CDAC MUMBAI

Concepts of Operating System

Assignment 2

What will the following commands do?

- echo "Hello, World!"
  Ans: Prints the text "Hello, World!"

```
cdac@LAPTOP-TJIPEEAU:~$ cd LinuxAssignment
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ echo "Hello, World!"
Hello, World!
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- name="Productive"
  Ans: Assigns the string "Productive" to the variable name

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ name="Productive"
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ echo $name
Productive
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- touch file.txt
  Ans : touch is use to create a file.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ touch rs.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      RGS       docs         dublicates.txt  file2.txt   fruts5.txt  newdirectory  output.txt  unique_lines.txt
Docs2.zip  data.txt  docs.tar.gz  file.txt        fruits.txt  input.txt   numbers.txt   rs.txt      vdocs.zip
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- ls -a
  Ans: Lists all files and directories in the current directory, including hidden files (those starting with a dot .).

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -a
.       Docs2.zip  docs           file.txt    fruts5.txt    numbers.txt  unique_lines.txt
..      RGS        docs.tar.gz    file2.txt   input.txt     output.txt   vdocs.zip
Docs2   data.txt   dublicates.txt  fruits.txt  newdirectory  rs.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- rm file.txt
  Ans: Deletes the file named file.txt

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ rm rs.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      RGS       docs         dublicates.txt  file2.txt   fruts5.txt  newdirectory  output.txt       vdocs.zip
Docs2.zip  data.txt  docs.tar.gz  file.txt        fruits.txt  input.txt   numbers.txt   unique_lines.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- cp file1.txt file2.txt
  ans: command used to Copy file1.txt to file2.txt. If file2.txt exists, it will be overwritten the existing file.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ cp file.txt file2.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      RGS       docs         dublicates.txt  file2.txt   fruts5.txt  newdirectory  output.txt       vdocs.zip
Docs2.zip  data.txt  docs.tar.gz  file.txt        fruits.txt  input.txt   numbers.txt   unique_lines.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- mv file.txt /path/to/directory/
  Ans: Moves file.txt to the specified directory

```
/home/cdac
cdac@LAPTOP-TJIPEEAU:~$ touch file45.txt
cdac@LAPTOP-TJIPEEAU:~$ mv /home/cdac/file.txt /home/cdac/LinuxAssignment
mv: cannot stat '/home/cdac/file.txt': No such file or directory
cdac@LAPTOP-TJIPEEAU:~$ mv /home/cdac/file45.txt /home/cdac/LinuxAssignment
cdac@LAPTOP-TJIPEEAU:~$ cd LinuxAssignment
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      RGS       docs         dublicates.txt  file2.txt   fruits.txt  input.txt     numbers.txt  unique_lines.txt
Docs2.zip  data.txt  docs.tar.gz  file.txt        file45.txt  fruts5.txt  newdirectory  output.txt   vdocs.zip
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- chmod 755 script.sh
  Ans: The given command Changes the permissions of file45.txt to 755, giving the owner full read, write, and execute permissions, and giving others read and execute permissions.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ chmod 755 file45.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -l
total 56
drwxrwxr-x 2 cdac cdac 4096 Aug 29 12:06 Docs2
-rw-rw-r-- 1 cdac cdac  675 Aug 29 12:04 Docs2.zip
drwxrwxr-x 3 cdac cdac 4096 Aug 29 12:24 RGS
-rw-rw-r-- 1 cdac cdac  154 Aug 28 21:34 data.txt
drwxrwxr-x 2 cdac cdac 4096 Aug 29 00:29 docs
-rw-rw-r-- 1 cdac cdac  150 Aug 28 23:53 docs.tar.gz
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:12 dublicates.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 28 18:15 file.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 29 23:03 file2.txt
-rwxr-xr-x 1 cdac cdac    0 Aug 29 23:16 file45.txt
-rw-rw-r-- 1 cdac cdac   83 Aug 28 22:22 fruits.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 29 09:32 fruts5.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:56 input.txt
drwxrwxr-x 4 cdac cdac 4096 Aug 29 12:22 newdirectory
-rw-rw-r-- 1 cdac cdac   56 Aug 28 21:47 numbers.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:59 output.txt
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:15 unique_lines.txt
-rw-rw-r-- 1 cdac cdac  316 Aug 29 00:32 vdocs.zip
```

- grep "pattern" file.txt
  Ans: Searches for the string "pattern" in file.txt and displays all matching lines.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      data.txt     dublicates.txt  file45.txt  input.txt     output.txt        vdocs.zip
Docs2.zip  docs         file.txt        fruits.txt  newdirectory  rs45.txt.save
RGS        docs.tar.gz  file2.txt       fruts5.txt  numbers.txt   unique_lines.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ nano vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ grep "pattern" vk18.txt
this file contain some pattern
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ |
```

