Assignment 01_pattern.sh

Read 'n' and generate a pattern given below

1
1 2
1 2 3
1 2 3 4

Pr-requisites:-

- How to run a loops in shell scripts.
- How to execute a bash script.
- How to change execute permission of a file.

Objectives: -

- To understand the working of loops in a script.

Requirements: -

1. Read a value from user
2. Create a pattern as mentioned above

Sample execution: -

1. ./01_pattern.sh
```
1
1  2
1  2  3
1  2  3  4
```

Assignment 02_pattern.sh

Read 'n' and generate a pattern given below

1
2 3
4 5 6
7 8 9 10

Pr-requisites:-

- How to run a loops in shell scripts.
- How to execute a bash script.
- How to change execute permission of a file.

Objectives: -

- To understand the working of loops in a script.

Requirements: -

1. Read a value from user
2. Create a pattern as mentioned above

Sample execution: -

1. ./02_pattern.sh

```
1
2 3
4 5 6
7 8 9 10
```

Assignment 03_real_add.sh

Write a script for addition of two numbers for real numbers also

Pr-requisites:-
- How to add real numbers in script.
- How to use piping in commands.

Objectives: -
- To understand working of piping.
- To learn arithmetic operations in shell script

Requirements: -
1. Ask user to enter two numbers
2. User can enter real numbers also
3. Use bc command and piping to do

Sample execution: -

```
1.  ./03_real_add.sh
    Enter the numbers to addition
    10   20
    Answer is 30
2.  ./03_real_add.sh
    Enter the numbers to addition
    10.32   20.45
    Answer is 30.77
```

Assignment 04_calculator.sh

Write a script for arithmetic calculator using command line arguments

Pr-requisites:-
- How to use command-line arguments in script.
- How to do arithmetic operations in script.
- How to use piping in commands.

Objectives: -
- To understand working of command-line arguments
- To understand working of piping.
- To learn arithmetic operations in shell script

Requirements: -
- User must provide two numbers and operator through command-line
- Based on input do the operation and show the output.
- Use case to handle multiple operations
- Use expr or bc commands

Sample execution: -

1. *./05_arithmatic_calc.sh* ***25 + 41***
   67
2. *./05_arithmatic_calc* ***10 x 5***
   50
3. *./05_arithmatic_calc.sh* ***25 / 5***
   5
4. *./05_arithmatic_calc.sh* ***10.2 – 5.6***
   4.6
5. *./05_arithmatic_calc.sh*
   Please pass the arguments throght command line.

6. *./05_arithmatic_calc.sh 3.4*
   Error:Please pass 3 arguments.
   Usage:./05_arithmatic_calc 2.3 + 6.7

Assignment 05_largest.sh

Write a script to compare larger integer values from a 'n' number of arguments using command line arguments
Pr-requisites:-
- How to use command-line arguments in script.
- How to do integer comparison script.
- How to use loops in scripts.

Objectives: -
- To understand working of command-line arguments
- To understand working of integer comparison in script.

Requirements: -
1. Using command-line pass n arguments.
2. Compare all these arguments and print the largest value
3. Print error in-case no arguments.
4. Number of arguments can vary every time.

Sample execution: -
1. *./06_largest.sh 5 6 8 4 2 1*
   Largest value is 8
2. *./06_largest 9*
   Largest value is 9
3. *./06_largest*
   Error: No arguments passed

Assignment 06_reverse.sh

Write a script to print a given number in reverse order.
Pr-requisites:-
- How to use command-line arguments in script.
- How to do use modules operators in script.
- How to use loops in scripts.

Objectives: -
- To understand working of command-line arguments

- To learn arithmetic operations in shell script

Requirements: -

1. Read an multi-digit number from user and reverse the number.
2. Its not just printing in reverse order
3. You have to extract each digit and convert to reverse.
4. When ‘0’ comes as last digit, discard while reversing.

