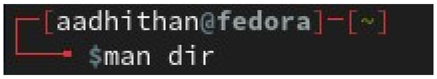
EXPERIMENT 1

Aim: Study of Unix general purpose utility command list.

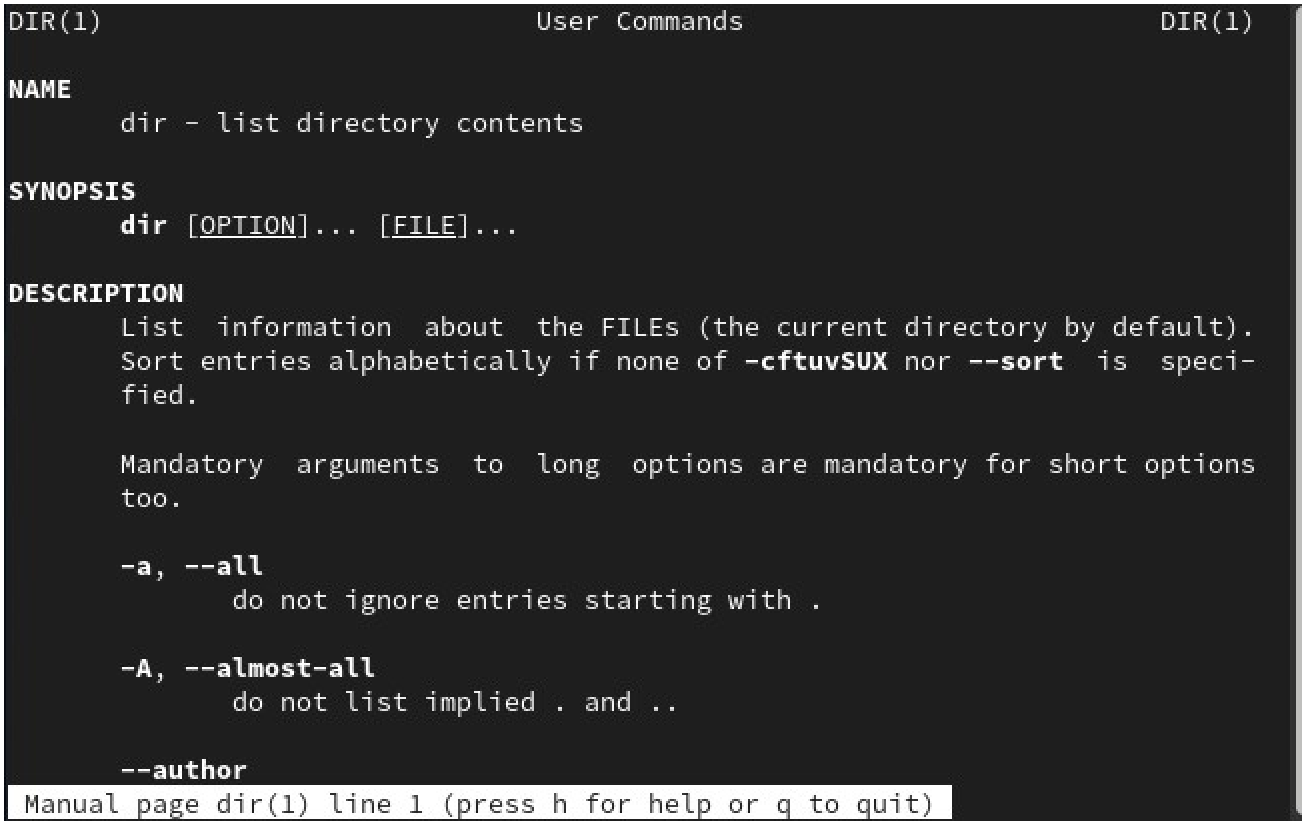
man command: Displays help of the following command keyword.

i/p:



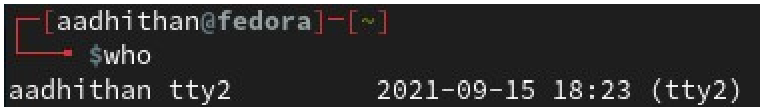
```
┌─[aadhithan@fedora]─[~]
└──╼ $man dir
```

o/p:



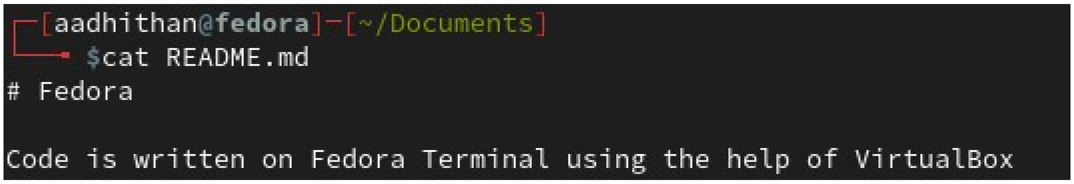
```
DIR(1)                        User Commands                        DIR(1)

NAME
       dir - list directory contents

SYNOPSIS
       dir [OPTION]... [FILE]...

DESCRIPTION
       List  information  about  the FILEs (the current directory by default).
       Sort entries alphabetically if none of -cftuvSUX nor --sort  is  speci-
       fied.

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

who command: Displays the current users of the system.



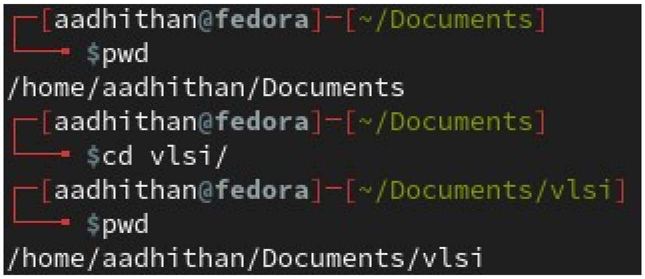
```
┌─[aadhithan@fedora]─[~]
└──╼ $who
aadhithan tty2         2021-09-15 18:23 (tty2)
```

cat command: Displays the content of the file in the shell terminal.



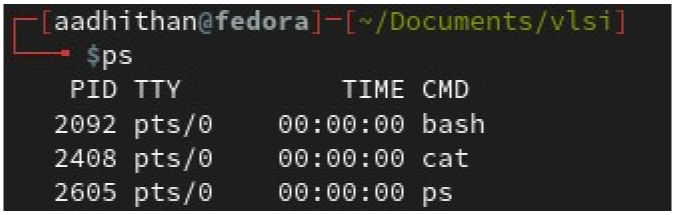
```
┌─[aadhithan@fedora]─[~/Documents]
└──╼ $cat README.md
# Fedora