- kill PID
  ans : Terminates the process with the specified Process ID (PID)

- mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
  ans: The command mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt is a sequence of commands combined using &&, which ensures that each

```
this file contain some pattern
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ cd
cdac@LAPTOP-TJIPEEAU:~$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt
Hello, World!
cdac@LAPTOP-TJIPEEAU:~/mydir$ |
```

- ls -l | grep ".txt"
  Ans: Lists files in the current directory in long format (-l) and filters the output to show only those with .txt in their names.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -l | grep ".txt"
-rw-rw-r-- 1 cdac cdac  154 Aug 28 21:34 data.txt
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:12 dublicates.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 28 18:15 file.txt
-rw-r--r-- 1 cdac cdac    0 Aug 29 23:03 file2.txt
-rwxr-xr-x 1 cdac cdac    0 Aug 29 23:16 file45.txt
-rw-rw-r-- 1 cdac cdac   83 Aug 28 22:22 fruits.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 29 09:32 fruts5.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:56 input.txt
-rw-rw-r-- 1 cdac cdac   56 Aug 28 21:47 numbers.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:59 output.txt
-rw------- 1 cdac cdac   91 Aug 30 01:27 rs45.txt.save
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:15 unique_lines.txt
-rw-rw-r-- 1 cdac cdac   69 Aug 30 01:29 vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ |
```

- cat file1.txt file2.txt | sort | uniq
  Ans: Concatenates file1.txt and file2.txt, sorts the combined output, and removes duplicate lines

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      data.txt     dublicates.txt  file45.txt  input.txt     output.txt        vdocs.zip
Docs2.zip  docs         file.txt        fruits.txt  newdirectory  rs45.txt.save     vk18.txt
RGS        docs.tar.gz  file2.txt       fruts5.txt  numbers.txt   unique_lines.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ cat numbers.txt fruits.txt | sort | uniq
100
111
117
126
131
139
152
156
162
171
205
209
212
264
apple
banana
grape
mango
mango
orange
pear
```

- ```
  *: command not found
  cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -l | grep "^d"
  drwxrwxr-x 2 cdac cdac 4096 Aug 29 12:06 Docs2
  drwxrwxr-x 3 cdac cdac 4096 Aug 29 12:24 RGS
  drwxrwxr-x 2 cdac cdac 4096 Aug 29 00:29 docs
  drwxrwxr-x 4 cdac cdac 4096 Aug 29 12:22 newdirectory
  cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ |
  ```
  show only
  ```

- grep -r "pattern" /path/to/directory/
  Ans: The command grep -r "pattern" /path/to/directory/ is used to search for a specific text pattern within all files in a directory and its subdirectories.
  Command: grep -r "pattern" /home/cdac/LinuxAssignment/

```
cdac@LAPTOP-TJIPEEAU:~$ grep -r "pattern" /home/cdac/LinuxAssignment/
/home/cdac/LinuxAssignment/vk18.txt:this file contain some pattern
/home/cdac/LinuxAssignment/rs45.txt.save:this line contain pattern please wait for it
cdac@LAPTOP-TJIPEEAU:~$ S
```

- cat file1.txt file2.txt | sort | uniq –d
  ans: Concatenates file1.txt and file2.txt, sorts the combined output, and displays only duplicate lines