Sample execution: -

1. *./07_reverse.sh* ***12345***
   Reversed number – 54321
2. *./07_reverse.sh* ***1540***
   Reversed number – 451
3. *./07_reverse.sh* ***5***
   Error: pass a multi-digit number
4. *./07_reverse.sh*
   Error: No argument passed

Assignment 07_delete_empty_lines.sh

Write a script to delete empty lines from a file

Pr-requisites:-

- Knowledge about sed.
- Knowledge about regular-exp.

Objectives: -

- To learn sed command-line
- To learn about regular-exp

Requirements: -

- Pass a filename through command-line.
- Delete all the empty lines from that file and save it back.
- Use sed command to do this

Sample execution: -

1. *./08_delete_empty_lines.sh* ***file.txt***
   Empty lines are deleted.
2. *./08_delete_empty_lines.sh*
   Error: Please pass the file name through command line.

   After this all empty lines must be deleted from given file.

Assignment 08_operator_dependent.sh

Write a script to perform arithmetic operation on digits of a given number depending upon the operator.

Pr-requisites:-

- Knowledge about arrays in script.
- Use of loops.
- How to access elements of a string

Objectives: -

- To learn more string manipulation in scripts.

Requirements: -

1. Read a string from user, must end with a operator symbol.
2. Number can be any length but must end with an operator character
3. Always do left to right operations.
4. If **8312-** passed do **8-3-1-2 = 2**

Sample execution: -

```
1. ./09_operator_dependent.sh 1234+
   Sum is 10
2. ./09_operator_dependent.sh 8312-
   Sub is 2

3. ./09_operator_dependent.sh 5487
   Error: Operator missing
4. ./09_operator_dependent.sh 1111\*
   Mul is 1
5. ./09_operator_dependent.sh
   Error : Please pass the argument.
   Usage : ./09_operator_dependent.sh 2345+
```

Assignment 09_fibonacci.sh

Write a script to read 'n' and generate Fibonacci numbers <= n
Pr-requisites:-

- Knowledge about Fibonacci series.

Objectives: -

- Learn to implement existing algorithms using loops

Requirements: -

1. Remember n is not number of elements to print
2. Its the boundary of elements to print.

Sample execution: -

```
1. ./fibonacci.sh
   Enter limit for fib series: 10
    0, 1, 1, 2, 3, 5, 8
2. ./fibonacci.sh
   Enter limit for fib series: 33
    0, 1, 1, 2, 3, 5, 8, 13, 21, 33
3. ./fibonacci.sh
   Enter limit for fib series: -10
   Error : Please enter only positive numbers.
```

Assignment 10_string_length.sh

Write a script to print the length of each and every string using arrays

Pr-requisites:-

- Knowledge about arrays.
- How to find length of string.
- How to access command-line arguments.

Objectives: -

- To learn more string manipulation in scripts.

Requirements: -

1. Pass some names or strings from command-line.
2. Print all the string lengths one-by-one.
3. Number of argument may vary.

Sample execution: -

1. *./11_string_length.sh* ***hello hai how are you ?***
```
Length of string (hello)  – 5
Length of string (hai)    – 3
Length of string (how)    – 3
Length of string (are)    – 3
Length of string (you)    – 3
Length of string (?)      - 1
```

Assignment 11_chess_board.sh

Write a script to print chess board.

Pr-requisites:-

- Knowledge about printing colors using echo
- Use of nested loops

Objectives: -

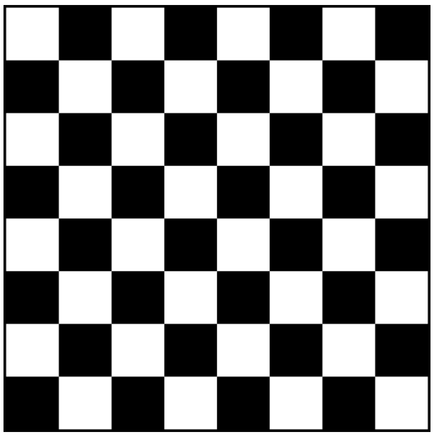
- Print colors using echo command.