Code is written on Fedora Terminal using the help of VirtualBox
```

cd command: Navigates across directories:



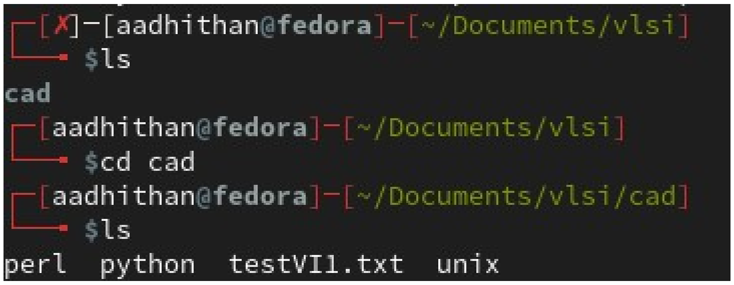
```
┌─[aadhithan@fedora]─[~/Documents]
└──╼ $pwd
/home/aadhithan/Documents
┌─[aadhithan@fedora]─[~/Documents]
└──╼ $cd vlsi/
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $pwd
/home/aadhithan/Documents/vlsi
```

ps command: Lists the current working processes



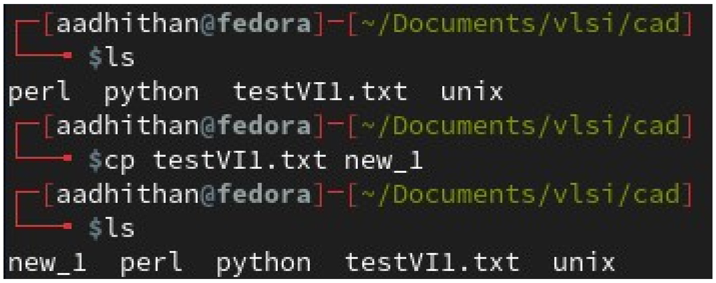
```
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ps
    PID TTY          TIME CMD
   2092 pts/0    00:00:00 bash
   2408 pts/0    00:00:00 cat
   2605 pts/0    00:00:00 ps
```

ls command: lists the content of the directory



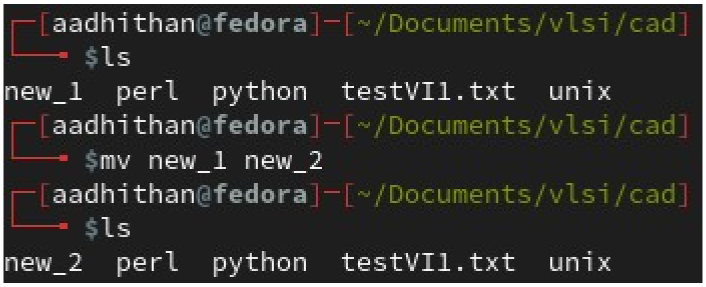
```
┌─[✗]─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ls
cad
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $cd cad
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
perl  python  testVI1.txt  unix
```

cp command: copies the file



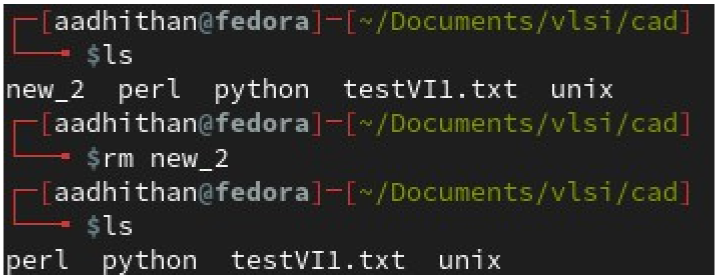
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
perl  python  testVI1.txt  unix
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $cp testVI1.txt new_1
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
new_1  perl  python  testVI1.txt  unix
```

mv command: moves the file to new destination



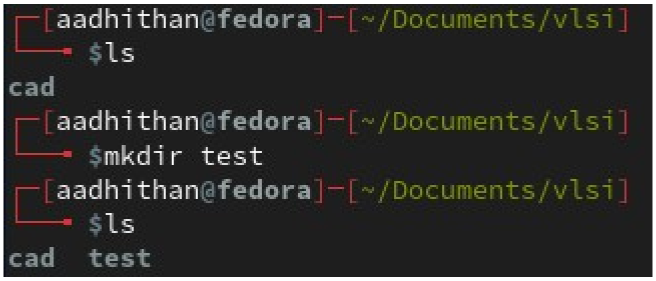
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
new_1  perl  python  testVI1.txt  unix
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $mv new_1 new_2
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
new_2  perl  python  testVI1.txt  unix
```

rm command: removes/deletes the file



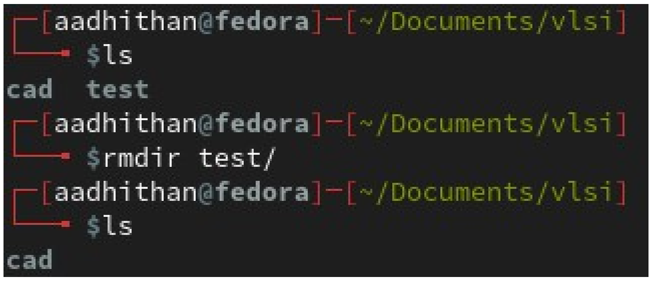
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
new_2  perl  python  testVI1.txt  unix
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $rm new_2
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
perl  python  testVI1.txt  unix
```

mkdir command: creates a new directory



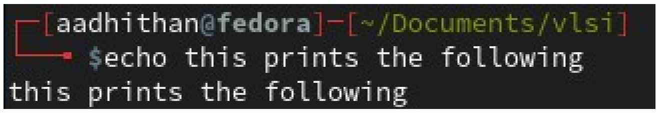
```
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ls
cad
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $mkdir test
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ls
cad  test
```

rmdir command: deletes an existing directory



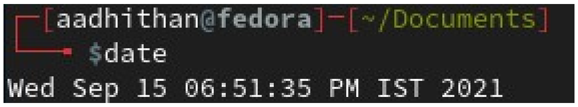
```
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ls
cad  test
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $rmdir test/
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $ls
cad
```

echo command: prints the text that follows the command



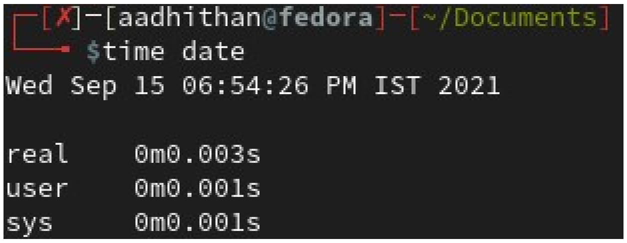
```
┌─[aadhithan@fedora]─[~/Documents/vlsi]
└──╼ $echo this prints the following
this prints the following
```

