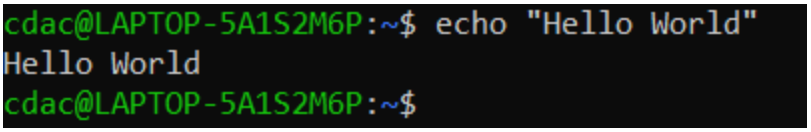
Cos Assignment 2

Part A : What will the following commands do?

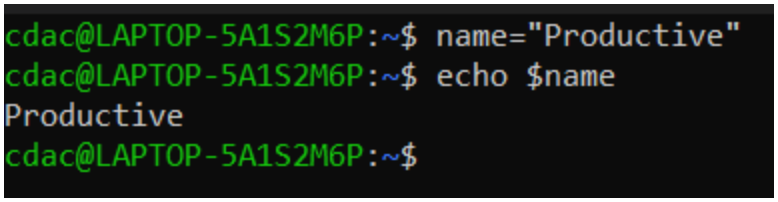
1. echo "Hello, World!”

   echo command is used to display some data or string on shell



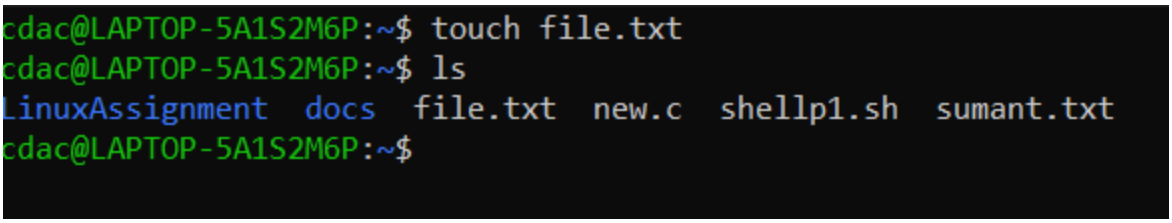
2. name="Productive"

   it will create a variable named name and store a string “Productive” in it.



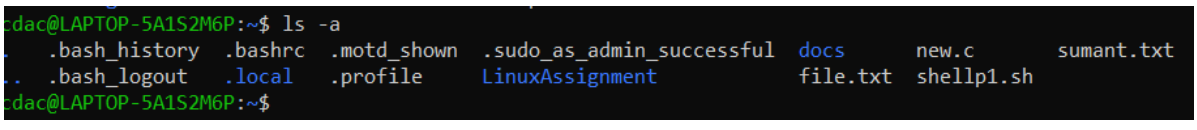
3. touch file.txt

   this command will create a file named file.txt in the present directory



4. ls -a

   This command will list all files including hidden files (files with names beginning with a dot)



5. rm file.txt

   this command would delete the file.txt file from the directory

```
cdac@LAPTOP-5A1S2M6P:~$ rm file.txt
cdac@LAPTOP-5A1S2M6P:~$ ls
LinuxAssignment  docs  new.c  shellp1.sh  sumant.txt
cdac@LAPTOP-5A1S2M6P:~$
```

6. cp file1.txt file2.txt

   The command will copy data of file1 to file2

```
cdac@LAPTOP-5A1S2M6P:~$ cat file1.txt
Sumant Reddy
cdac@LAPTOP-5A1S2M6P:~$ cat file2.txt
xyz
cdac@LAPTOP-5A1S2M6P:~$ cp file1.txt file2.txt
cdac@LAPTOP-5A1S2M6P:~$ cat file1.txt
Sumant Reddy
cdac@LAPTOP-5A1S2M6P:~$ cat file2.txt
Sumant Reddy
cdac@LAPTOP-5A1S2M6P:~$
```

7. mv file.txt /path/to/directory/

   this command will move file to given directory

```
cdac@LAPTOP-5A1S2M6P:~$ mv file1.txt /home/cdac/LinuxAssignment/
cdac@LAPTOP-5A1S2M6P:~$ cd LinuxAssignment
cdac@LAPTOP-5A1S2M6P:~/LinuxAssignment$ ls
docs  docs.zip  duplicate.txt  file1.txt  fruits.txt  numbers.txt  output.txt
cdac@LAPTOP-5A1S2M6P:~/LinuxAssignment$
```