Requirements: -

1. To print a black box ***echo -e -n "\e[40m" " "***
2. To print a white box ***echo -e -n "\e[47m" " "***
3. Call the commands in a loop.
4. After 8 columns make to normal color.
5. To make it normal ***echo -e -n "\e[0m" " "***

Sample execution: -

1. *./12_chess_board.sh*



Assignment 12_sorting.sh

Write a script to sort a given number in ascending or descending order.

Pr-requisites:-
- Knowledge about arrays.
- Bubble sort.

Objectives: -
- Learn about sorting mechanisms.
- Better array manipulations.

Requirements: -
1. Pass numbers through command-line arguments.
2. Provide a menu for user to choose ascending or descending.
3. Show sorted array according to user choice.

Sample execution: -

```
1. ./13_sorting.sh -a 5 4 6 2 3 8 9 7 1
   Ascending order of array is 1 2 3 4 5 6 7 8 9
2. ./13_sorting.sh -d 5 4 6 2 3 8 9 7 1
   descending order of array is 9 8 6 5 4 3 2 1
3. ./13_sorting.sh  5 4 6 2 3 8 9 7 1
   Error : Please pass the choice.
   Usage : ./13_sorting -a/-d 4 23 5 6 3
4. 13_sorting.sh
   Error : Please pass the argument through command line.
   Usage : ./13_sorting -a/-d 4 23 5 6 3
```

Assignment 13_system_info.sh

Write a script to print the following:
- Currently logged users
- Your shell directory
- Home directory
- OS name & version
- Current working directory
- Number of users logged in
- Show all available shells in your system
- Hard disk information
- CPU information.
- Memory information.
- File system information.
- Currently running process

Pr-requisites:-
- Knowledge about user commands w, who, whoami
- Bash environmental variables.
- /proc file-system
- Other system info commands like df, du, uname, ps

Objectives: -
- To learn system information commands

Requirements: -
1. Provide a menu for user about what information he wants to check
2. Using switch case display output for selected option.

Sample execution: -

./14_system_info.sh

1. *Currently logged users*
2. *Your shell directory*
3. *Home directory*
4. *OS name & version*
5. *Current working directory*
6. *Number of users logged in*
7. *Show all available shells in your system*
8. *Hard disk information*
9. *CPU information.*
10. *Memory information.*
11. *File system information.*
12. *Currently running process.*

Enter the choice : 2
Your shell directory is /bin/bash

Assignment 14_file_upper_lower.sh

Write a script to rename a file/directory replaced by lower/upper case letters.
WARNING: Dont try this in your **home/** or **assignment/** directory.
Please create a seperate directory to test this script.

Pr-requisites:-

- Knowledge about mv and tr commands.
- To check a file type in script

Objectives: -

- To learn filter/translate commands
- Identifying file types in script

Requirements: -

1. Rename all files from current directory to lowercase letters.
2. Rename all directories from current directories to uppercase.
3. Digits and other symbols should remain same.

Sample execution: -

Before running the script

```
$ ls
File.txt  MyScript.SH  MyFile007.txt  dir/  Assign1/ newfolder/

$ ./15_file_upper_lower
Files are rename in lowercase and directories are renamed in upper case

$ ls
file.txt myfile007.txt myscript.sh DIR/ ASSIGN1/ NEWFOLDER/
```

Assignment 15_rename_cur_dir.sh

Write a script to rename current working directory with given name.
WARNING: Dont try this in your **home/** or **assignment/** directory.
Please create a seperate directory to test this script.

Pr-requisites:-

- Knowledge about mv and tr commands.

Objectives: -

- To learn filter/translate commands

Requirements: -

- After execting this script your current directory will be renamed to given name
- Pass new name through command-line.