```
grep: /vk18/rp17/LinuxAssignment: No such file or directory
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ cat numbers.txt fruits.txt | sort | uniq -d
apple
banana
grape
mango
orange
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- chmod 644 file.txt
  Ans: the Chmod 644 Changes the permissions of file.txt to 644, giving the owner read and write permissions, and giving others read-only permissions

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ chmod 644 vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -l
total 64
drwxrwxr-x 2 cdac cdac 4096 Aug 29 12:06 Docs2
-rw-rw-r-- 1 cdac cdac  675 Aug 29 12:04 Docs2.zip
drwxrwxr-x 3 cdac cdac 4096 Aug 29 12:24 RGS
-rw-rw-r-- 1 cdac cdac  154 Aug 28 21:34 data.txt
drwxrwxr-x 2 cdac cdac 4096 Aug 29 00:29 docs
-rw-rw-r-- 1 cdac cdac  150 Aug 28 23:53 docs.tar.gz
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:12 dublicates.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 28 18:15 file.txt
-rw-r--r-- 1 cdac cdac    0 Aug 29 23:03 file2.txt
-rwxr-xr-x 1 cdac cdac    0 Aug 29 23:16 file45.txt
-rw-rw-r-- 1 cdac cdac   83 Aug 28 22:22 fruits.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 29 09:32 fruts5.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:56 input.txt
drwxrwxr-x 4 cdac cdac 4096 Aug 29 12:22 newdirectory
-rw-rw-r-- 1 cdac cdac   56 Aug 28 21:47 numbers.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:59 output.txt
-rw------- 1 cdac cdac   91 Aug 30 01:27 rs45.txt.save
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:15 unique_lines.txt
-rw-rw-r-- 1 cdac cdac  316 Aug 29 00:32 vdocs.zip
-rw-r--r-- 1 cdac cdac   69 Aug 30 01:29 vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- cp -r source_directory destination_directory
  Ans: cp -r source_directory destination_directory makes a complete copy of source_directory and everything inside it, putting the copy in destination_directory
  Command: cp -r mydir LinuxAssignment

```
-rw-rw-r-- 1 cdac cdac 14 Aug 30 01:39 file.txt
cdac@LAPTOP-TJIPEEAU:~/mydir$ nano file.txt
cdac@LAPTOP-TJIPEEAU:~/mydir$ cp -r mydir LinuxAssignment
cp: cannot stat 'mydir': No such file or directory
cdac@LAPTOP-TJIPEEAU:~/mydir$ cd
cdac@LAPTOP-TJIPEEAU:~$ cp -r mydir LinuxAssignment
cdac@LAPTOP-TJIPEEAU:~$ cd LinuxAssignment
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls
Docs2      data.txt     dublicates.txt  file45.txt  input.txt     numbers.txt    unique_lines.txt
Docs2.zip  docs         file.txt        fruits.txt  mydir         output.txt     vdocs.zip
RGS        docs.tar.gz  file2.txt       fruts5.txt  newdirectory  rs45.txt.save  vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- find /path/to/search -name "*.txt"
  Ans: Searches for all files with a .txt extension within the specified directory and its subdirectories.

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ find /home/cdac/LinuxAssignment/ -name "*.txt"
/home/cdac/LinuxAssignment/RGS/Docs2/tre.txt
/home/cdac/LinuxAssignment/RGS/Docs2/tile3.txt
/home/cdac/LinuxAssignment/vk18.txt
/home/cdac/LinuxAssignment/newdirectory/docs/file2.txt
/home/cdac/LinuxAssignment/newdirectory/Docs2/tre.txt
/home/cdac/LinuxAssignment/newdirectory/Docs2/tile3.txt
```