8. chmod 755 script.sh *(revise)*

   chmod 755 file.txt. Will give: Read, write, and execute permissions to the owner of the file

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
total 0
-rw-r--r-- 1 cdac cdac 0 Aug 30 15:18 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ chmod 755 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
total 0
-rwxr-xr-x 1 cdac cdac 0 Aug 30 15:18 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

9. grep "pattern" file.txt

   grep command will search for the word “Pattern” in file.txt

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ cat file1.txt
Sumant Reddy
pattern
Cdac
Kharghar
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ grep "pattern" file1.txt
pattern
```

10. kill PID *(revise)*

    kill command is used to terminate/end process.

11. mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > file.txt && cat file.txt *(revise)*

    && : Logical AND will execute second command only if the first command is executed successfully with exit status zero

    The program will make a new directory named mydir, enters it, creates a new file named file.txt, adds echo ”Hello, World!” instruction into file.txt file and returns it’s contains

```
dac@LAPTOP-5A1S2M6P:~/Assignment2$ mkdir mydir && cd mydir && touch file.txt && echo "Hello, World!" > fil
.txt && cat file.txt
Hello, World!
dac@LAPTOP-5A1S2M6P:~/Assignment2/mydir$ ls
file.txt
dac@LAPTOP-5A1S2M6P:~/Assignment2/mydir$ pwd
/home/cdac/Assignment2/mydir
dac@LAPTOP-5A1S2M6P:~/Assignment2/mydir$
```

12. ls -l | grep ".txt" *(revise)*

'|' pipe operator command will give output of first program as input to other.

so, in this case, the command will return all the files that are with ".txt" extention

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l | grep ".txt"
-rw-r--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

13. cat file1.txt file2.txt | sort | uniq

    This command will concatenate file1 and file 2 data and sort them and return unique lines out of them

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ cat file1.txt file2.txt | sort | uniq
Aditya Sir
Cdac
Kharghar
Malkeet Sir
Sumant Reddy
Vipul Sir
pattern
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

14. ls -l | grep "^d" *(revise)*

    this will return sub directories in present directory

    '^' command returns all the lines that starts with d

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l | grep "^d"
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
total 12
-rw-r--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
-rw-r--r-- 1 cdac cdac   46 Aug 30 15:37 file2.txt
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
-rwxr-xr-x 1 cdac cdac    0 Aug 30 15:18 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

15. grep -r "pattern" /path/to/directory/

it would look for the matching pattern in all the files in directory and sub directories

```
dac@LAPTOP-5A1S2M6P:~/Assignment2$ cd ..
dac@LAPTOP-5A1S2M6P:~$ grep -r "Sumant" Assignment2/
ssignment2/file1.txt:Sumant Reddy
ssignment2/file2.txt:Sumant Reddy
dac@LAPTOP-5A1S2M6P:~$
```

16. cat file1.txt file2.txt | sort | uniq –d

    it would output the lines that were repeated in file1 and file 2 combined

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ cat file1.txt file2.txt | sort | uniq -d
Sumant Reddy
```

17. chmod 644 file.txt *(revise)*

    This command would set file permission as read write for owner and read for other users

```
dac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
otal 12
rw-r--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
rwxr--r-- 1 cdac cdac   54 Aug 30 15:47 file2.txt
rwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
rwxr-xr-x 1 cdac cdac    0 Aug 30 15:18 new.sh
dac@LAPTOP-5A1S2M6P:~/Assignment2$ chmod 644 file2.txt
dac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
otal 12
rw-r--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
rw-r--r-- 1 cdac cdac   54 Aug 30 15:47 file2.txt
rwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
rwxr-xr-x 1 cdac cdac    0 Aug 30 15:18 new.sh
dac@LAPTOP-5A1S2M6P:~/Assignment2$
```