Sample execution: -

Before executing the script let assume your current working directory is

```
$ pwd
/home/user/ECEP/Linux_Systems/Test_16

$ ./rename_cur_dir.sh Assign2
```

current directory will be renamed to **Assign2**

```
$ pwd
/home/user/ECEP/Linux_Systems/Test_16

$ ./rename_cur_dir.sh
Error : Please pass the new directory name
```

Assignment 16_rename_album.sh

Given album name and corresponding directory this scripts renames them properly by inserting index numbers. For example given file numbers .
WARNING: Dont try this in your **home/** or **assignmen/** directory.
Please create a separate directory to test this script.

Pr-requisites:-

- Knowledge about mv and tr commands.

Objectives: -

- To learn filter/translate commands

Requirements: -

1. Aim of this project is to rename all files in one directory with a common name and indexing.
2. Usually when we takes pics in camera or mobile default names are like DSN001.jpg, DSN002.jpg
3. These files need to be renamed by user given prefix name
4. Prefix name pass through command-line argument.

Sample execution: -

Before executing the script
$ ls
DSN001.jpg DSN002.jpg DSN003.jpg DSN004.jpg DSN005.jpg

```
./17_rename_album.sh day_out
```
All .jpg files in current directory is renamed as
day_out_001.jpg day_out_002.jpg day_out_003.jpg day_out_004.jpg day_out_005.jpg

```
./17_rename_album.sh
Error : Please pass the prefix name through command line.
```

Assignment 17_print_lines.sh

Write script to print contents of file from given line number to next given number of lines.

Pr-requisites:-
- Piping in shell
- head and tail commands

Objectives: -
- To learn about file filter commands.

Requirements: -
1. Pass three command-line arguments
2. 1- starting line number
3. 2-number of lines and filename
4. Script will print n lines from given starting line

Sample execution: -

```
1. /18_print_lines.sh 5 3 myfile.txt
   line number 5
   line number 6
   line number 7

2. ./18_print_lines.sh myfile.txt
   Error: arguments missing!
   Usage: ./file_filter.sh start_line   uptoline   filename
   For eg. ./20_file_filter.sh 5 5 <file>
```

Assignment 18_largest_username.sh

Display the longest and shortest user-names on the system.

Pr-requisites:-
- Filter commands cut and tr.
- Arrays in script.

- String operations.

Objectives: -
- Learn about etc configuration files.

Requirements: -
1. Fetch user-names from the first field in /etc/passwd file.
2. Print longest and shortest name.

Sample execution: -
```
1. ./19_largest_username.sh
   The Longest Name is:      speech-dispatcher
   The Shortest Name is:     lp
```

Assignment 19_delete_display_swp.sh

Write a script to delete all the .swp files found in your system or directory.

Pr-requisites:-
- Knowledge about find command.

Objectives: -
- Learn various usage of find command.

Requirements: -
1. Find and delete all .swp files (Temperory vi files).
2. If command-line directories are passed delete only from that directories
3. If no arguments passed delete from entire ~/ directory
4. If no file present show a message.

Sample execution: -
```
1. ./20_delete_display_swp.sh
   swp file found :
   /home/user/ConventionalMethod/.pic16F887.as.swp
   /home/user/Development/BeagleBone-Xm/VideoApp/.cmds.swp
   /home/user/Development/LDD/.expt_drv.c.swp
   /home/user/Development/LDD/.ddk_block.c.swp

2. ./20_delete_display_swp.sh
   swp files not found.

3. ./20_delete_display_swp.sh test_swp
   No swp files found in test_swp.
```

Assignment 20_random_password.sh

Write a script for generating random 8-character passwords including alpha numeric characters.

Pr-requisites:-
- Knowledge about rand, tr and cut commands.
- Use of /dev/urandom file.

- Piping

Objectives: -
- Piping between multiple commands.
- Generate random values.

Requirements: -
1. Every time a new password must created.
2. Password must contains a alpha-numeric and special characters.