- chmod u+x file.txt
  Ans: Adds execute permission for the owner (user) of file.txt

```
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ chmod u+x file.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ ls -l
total 64
drwxrwxr-x 2 cdac cdac 4096 Aug 29 12:06 Docs2
-rw-rw-r-- 1 cdac cdac  675 Aug 29 12:04 Docs2.zip
drwxrwxr-x 3 cdac cdac 4096 Aug 29 12:24 RGS
-rw-rw-r-- 1 cdac cdac  154 Aug 28 21:34 data.txt
drwxrwxr-x 2 cdac cdac 4096 Aug 29 00:29 docs
-rw-rw-r-- 1 cdac cdac  150 Aug 28 23:53 docs.tar.gz
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:12 dublicates.txt
-rwxrw-r-- 1 cdac cdac    0 Aug 28 18:15 file.txt
-rw-r--r-- 1 cdac cdac    0 Aug 29 23:03 file2.txt
-rwxr-xr-x 1 cdac cdac    0 Aug 29 23:16 file45.txt
-rw-rw-r-- 1 cdac cdac   83 Aug 28 22:22 fruits.txt
-rw-rw-r-- 1 cdac cdac    0 Aug 29 09:32 fruts5.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:56 input.txt
drwxrwxr-x 4 cdac cdac 4096 Aug 29 12:22 newdirectory
-rw-rw-r-- 1 cdac cdac   56 Aug 28 21:47 numbers.txt
-rw-rw-r-- 1 cdac cdac   21 Aug 28 21:59 output.txt
-rw------- 1 cdac cdac   91 Aug 30 01:27 rs45.txt.save
-rw-rw-r-- 1 cdac cdac   65 Aug 28 22:15 unique_lines.txt
-rw-rw-r-- 1 cdac cdac  316 Aug 29 00:32 vdocs.zip
-rw-r--r-- 1 cdac cdac   69 Aug 30 01:29 vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

- echo $PATH
  Ans: echo $PATH shows the list of folders (or path) where your computer looks for programs to run when we used this command.

```
-rw-r--r-- 1 cdac cdac   69 Aug 30 01:29 vk18.txt
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ echo $path

cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/snap/bin
cdac@LAPTOP-TJIPEEAU:~/LinuxAssignment$
```

Identify True or False:

1. **ls** is used to list files and directories in a directory.

   Ans: True
2. **mv** is used to move files and directories.
   Ans: True
3. **cd** is used to copy files and directories.
   Ans : False
4. **pwd** stands for "print working directory" and displays the current directory.\
   ans: True
5. **grep** is used to search for patterns in files.
   Ans: True

6. **chmod 755 file.txt** gives read, write, and execute permissions to the owner, and read and execute permissions to group and others.

   Ans: True
7. **mkdir -p directory1/directory2** creates nested directories, creating directory2 inside directory1 if directory1 does not exist.
   Ans: True
8. **rm -rf file.txt** deletes a file forcefully without confirmation.
   Ans: True

Identify the Incorrect Commands:

1. **chmodx** is used to change file permissions. **False**

   Ans: chmod
2. **cpy** is used to copy files and directories. **False**
   Ans: cp
3. **mkfile** is used to create a new file. **false**
   Ans: nano filename, cat, touch
4. **catx** is used to concatenate files. **False**
   Ans: cat
5. **rn** is used to rename files**. true**
   Ans: True

Part C

Question 1: Write a shell script that prints "Hello, World!" to the terminal.