18. cp -r source_directory destination_directory

    will copy source directory into destination directory

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ cp -r mydir/ copy/
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ cd copy/
cdac@LAPTOP-5A1S2M6P:~/Assignment2/copy$ ls
mydir
cdac@LAPTOP-5A1S2M6P:~/Assignment2/copy$ cd mydir && ls
file.txt
cdac@LAPTOP-5A1S2M6P:~/Assignment2/copy/mydir$
```

19. find /path/to/search -name "*.txt"

    the command would search for .txt file type in provided path and it's sub directories

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ find copy/ -name "*.txt"
copy/mydir/file.txt
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

20. chmod u+x file.txt

    This command will set execute permission to file.txt

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
total 16
drwxr-xr-x 3 cdac cdac 4096 Aug 30 16:02 copy
-rw-r--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
-rw-r--r-- 1 cdac cdac   54 Aug 30 15:47 file2.txt
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
-rwxr-xr-x 1 cdac cdac    0 Aug 30 15:18 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ chmod u+x file1.txt
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ ls -l
total 16
drwxr-xr-x 3 cdac cdac 4096 Aug 30 16:02 copy
-rwxr--r-- 1 cdac cdac   35 Aug 30 15:20 file1.txt
-rw-r--r-- 1 cdac cdac   54 Aug 30 15:47 file2.txt
drwxr-xr-x 2 cdac cdac 4096 Aug 30 15:29 mydir
-rwxr-xr-x 1 cdac cdac    0 Aug 30 15:18 new.sh
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

21. echo $PATH *(revise)*

    the command would run path environmental variable

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2$ echo $PATH
/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin:/sbin:/bin:/usr/games:/usr/local/games:/usr/lib/wsl/lib:/
mnt/c/Program Files/Common Files/Oracle/Java/javapath:/mnt/c/ProgramData/Oracle/Java/javapath:/mnt/c/Window
s/system32:/mnt/c/Windows:/mnt/c/Windows/System32/Wbem:/mnt/c/Windows/System32/WindowsPowerShell/v1.0/:/mnt
/c/Windows/System32/OpenSSH/:/mnt/c/Program Files (x86)/NVIDIA Corporation/PhysX/Common:/mnt/c/Program File
s/NVIDIA Corporation/NVIDIA NvDLISR:/mnt/c/Program Files/Microsoft VS Code/bin:/mnt/c/Program Files/Git/cmd
:/mnt/c/Program Files/nodejs/:/mnt/c/ProgramData/chocolatey/bin:/mnt/c/WINDOWS/system32:/mnt/c/WINDOWS:/mnt
/c/WINDOWS/System32/Wbem:/mnt/c/WINDOWS/System32/WindowsPowerShell/v1.0/:/mnt/c/WINDOWS/System32/OpenSSH/:/
mnt/c/Program Files/dotnet/:/mnt/c/Program Files/MySQL/MySQL Shell 8.0/bin/:/mnt/c/Users/reddy/AppData/Loca
l/Programs/Python/Python37/Scripts/:/mnt/c/Users/reddy/AppData/Local/Programs/Python/Python37/:/mnt/c/Users
/reddy/AppData/Local/Microsoft/WindowsApps:/mnt/c/Users/reddy/AppData/Roaming/npm:/mnt/c/MinGW/bin:/snap/bi
n
cdac@LAPTOP-5A1S2M6P:~/Assignment2$
```

Part B

Identify True or False:

1. ls is used to list files and directories in a directory : True
2. mv is used to move files and directories: True
3. cd is used to copy files and directories : False (CP Command is used to copy)
4. pwd stands for "print working directory" and displays the current directory: True
5. grep is used to search for patterns in files: True
6. chmod 755 file.txt gives read, write, and execute permissions to the owner, and read and execute permissions to group and others. - True *(revise)*
7. mkdir -p directory1/directory2 creates nested directories, creating directory2 inside directory1
   if directory1 does not exist. - True
8. rm -rf file.txt deletes a file forcefully without confirmation: False. It will recursively and forcefully remove the file

Identify the Incorrect Commands:

1. chmodx is used to change file permissions.

   incorrect. chmod command is used to change permissions.

2. cpy is used to copy files and directories.

Incorrect. cp command is used to copy files.

3. mkfile is used to create a new file.

   incorrect. touch command is used to create files.