Sample execution: -
1. *./21_random_password.sh*
```
nH@Rh0Pv
O8ug&HfD
ro0IUJ$f
wx!Kox3U
i3?BkomA
S89R%OA3
#e3v8PzS
d?F8TEo6
Lrb-qvl9
T!ilf1C5
```

Assignment 21_say_hello.sh

Write script called say_hello, which will print greetings based on time.

Pr-requisites:-
- Knowledge about date command.
- Filter commands, cut and tr.
- Bash configuration files.

Objectives: -
- Using time in script
- Understanding bash configuration files.

Requirements: -
1. The script should run as soon as you log-on to system
2. Print greetings based on time as follows.
3. "Good morning" (5 AM – 12 PM)
4. "Good noon" (12 PM – 1 PM)
5. "Good afternoon" (2 PM – 5 PM)
6. "Good evening" (5PM – 9 PM)
7. “Good night” (9 PM – 5 AM)

Sample execution: -

1. When we start shell (whenever you opening new tab or terminal)
```
Good Morning user, Have nice day!
This is Thursday 08 in June of 2017 (10:44:10 AM)
```

Assignment 22_upper_lower.sh

Shell script to convert string lower to upper and upper to lower.

Pr-requisites:-

- Knowledge about tr command

Objectives: -

- Command output translation.

Requirements: -

1. Provide a filename through command-line.
2. Ask user for conversion Lower to Upper / Upper to Lower.

Sample execution: -

```
1. ./23_upper_lower.sh file.txt
   1 – Lower to upper
   2 – Upper to lower
   Please select option : 2
   WHAT IS OS?
   WHAT ARE THE DIFFERENT OS?
   WHEN IS OS USED?
   WHAT IS PARTITION AND ITS USE?
   HOW MANY PARTITIONS CAN BE DONE?

2. ./23_upper_lower.sh file.txt
   1 – Lower to upper
   2 – Upper to lower
   Please select option : 1
   what is os?
   what are the different os?
   when is os used?
   what is partition and its use?
   how many partitions can be done?

3. ./23_upper_lower.sh fle.txt
   Error : file is not exist.
4. ./23_upper_lower.sh file1.txt
   Error : No contents inside the file.
5. ./23_upper_lower.sh
   Error : Please pass the file name through command line.
```

Assignment 23_print_fifth_line.sh

WAS to print the 5th line of a file passes through command line.

Pr-requisites:-

- Piping in shell
- head and tail commands

Objectives: -

- To learn about file filter commands.

Requirements: -

1. Pass the filename through command line.

2. Check file is already exist or not and check the contents are available or not.
3. If the no.of lines is less than 5 then print the error.

Sample execution:

1. *./25_print_fifth_line* ***file.txt***
   Fifth line of file.txt is line 5

2. *./25_print_fifth_line* ***file1.txt***
   Error : file1.txt having only 2 lines. So can't print 5th line.
3. *./25_print_fifth_line* ***file2.txt***
   Error : No such a file.
4. *./25_print_fifth_line*
   Error : Please pass the file name in command line.

Assignment 24_redirection.sh

Use pipes or redirection to create an infinite feedback loop.

Pr-requisites:-
- Knowledge about piping and redirection.
- Use of tail command with follow option.

Objectives: -
- Learn about following a file.
- Redirection

Requirements: -
1. The final output becomes the input again to the command line.
2. Be alert, remember to stop this command before it fills your hard disk.
3. Look at the documentation for the tail command

Sample execution: -

1.
```
./redirection.sh
Before loop file contents
Hai hello
After loop file contents
Hai hello
Hai hello
Hai hello
Hai hello
Hai hello
Hai hello
Hai hello
.
.
.
```

Assignment 25_recusion.sh

Use a recursive function to print each argument passed to the function.
Pr-requisites:-

- Working of functions in script.
- Argument passing to functions.
- Recursive functions.