```
bash: h: No such file or directory
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano h.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash h
bash: h: No such file or directory
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash h.sh
HELLO WORLD!
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 2: Declare a variable named "name" and assign the value "CDAC Mumbai" to it. Print the value of the variable.
Command: name=”CDAC Mumbai”
Command: echo $name

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ name=”CDAC Mumbai”
Mumbai”: command not found
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ name=”CDACMumbai”
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ echo $name
”CDACMumbai”
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 3: Write a shell script that takes a number as input from the user and prints it.
Command: nano qes3.sh
This command for crate file and write shell script.
#! /bin/bash

```
echo "Enter number"

read number
    echo $number
```

Command: bash qes3.sh

```
"CDACMumbai"
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qes3.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qes3.sh
Enter number
45
45
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 4: Write a shell script that performs addition of two numbers (e.g., 5 and 3) and prints the result.

Command: nano qes4.sh

This command for crate file and we write script that performs addition of two numbers

```
#! /bin/bash
a=5
b=3

c=$[a+b]
    echo $c
```

Command: echo $name

This command for show output on terminal

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qes4.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qes4.sh
8
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 5: Write a shell script that takes a number as input and prints "Even" if it is even, otherwise prints "Odd".

Command: nano evenoddq4.sh

This command for crate file and write shell script.

```
#!/bin/bash
Echo "Enter a number: "
Read num
if [ $((num % 2)) -eq 0 ];
 then
   echo "the given number is Even"
else
   echo "the given number is Odd"
     fi
```

Command: bash evenoddq4.sh

Command use for perform execution of code in terminal

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano evenoddq4.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash evenoddq4.sh
Enter a number:
45
the given number is Odd
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash evenoddq4.sh
Enter a number:
22
the given number is Even
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

Command: nano qestion6.sh
This command for crate file in this file we right program
#! /bin/bash

For ((i = 1; i<=5; i++))

do
echo $i
done
command: bash qestion6.sh
for output on terminal

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qestion6.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion6.sh
1
2
3
4
5
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 7: Write a shell script that uses a while loop to print numbers from 1 to 5.

Command: nano qestion7.sh

#! /bin/bash

i=1

while [$i -le 5]

do

echo $i

i=$(($i+1))

done

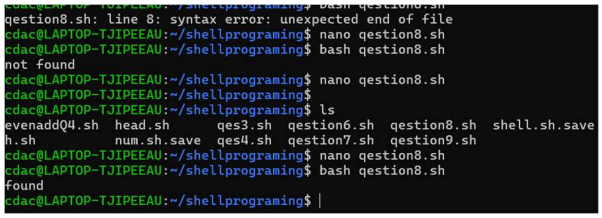
Command: bash qestion7.sh

```
qestion7.sh: line 4: [: syntax error: `-' unexpected
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qestion7.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion7.sh
1
2
3
4
5
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 8: Write a shell script that checks if a file named "file.txt" exists in the current directory. If it does, print "File exists", otherwise, print "File does not exist".

Command:

```
nano qestion8.sh
#! /bin/bash
f="h.sh"
if [ -f "$f" ]
then
        echo "found"
else
        echo "not found"
fi
bash qestion8.sh
```



Question 9: Write a shell script that uses the if statement to check if a number is greater than 10 and prints a message accordingly.

Command:

```
nano qestion9.sh

#! /bin/bash

echo "enter a number:"

read number

if [ $number -gt 10 ]

then

        echo " $number is greater than 10"

else

        echo " $number is less than 10"

fi

bash qestion9.sh
```

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qestion9.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion9.sh
enter a number:
14
the given number is greater than 10
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion9.sh
enter a number:
8
the given number is less than 10
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qestion9.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion9.sh
enter a number:
4
 4 is less than 10
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion9.sh
enter a number:
45
 45 is greater than 10
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 10: Write a shell script that uses nested for loops to print a multiplication table for numbers from 1 to 5. The output should be formatted nicely, with each row representing a number and each column representing the multiplication result for that number.

Command: nano q10.sh

```
#!/bin/bash
for ((i=1; i<=5; i++))
do
    for ((j=1; j<=5; j++))
    do
        printf "%4d" "$(( i * j ))"
    done
    echo
done
```

```
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano q10.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash q10.sh
   1   2   3   4   5
   2   4   6   8  10
   3   6   9  12  15
   4   8  12  16  20
   5  10  15  20  25
cdac@LAPTOP-TJIPEEAU:~/shellprograming$
```

Question 11: Write a shell script that uses a while loop to read numbers from the user until the user enters a negative number. For each positive number entered, print its square. Use the **break** statement to exit the loop when a negative number is entered.