4. catx is used to concatenate files.

   incorrect. cat command is used to concatenate.

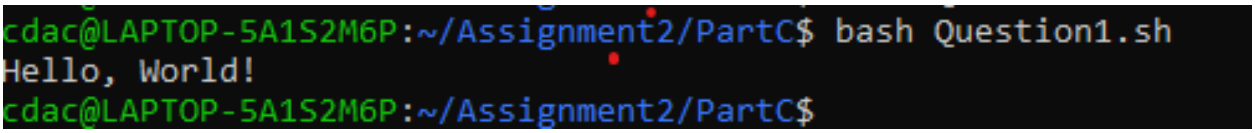
5. rn is used to rename files.

   incorrect. mv command is used to rename files.

Part C *(revise)*

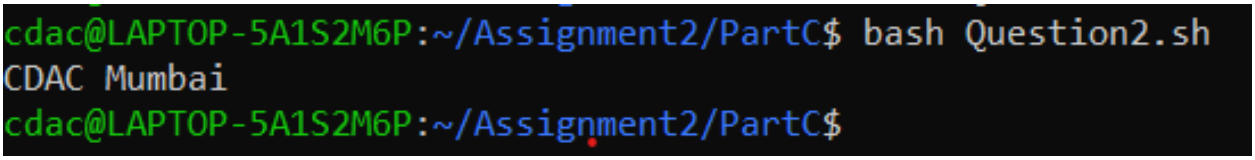
Question 1: Write a shell script that prints "Hello, World!" to the terminal.

```
echo Hello, World!
```



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

```
name="CDAC Mumbai";
echo $name;
```



Question 3: Write a shell script that takes a number as input from the user and prints it.

```
echo "Enter the Number: " ;
read num;
echo $num;
```

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question3.sh
Enter the Number:
20
20
```

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

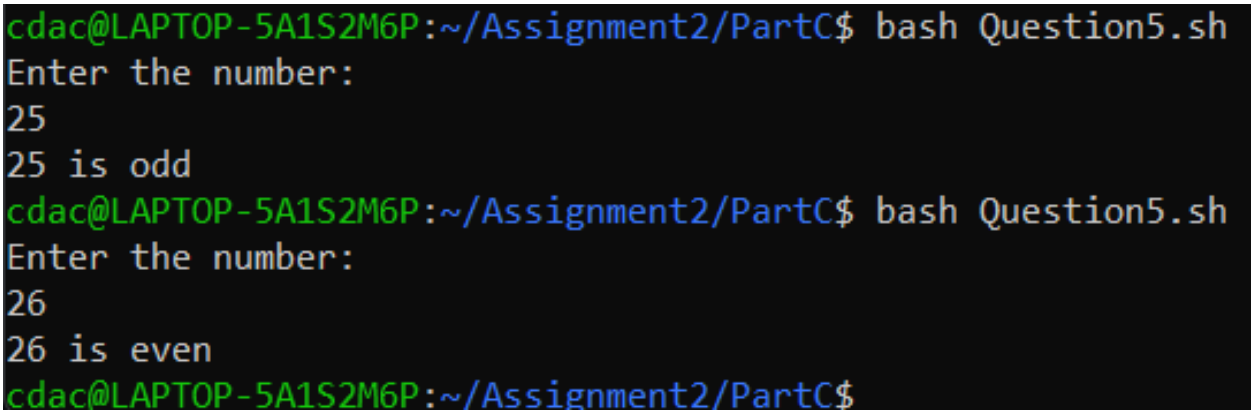
```
echo "Enter the Num1: " ;
read num1;
echo "Enter the Num2: " ;
read num2;
echo $((num1+num2));
#Can be also done:
#sum=`expr $num1 + $num2`
```

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question4.sh
Enter the Num1:
10
Enter the Num2:
20
30
```

Question 5: Write a shell script that takes a number as input and prints "Even" if it