Objectives: -

- Learn more about functions

Requirements: -

1. We pass command-line arguments to script.
2. Script call function with same arguments.
3. Regardless of how many arguments are passed. You are allowed to echo only the first positional argument (echo $1).

Sample execution: -

```
1. ./26_recursion.sh 5 2 4 1 n kj i
   5
   2
   4
   1
   n
   kj
   i
2. ./26_recursion.sh
   Error : Pass the arguments through command line.
```

Assignment 26_mounted_fs.sh

Write a script to determine whether a given file system or mount point is mounted.

Pr-requisites:-

- Must know commands df, tr and cut
- Use of arrays and loops.

Objectives: -

- Learn more about mounting, file-systems and device files.

Requirements: -

1. Check that given file-system is mounted or not
2. If its mounted, print free-space available in it.
3. Other-wise print error message.

Sample execution: -

```
1. ./27_mounted_fs.sh /dev/sda2
   File-system /dev/sda2 is mounted on / and it is having  98%
   used space with 3298220 KB free.
2. ./27_mounted_fs.sh /dev
   /dev is not mounted.
3. ./27_mounted_fs.sh
   Error : Please pass the name of the file-system through
```

command line.

Assignment 27_output_ls.sh

Write a script that takes any number of directories as command-line arguments and then lists the contents of each of these directories.

Pr-requisites:-

- Use of loops.
- Print content of current directory without ls.

Objectives: -

- Accessing various directories using script.

Requirements: -

1. This script will work like a ls command.
2. Don't use ls command.
3. Pass any number of directories through command-line.
4. If no arguments passed, list current directory

Sample execution: -

Let assume your pwd is /home/user/ECEP/Linux_Systems/

1.
   ```
   ./28_output_ls.sh
   Assignments Classwork
   ```

2.
   ```
   ./28_ouput_ls.sh ~ ~/ECEP
   /home/user:
   Downloads Documents Desktop Music Pictures Public Videos
   ECEP

   /home/user/ECEP:
   Linux_Systems Advnc_C Linux_Internals Data_Structures MC
   ```

3.
   ```
   ./28_ouput_ls.sh Test
   28_output_ls.sh: Cannot access 'Test' : No such a file or
   directory.
   ```

Assignment 28_lock_permissions.sh

Write a script to locks down file permissions for a particular directory.

Pr-requisites:-

- Must know working of chmod command.

Objectives: -

- Learn about file permissions.

Requirements: -

1. Remove all permissions for groups and others.
2. Provide directory name through command-line.
3. After running script all files in the given directory, Only should have all the permissions.

4. But remember dont add any permission to user only change to others and groups.

Sample execution: -

```
1. ./29_lock_permissions.sh Dir/

   Before locking
   total 0
   -rw-rw-r-- 1 biju biju 0 Jun  8 12:37 D2file1
   -rw-rw-r-- 1 biju biju 0 Jun  8 12:37 D2file2
   -rw-rw-r-- 1 biju biju 0 Jun  8 12:37 D2file3

   After locking
   total 0
   -rw------- 1 biju biju 0 Jun  8 12:37 D2file1
   -rw------- 1 biju biju 0 Jun  8 12:37 D2file2
   -rw------- 1 biju biju 0 Jun  8 12:37 D2file3
2. ./29_lock_permissions.sh
   Error : Please pass the directory in command line
```

Assignment 29_free_space.sh

Display the names of any file-system which have less than 10% free space available

Pr-requisites:-
- Must know commands df, tr and cut
- Use of arrays and loops.

Objectives: -
- Learn more about mounting, file-systems and device files.

Requirements: -
1. When you run the script show all file-system present in system.
2. Then print file-systems that have only 10% memory remaining.

Sample execution: -

```
1. ./29_free_space.sh
   Filesystem /dev/sda5 have less than 10% freespace
```

Assignment 30_print_user_ids.sh

Count the number of users with user IDs between 500 and 10000 on the system

Pr-requisites:-
- Must know df, cut & tr commands.
- Loops and arrays.