Command:nano question11.sh

```
#!/bin/bash
while true
 do
   echo  "Enter a number: "
   read number

   if [ $number -lt 0 ]
 then
      echo "Negative number entered. Exit"
      break
   fi
   square=(( number * number ))
   echo "The square of $number is $square"
```

done

```
Negative number entered. Exit
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ bash qestion11.sh
Enter a number:
14
The square of 14 is 196
Enter a number:
45
The square of 45 is 2025
Enter a number:
56
The square of 56 is 3136
Enter a number:
-54
Negative number entered. Exit
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ nano qestion11.sh
cdac@LAPTOP-TJIPEEAU:~/shellprograming$ |
```

Part E

Formulas:

1. **waiting time of process = CPU allocation – Arrival Time**
2. **completion time = time taken by the process complete**
3. **TAT = complete time – Arrival time**

1. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	5
P2	1	3
P3	2	6

Calculate the average waiting time using First-Come, First-Served (FCFS) scheduling.

PID	Arrival Time	Burst time	wait time	TAT	ct
p1	0	5	0	5	5
p2	1	3	4	7	8
p3	2	6	6	12	14
			3.33333	8	9

Gant chart

avg wt = 0+7+1+1/4

3.33

	p1	p2	p3	
0		5	8	14

Waiting Time of Process= CPU Allocation-Arrival Time

Avg Waiting Time = Sum WT of All process / no. of processes

2. Consider the following processes with arrival times and burst times:

Process	Arrival Time	Burst Time
P1	0	3
P2	1	5
P3	2	1
P4	3	4

Calculate the average turnaround time using Shortest Job First (SJF) scheduling.

PID	Arrival Time	Burst time	wait time	TAT	ct
p1	0	3	0	3	3
p2	1	5	7	12	13
p3	2	1	1	2	4
p4	3	4	1	5	8
			2.25	5.5	7

avg TAT = 3+12+2+5/4= 5.5

Gant chart

p1	p3	p4	p2	
0	3	4	8	13

3 Consider the following processes with arrival times, burst times, and priorities (lower number indicates higher priority):

Process	Arrival Time	Burst Time	Priority
P1	0	6	3
P2	1	4	1
P3	2	7	4
P4	3	2	2

Calculate the average waiting time using Priority Scheduling.

PID	Arrival Time	Burst time	priority	wait time	TAT	ct
p1	0	6	3	7	12	12
p2	1	4	1	0	4	5
p3	2	7	4	10	17	19
p4	3	2	2	2	4	7
				4.75	9.25	10.75

Gant chart

avg wt = 7+0+10+2/4 =4.75

	p1	p2	p4	p1	p3
0	1	5	7	12	19

4.Consider the following processes with arrival times and burst times, and the time quantum for Round Robin scheduling is 2 units:

Process	Arrival Time	Burst Time
P1	0	4
P2	1	5
P3	2	2
P4	3	3

Calculate the average turnaround time using Round Robin scheduling

PID	Arrival Time	Burst time	wait time	TAT	ct
p1	0	5	10	16	16
p2	1	4	7	11	12
p3	2	2	2	4	6
p4	3	3	7	11	14
			6.5	10.5	

avg tat = 16+11+4+11 = 10.5

Gant chart

	p1	p2	p3	p4	p1	p2	p4	p1
0	2	4	6	8	10	12	14	16

5.Consider a program that uses the **fork()** system call to create a child process. Initially, the parent process has a variable **x** with a value of 5. After forking, both the parent and child processes increment the value of **x** by 1.

What will be the final values of **x** in the parent and child processes after the **fork()** call?

```
#include <stdio.h>

void main() {

    int x = 5;
    fork();
    x = x+1;
    printf("x = %d\n",x);
    }
```