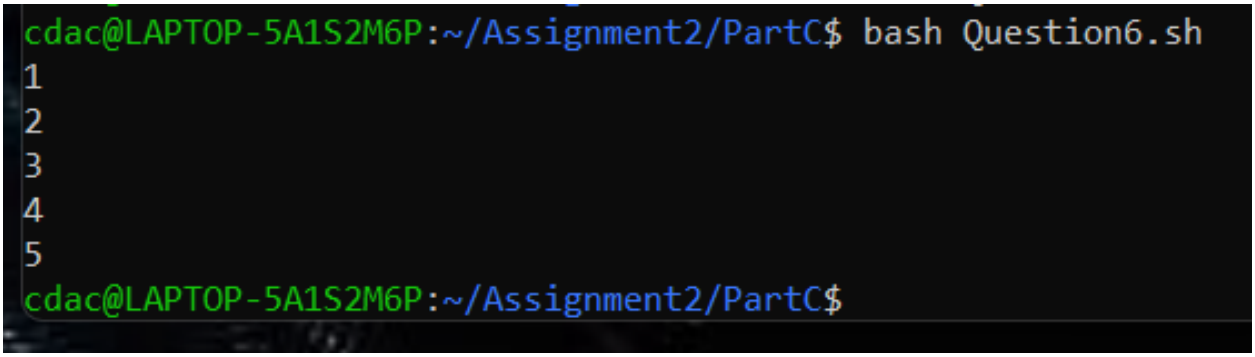
is even, otherwise
prints "Odd".

```
#!/bin/bash
echo "Enter the number: ";
read num;
if [ `expr $num % 2` != 0 ]
then
        echo $num is odd
else
        echo $num is even
fi
```



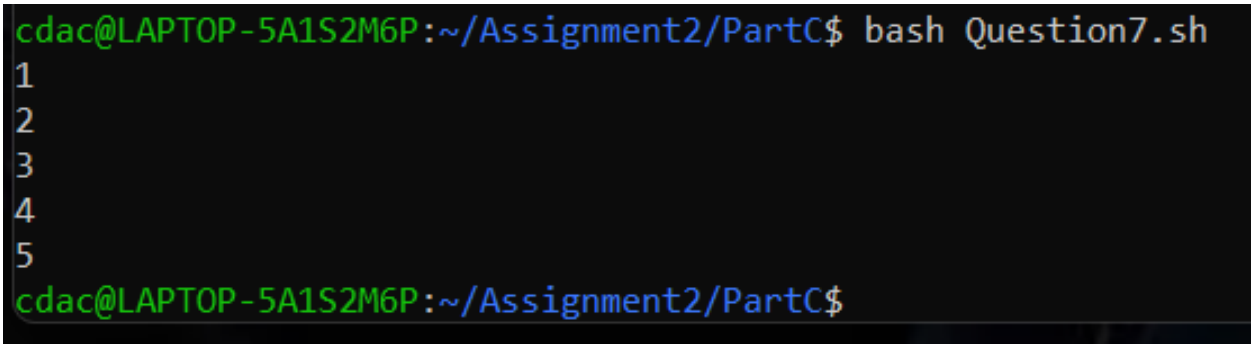
Question 6: Write a shell script that uses a for loop to print numbers from 1 to 5.

```
for num in 1 2 3 4 5
do
        echo $num
done
```



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

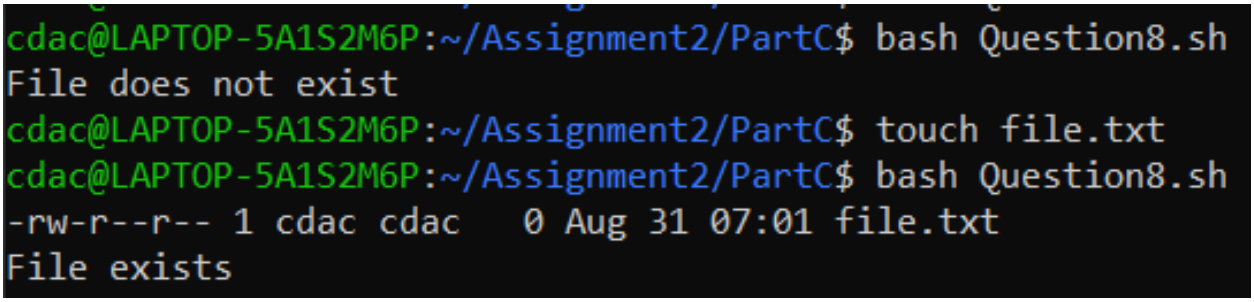
```
#!/bin/bash
num=1
while [ $num -le 5 ]
do
        echo $num
        num=`expr $num + 1`
done
```