Objectives: -
- Learn about etc configuration files.

Requirements: -
1. Fetch user-ids from the in /etc/passwd file.
2. Display only usernames between the range.
3. User can change the range using command-line arguments.
4. Default is 500 - 100000

Sample execution: -
1. ./users.sh
   ```
   Total count of user ID between 500 to 10000 is:  2
   ```
2. ./users.sh **0 100**
   ```
   Total count of user ID between 0 to 100 is : 3
   ```

Assignment 31_executable_path.sh

For each directory in the $PATH, display the number of executable files in that directory.

Pr-requisites:-
- Must know bash environmental variables.
- Working of tr command.
- Loops and arrays.
- Checking permission of files in script.

Objectives: -
- Learn significance of PATH variable.

Requirements: -
1. Fetch each directories from PATH variable.
2. Use -x option if if condition to check executable permission.
3. Print directory and number of executable files one-by-one.
4. Print the total number of executable files at last.
5. Count only files have executable permission.
6. Verify path is present every-time.

Sample execution: -

1. ./32_executable_path.sh

   ```
   Current dir: /usr/local/sbin
   current count: 0

   Current dir: /usr/local/bin
   current count: 0

   Current dir: /usr/sbin
   current count: 205

   Current dir: /usr/bin
   current count: 1889

   Current dir: /sbin
   ```

```
current count: 187

Current dir: /bin
current count: 159

Current dir: /usr/games
current count: 5

Current dir: /usr/local/games
current count: 0

Total – 2445
```

Assignment 32_user_present.sh

Write a script to search a user present in your system.

Pr-requisites:-
- Must know df, cut & tr commands.
- Loops and arrays.

Objectives: -
- Learn about etc configuration files.

Requirements: -
1. Fetch user-names from the first field in /etc/passwd file.
2. Search given name in the list.

Sample execution: -

1. *./33_user_present.sh xyz*
   xyz not present

2. *./33_user_present.sh root*
   root is present

3. *./33_user_present.sh*
   Error : Please pass the argument through command line.

Assignment 33_replace<DEL>.sh

Write a script to replace 20% lines in a C file randomly and replace it with the pattern <---DEL--->

Pr-requisites:-
- Knowledge about sed command.
- How to create random number.
- Editing file using sed command.

Objectives: -

- Learn more about sed command.

Requirements: -

1. Provede a .c file to this script through command-line.
2. Check the given file is exist or not and check it is having some contents or not.
3. Randomly delete 20% lines from the file.
4. Where ever you deleted replace a string **<-----------Deleted------------>**

Sample execution: -

1. *./33_replace<DEL>.sh main.c*
   Before replacing
   ```
   #incude <stdio.h>
   int main()
   {
       printf("Hello world\n");
   }
   ```
   After replaced
   ```
   #incude <stdio.h>
   int main()
   {
       <-----------Deleted------------>
   }
   ```

2. *./33_replace<DEL>.sh main1.c*
   Error : No such a file.

3. *./33_replace<DEL>.sh main2.c*
   Error : main2.c is empty file. So can't replace the string.

4. *./33_replace<DEL>.sh*
   Error : Please pass the file name through command line.

Assignment 34_BMI.sh

WAS to calculate the BMI.

$$BMI = \frac{weight(kg)}{height(m) * height(m)}$$

Then display the following information.

Underweight	: less than 18.5
Normal	: between 18.5 and 24.9
Overweight	: between 25 and 29.9
Obese	: 30 or greater

Pr-requisites:-

- How to compare the real numbers.
- How to use piping in commands.

Objectives: -

- To understand working of piping.
- To learn arithmetic operations in shell script.

Requirements: -

1. Ask user to enter two numbers
2. User can enter real numbers also
3. Use bc command and piping to do

Sample execution:

Ex:1 ./04_BMI.sh

Enter the weight in Kg : ***45.5***

Enter the height in meters : ***1.5***

Your Normal.