date command: prints the current date and time



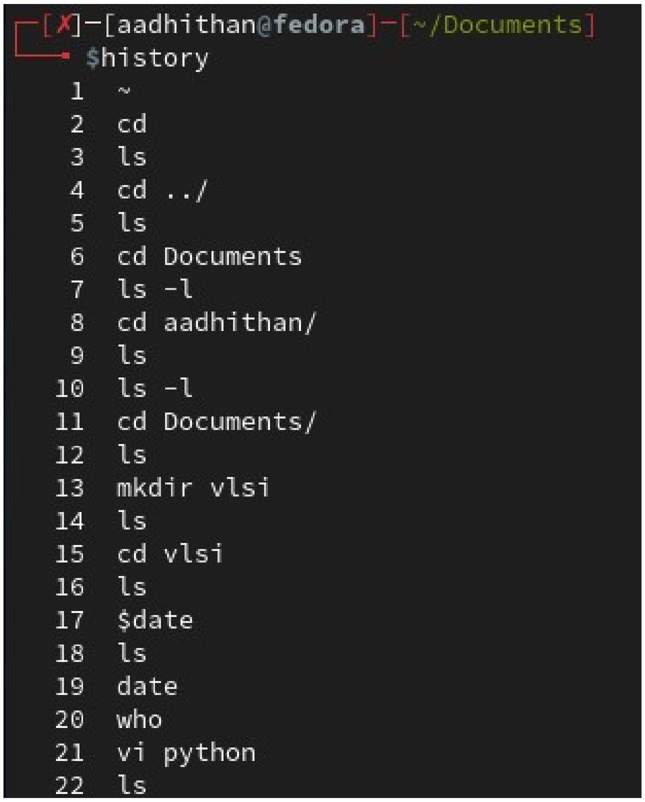
```
┌─[aadhithan@fedora]─[~/Documents]
└──╼ $date
Wed Sep 15 06:51:35 PM IST 2021
```

time command: displays the time required to execute a command



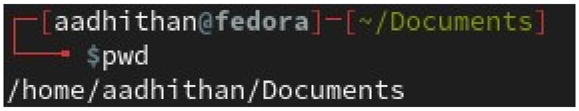
```
┌─[✗]─[aadhithan@fedora]─[~/Documents]
└──╼ $time date
Wed Sep 15 06:54:26 PM IST 2021

real    0m0.003s
user    0m0.001s
sys     0m0.001s
```

history command: Displays the history of commands



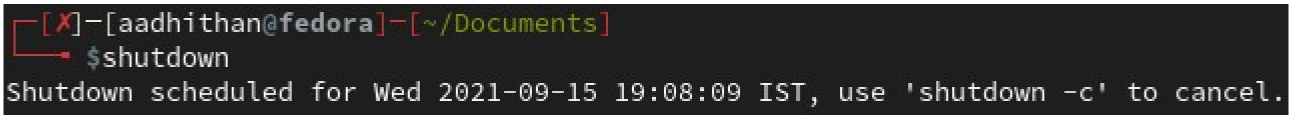
```
┌─[✗]─[aadhithan@fedora]─[~/Documents]
└──╼ $history
    1  ~
    2  cd
    3  ls
    4  cd ../
    5  ls
    6  cd Documents
    7  ls -l
    8  cd aadhithan/
    9  ls
   10  ls -l
   11  cd Documents/
   12  ls
   13  mkdir vlsi
   14  ls
   15  cd vlsi
   16  ls
   17  $date
   18  ls
   19  date
   20  who
   21  vi python
   22  ls
```

pwd command: prints current working directory



```
┌─[aadhithan@fedora]─[~/Documents]
└──╼ $pwd
/home/aadhithan/Documents
```

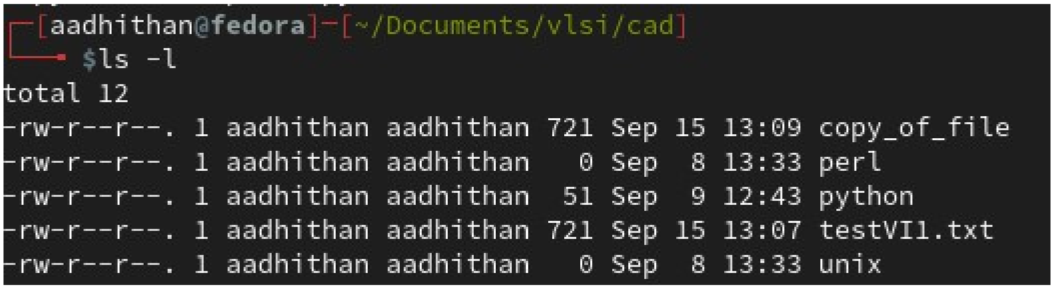
shutdown command: schedules shutdown to happen in 2 minutes



```
┌─[✗]─[aadhithan@fedora]─[~/Documents]
└──╼ $shutdown
Shutdown scheduled for Wed 2021-09-15 19:08:09 IST, use 'shutdown -c' to cancel.
```

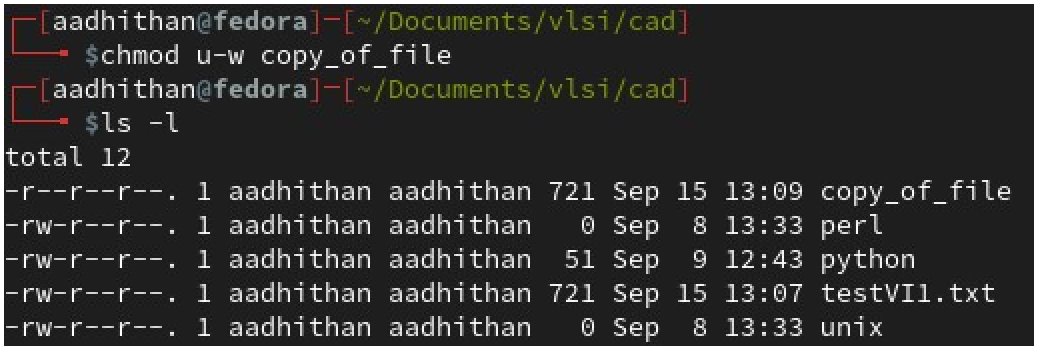
chmod command: used to change preferences of a file

Before changing user preferences:



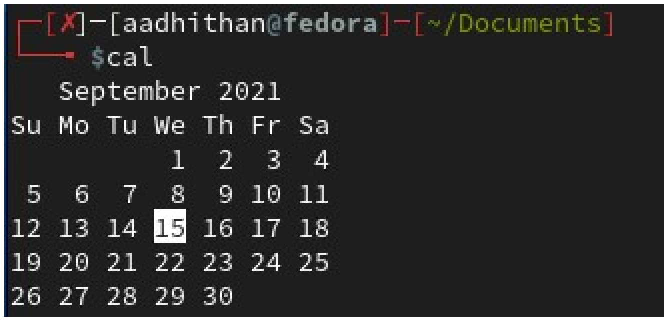
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls -l
total 12
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:09 copy_of_file
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 perl
-rw-r--r--. 1 aadhithan aadhithan  51 Sep  9 12:43 python
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:07 testVI1.txt
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 unix
```

Changing preferences:



```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $chmod u-w copy_of_file
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls -l
total 12
-r--r--r--. 1 aadhithan aadhithan 721 Sep 15 13:09 copy_of_file
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 perl
-rw-r--r--. 1 aadhithan aadhithan  51 Sep  9 12:43 python
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:07 testVI1.txt
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 unix
```

cal command: Displays the calendar



```
┌─[✗]─[aadhithan@fedora]─[~/Documents]
└──╼ $cal
   September 2021
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
```

kill command: kills a program that’s running

more command: shifts the file one screen at a time

chown command: change owner of a file

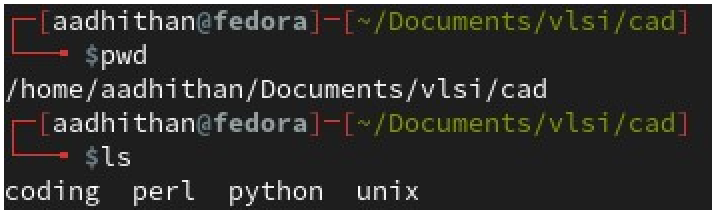
finger command: gets details about a domain

logout command: allows to programmatically logout from the session

EXPERIMENT 2: File Handling

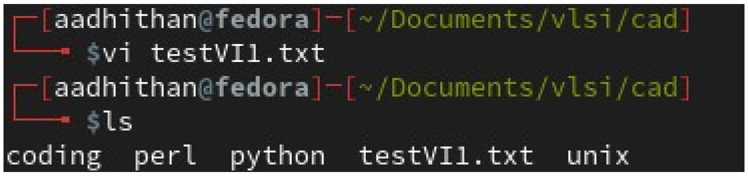
1, Create a file named testVI1.txt in the CAD folder created last time. Type minimum of 2 paragraphs of 10 lines at least.

cad folder before the creation of file:



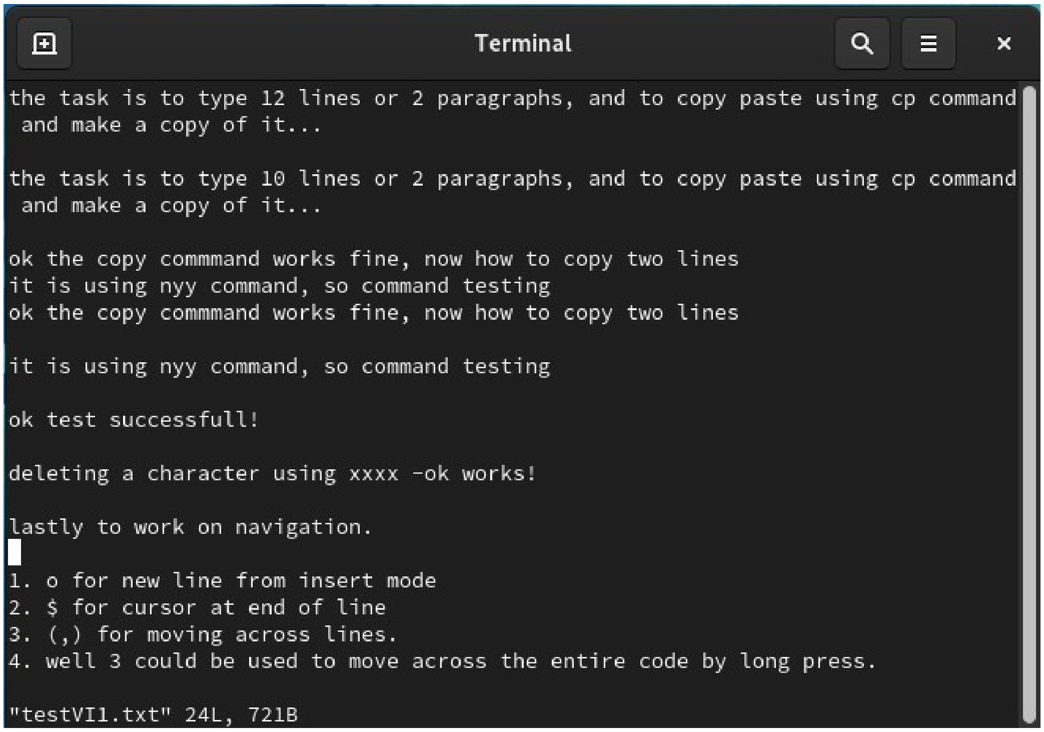
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $pwd
/home/aadhithan/Documents/vlsi/cad
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
coding  perl  python  unix
```

cad folder after the creation of file:



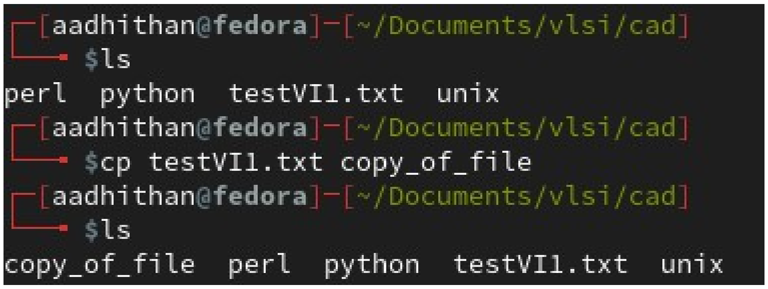
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $vi testVI1.txt
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
coding  perl  python  testVI1.txt  unix
```

The text file created:



```
the task is to type 12 lines or 2 paragraphs, and to copy paste using cp command
 and make a copy of it...

the task is to type 10 lines or 2 paragraphs, and to copy paste using cp command
 and make a copy of it...

ok the copy commmand works fine, now how to copy two lines
it is using nyy command, so command testing
ok the copy commmand works fine, now how to copy two lines

it is using nyy command, so command testing

ok test successfull!

deleting a character using xxxx -ok works!

lastly to work on navigation.
█
1. o for new line from insert mode
2. $ for cursor at end of line
3. (,) for moving across lines.
4. well 3 could be used to move across the entire code by long press.

"testVI1.txt" 24L, 721B
```

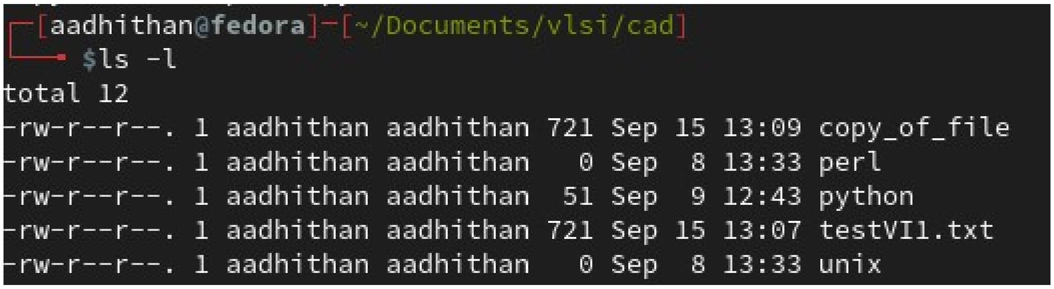
2, Use the cp command to create the copy of the file.



