abs(request[i] - initial_head);
        initial_head = request[i];
    }
    printf("Seek Time: %d\n", seek_time);
}
```

```
int lookscan()
{
    int i, no_of_requests, initial_head, limit, j, choice, previous_head;
    printf("Enter the number of requests: ");
    scanf("%d", &no_of_requests);
    struct request req[no_of_requests];
    printf("Enter the requests: ");
    for (i = 0; i < no_of_requests; ++i)
    {
        scanf("%d", &req[i].request_track_number);
        req[i].visited = false;
    }
    printf("Enter initial position of R/W head: ");
    scanf("%d", &initial_head);

    printf("Enter the previous position of R/W head: ");
    scanf("%d", &previous_head);

    printf("Enter the cylinder size: ");
    scanf("%d", &limit);

    if (previous_head - initial_head > 0)
    {
        choice = 2;
    }
    else
        choice = 1;
    // scanf("%d",&choice);
    int seek_time = 0;
    printf("%d -> ", initial_head);
    if (choice == 1)
    {
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    else if (choice == 2)
    {
        for (i = initial_head; i >= 0; i--)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        for (i = initial_head; i < limit; i++)
        {
            for (j = 0; j < no_of_requests; j++)
            {
                if (req[j].request_track_number == i && req[j].visited ==
false)
                {
                    printf("%d -> ", req[j].request_track_number);
                    req[j].visited = true;
                    seek_time += abs(req[j].request_track_number -
initial_head);
                    initial_head = req[j].request_track_number;
                }
            }
        }
        printf("\n");
    }
    printf("Seek Time: %d\n", seek_time);
}

int main()
{
    while (true)
    {
        /* code */
        int n;
        printf("1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK \n");
        scanf("%d", &n);
        switch (n)
        {
        case 1:
        {
            fcfs();
        }
        break;

        case 2:
        {
            sstf();
        }
        break;
        case 3:
        {
            scan();
        }
        break;
        case 4:
        {
            cscan();
        }
        break;
        case 5:
        {
            lookscan();
        }
        break;

        case 6:
        {
            clook();
        }
        break;

        default:
            break;
        }
    }
}
```

Output:

```
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
1
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
50 -> 82 -> 170 -> 43
Seek Time: 247
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
2
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
50 -> 43 ->82 ->170 ->
Seek Time: 134
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
3
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
Enter the previous position of R/W head: 50
Enter the cylinder size: 190
50 -> 82 -> 170 -> 189 -> 43 -> 0
Seek Time: 328
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
4
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
Enter the previous position of R/W head: 10
Enter the cylinder size: 190
50 -> 82 -> 170 -> 189 ->
43 ->
Seek Time: 182
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
5
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
Enter the previous position of R/W head: 10
Enter the cylinder size: 190
50 -> 82 -> 170 -> 43 ->
Seek Time: 247
1->FCFS 2->SSTF 3->SCAN 4->C-SCAN 5->LOOK 6->C-LOOK
6
Enter the number of requests: 3
Enter the requests: 82 170 43
Enter initial position of R/W head: 50
Enter the previous position of R/W head: 10
Enter the cylinder size: 190
50 -> 82 -> 170 -> 43 ->
Seek Time: 163
```