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".

```
#!/bin/bash
ls -l | grep 'file.txt'
```

```
if [ $? -eq 0 ]
then
        echo File exists
else
        echo File does not exist
fi
```



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.

```
#!/bin/bash
echo enter the number
read num

if [ $num -gt 10 ]
then
        echo number is greater than 10
else
        echo number is less than 10
fi
```

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question9.sh
enter the number
15
number is greater than 10
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question9.sh
enter the number
9
number is less than 10
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$
```

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.

```
#!/bin/bash

for var in 1 2 3 4 5
do
for num in 1 2 3 4 5 6 7 8 9 10
do
        echo -n $((var * num))" "
done
        echo
done
```

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question10.sh
1 2 3 4 5 6 7 8 9 10
2 4 6 8 10 12 14 16 18 20
3 6 9 12 15 18 21 24 27 30
4 8 12 16 20 24 28 32 36 40
5 10 15 20 25 30 35 40 45 50
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$
```

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.

```
while [ 0 ]
do
echo enter the number
read num
        if [ $num -gt 0 ]
        then
                echo square of number is $((num * num))
        else
                echo invalid number
                break
        fi
done
```

```
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$ bash Question11.sh
enter the number
25
square of number is 625
enter the number
10
square of number is 100
enter the number
-5
invalid number
cdac@LAPTOP-5A1S2M6P:~/Assignment2/PartC$
```

Part E (*revise*)

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

First Come First Serve Scheduling

Process	Arrival Time	Burst Time	Waiting Time
P1	0	5	0
P2	1	3	4
P3	2	6	6

	P1	P2	P3	
0	5	8	14	

Avg Waiting Time = (0+4+6)/3 = 10/3 = 3.3

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

Shortest Job First Scheduling

Process	Arrival Time	Burst Time	Waiting Time	TAT
p1	0	3	0	3
p2	1	5	7	12
p3	2	1	1	2
p4	3	4	1	5

	p1	p3	p4	p2
0	3	4	8	13

Avg Waiting Time = (0+7+1+1)/4 = 2.25

Avg Turn Around Time = (3+12+2+5)/4 = 22/4 = 5.5

3. Calculate the average waiting time using Priority Scheduling

Priority Scheduling

Process	Arrival Tim	Burst Time	Priority	Waiting Time
p1	0	6	3	6
p2	1	4	1	0
p3	2	7	4	10
p4	3	2	2	2

	p1	p2	p4	p1	p3
0	1	5	7	12	19

Avg Waiting Time = (6+0+10+2)/4 = 18/4 = 4.5

4. Calculate the average turnaround time using Round Robin scheduling.

Round Robin Scheduling: Quantum = 2

Process	Arrival Time	Burst Time	Waiting Time	TAT
p1	0	4	6	10
p2	1	5	9	13
p3	2	2	4	4
p4	3	3	10	10

	p1	p2	p3	p4	p1	p2	p4	p2
0	2	4	6	8	10	12	13	14

Avg waiting time= (6+9+4+10)/4 = 29/4 = 7.25

Avg TAT = (10+13+4+10)/4 = 37/4 = 9.25

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?
→ Final value of x would be same for both parent and child processes.

```c
#include <stdio.h>

int main() {
    int x = 5;
    fork();
    x += 1;
    printf("Value of x is: %d\n", x);
    return 0;
}
```

```
/tmp/kCQoSwEBf3.c: In function 'main':
/tmp/kCQoSwEBf3.c:7:5: warning: implicit declaration of function 'fork' [-Wimplicit
   -function-declaration]
   7 |     fork();
     |     ^~~~
/tmp/kCQoSwEBf3.o
Value of x is: 6
Value of x is: 6


=== Code Execution Successful ===
```