```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
perl  python  testVI1.txt  unix
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $cp testVI1.txt copy_of_file
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
copy_of_file  perl  python  testVI1.txt  unix
```

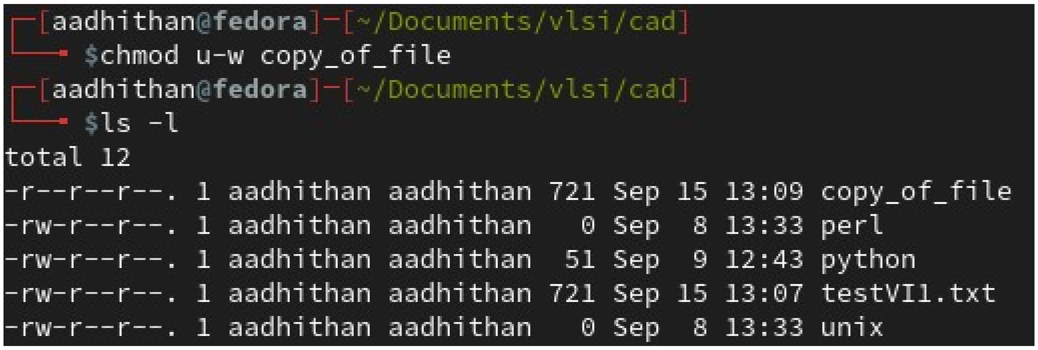
3, Try out opening the second file in read only mode and delete it later

Before changing user preferences:



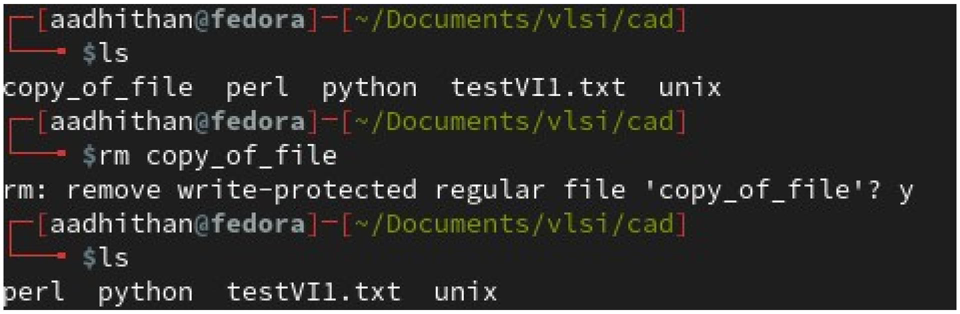
```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls -l
total 12
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:09 copy_of_file
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 perl
-rw-r--r--. 1 aadhithan aadhithan  51 Sep  9 12:43 python
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:07 testVI1.txt
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 unix
```

Changing preferences:



```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $chmod u-w copy_of_file
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls -l
total 12
-r--r--r--. 1 aadhithan aadhithan 721 Sep 15 13:09 copy_of_file
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 perl
-rw-r--r--. 1 aadhithan aadhithan  51 Sep  9 12:43 python
-rw-r--r--. 1 aadhithan aadhithan 721 Sep 15 13:07 testVI1.txt
-rw-r--r--. 1 aadhithan aadhithan   0 Sep  8 13:33 unix
```

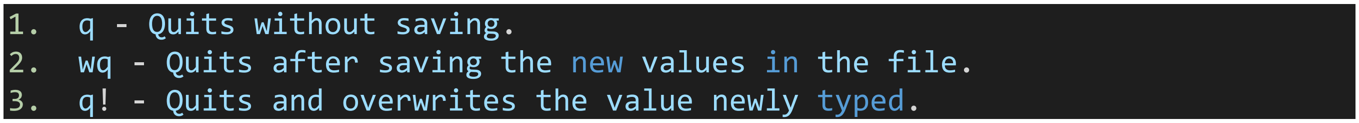
Deleting the file:



```
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
copy_of_file  perl  python  testVI1.txt  unix
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $rm copy_of_file
rm: remove write-protected regular file 'copy_of_file'? y
┌─[aadhithan@fedora]─[~/Documents/vlsi/cad]
└──╼ $ls
perl  python  testVI1.txt  unix
```

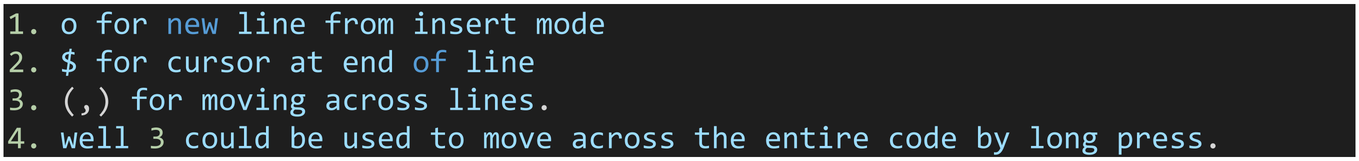
4, Inside the file

Ways to come out of the file:



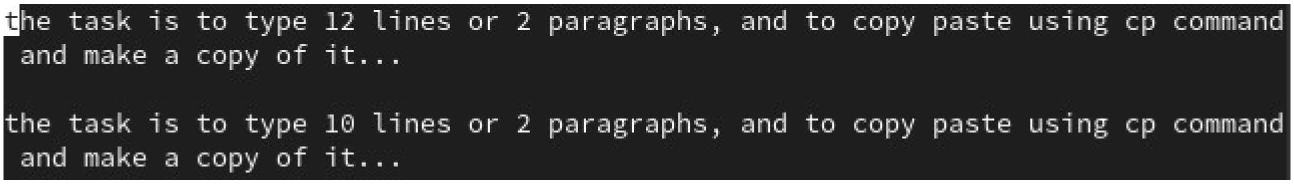
```
1.  q - Quits without saving.
2.  wq - Quits after saving the new values in the file.
3.  q! - Quits and overwrites the value newly typed.
```

Used various key for navigation:



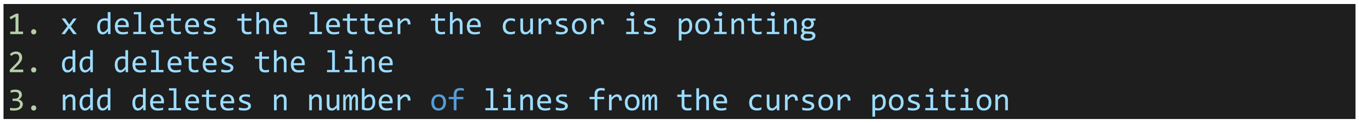
```
1. o for new line from insert mode
2. $ for cursor at end of line
3. (,) for moving across lines.
4. well 3 could be used to move across the entire code by long press.
```

Copy paste using yy and p command:



```
the task is to type 12 lines or 2 paragraphs, and to copy paste using cp command
 and make a copy of it...

the task is to type 10 lines or 2 paragraphs, and to copy paste using cp command
 and make a copy of it...
```

Deleting in the file:



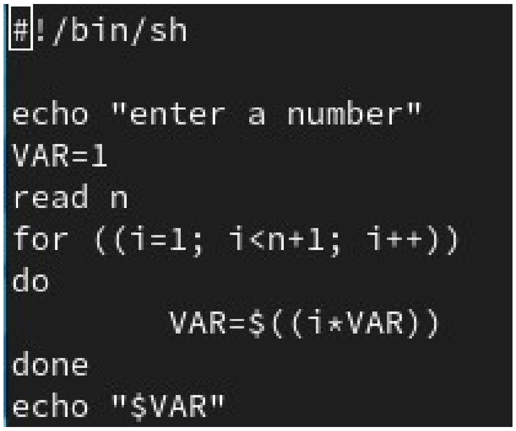
```
1. x deletes the letter the cursor is pointing
2. dd deletes the line
3. ndd deletes n number of lines from the cursor position
```

EXPERIMENT 3: Shell Scripting

a) Write a Shell Script program to find factorial of a number.

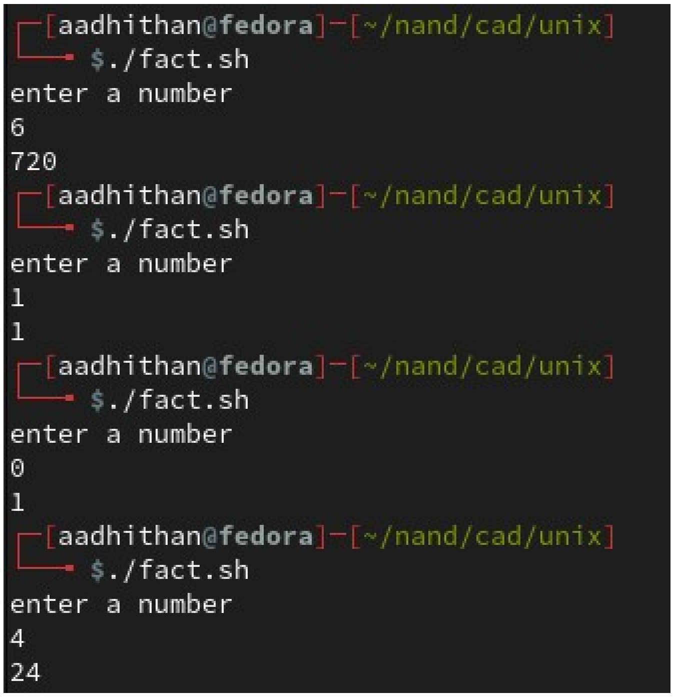
Flowchart:

Code:



```
#!/bin/sh

echo "enter a number"
VAR=1
read n
for ((i=1; i<n+1; i++))
do
        VAR=$((i*VAR))
done
echo "$VAR"
```

Output:

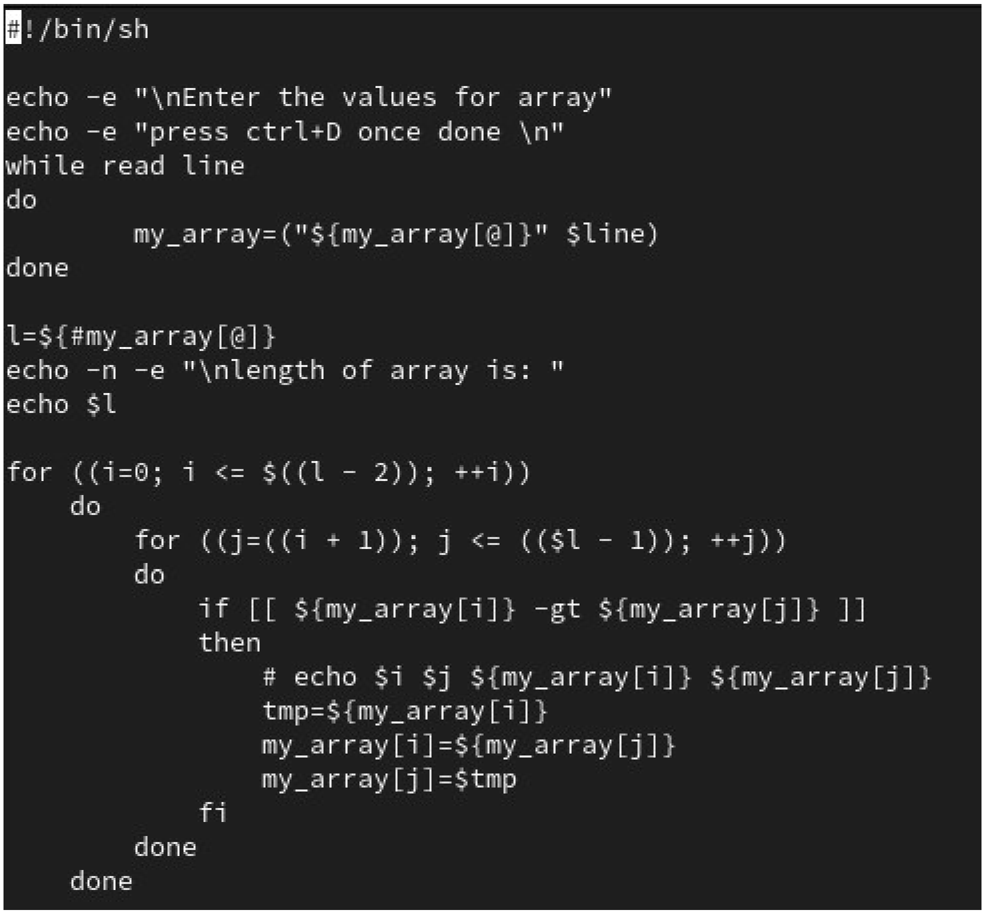


```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./fact.sh
enter a number
6
720
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./fact.sh
enter a number
1
1
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./fact.sh
enter a number
0
1
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./fact.sh
enter a number
4
24
```

b) Write a Shell Script program to sort an array in ascending order.

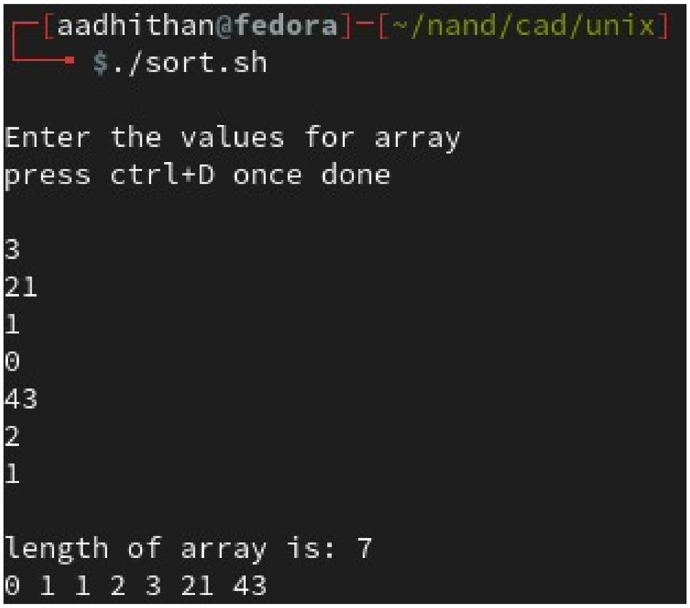
Flowchart:

Code:



```
#!/bin/sh

echo -e "\nEnter the values for array"
echo -e "press ctrl+D once done \n"
while read line
do
        my_array=("${my_array[@]}" $line)
done

l=${#my_array[@]}
echo -n -e "\nlength of array is: "
echo $l

for ((i=0; i <= $((l - 2)); ++i))
    do
        for ((j=((i + 1)); j <= (($l - 1)); ++j))
        do
            if [[ ${my_array[i]} -gt ${my_array[j]} ]]
            then
                # echo $i $j ${my_array[i]} ${my_array[j]}
                tmp=${my_array[i]}
                my_array[i]=${my_array[j]}
                my_array[j]=$tmp
            fi
        done
    done
```

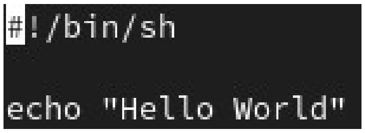
Output:



```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./sort.sh

Enter the values for array
press ctrl+D once done

3
21
1
0
43
2
1

length of array is: 7
0 1 1 2 3 21 43
```

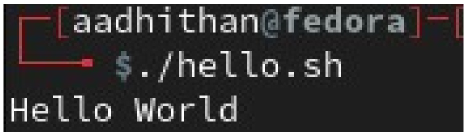
c) **Write a Shell Script program to display “Hello World”.**

Code:



```
#!/bin/sh

echo "Hello World"
```

Output:

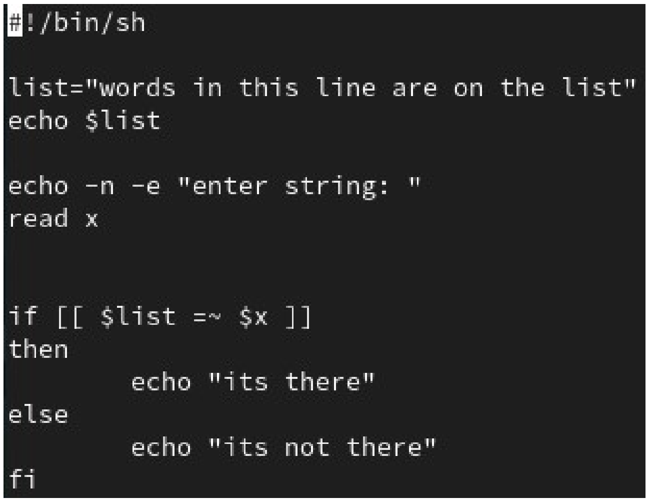


```
┌─[aadhithan@fedora]─[
└──╼ $./hello.sh
Hello World
```

d) **Write a Shell Script program to search whether element is present in the list or not.**

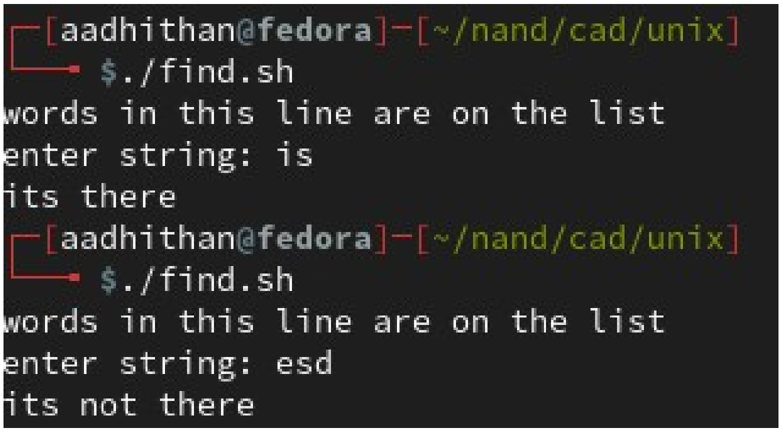
Flowchart:

Code:



```
#!/bin/sh

list="words in this line are on the list"
echo $list

echo -n -e "enter string: "
read x


if [[ $list =~ $x ]]
then
        echo "its there"
else
        echo "its not there"
fi
```

Output:



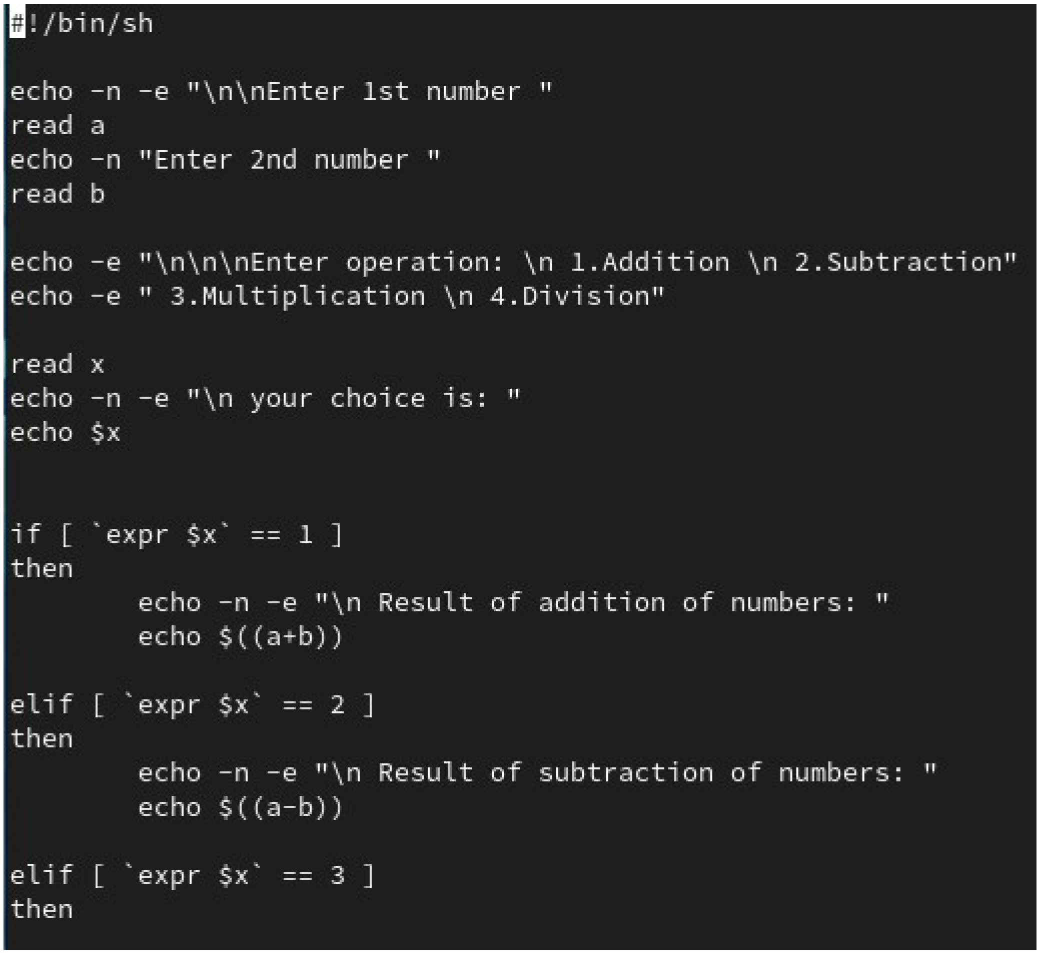
```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./find.sh
words in this line are on the list
enter string: is
its there
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./find.sh
words in this line are on the list
enter string: esd
its not there
```

EXPERIMENT 4: VI Editor

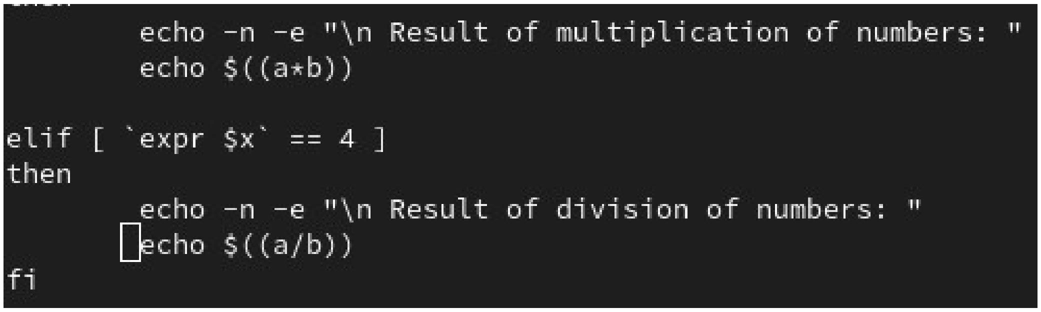
a) Write a Shell Script program to develop a calculator.

Flowchart:

Code:

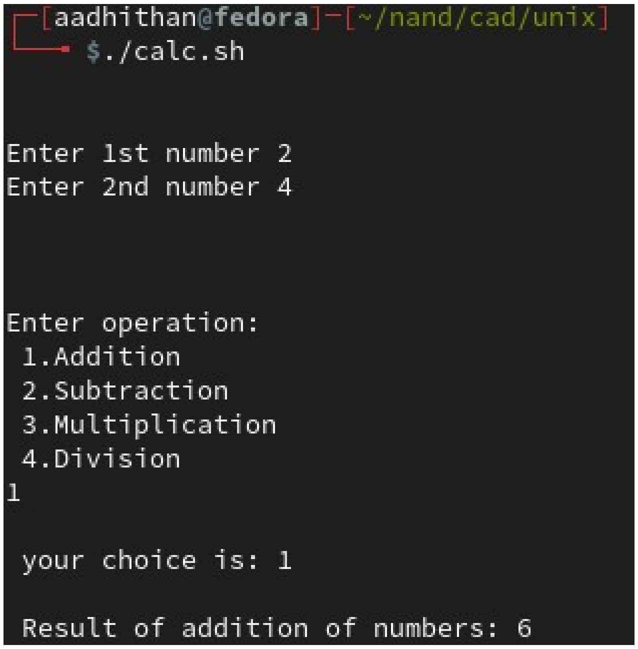


```
#!/bin/sh

echo -n -e "\n\nEnter 1st number "
read a
echo -n "Enter 2nd number "
read b

echo -e "\n\n\nEnter operation: \n 1.Addition \n 2.Subtraction"
echo -e " 3.Multiplication \n 4.Division"

read x
echo -n -e "\n your choice is: "
echo $x


if [ `expr $x` == 1 ]
then
        echo -n -e "\n Result of addition of numbers: "
        echo $((a+b))

elif [ `expr $x` == 2 ]
then
        echo -n -e "\n Result of subtraction of numbers: "
        echo $((a-b))

elif [ `expr $x` == 3 ]
then
```

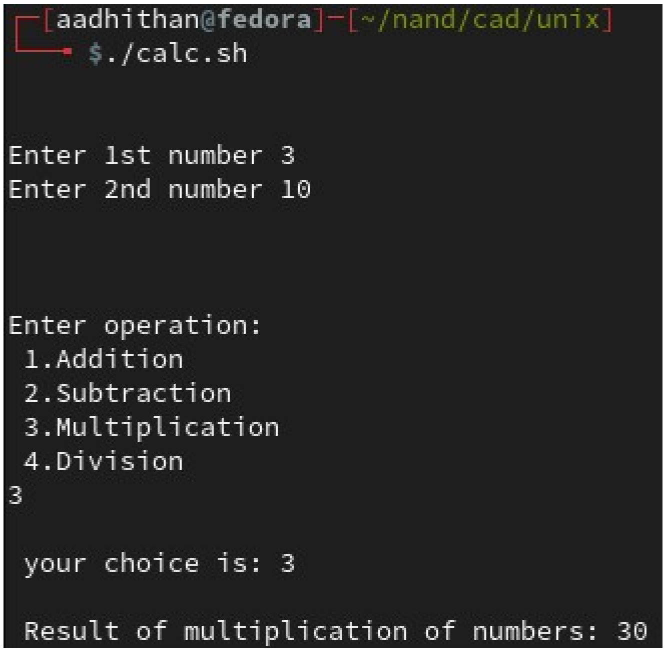


```
then
        echo -n -e "\n Result of multiplication of numbers: "
        echo $((a*b))

elif [ `expr $x` == 4 ]
then
        echo -n -e "\n Result of division of numbers: "
        echo $((a/b))
fi
```

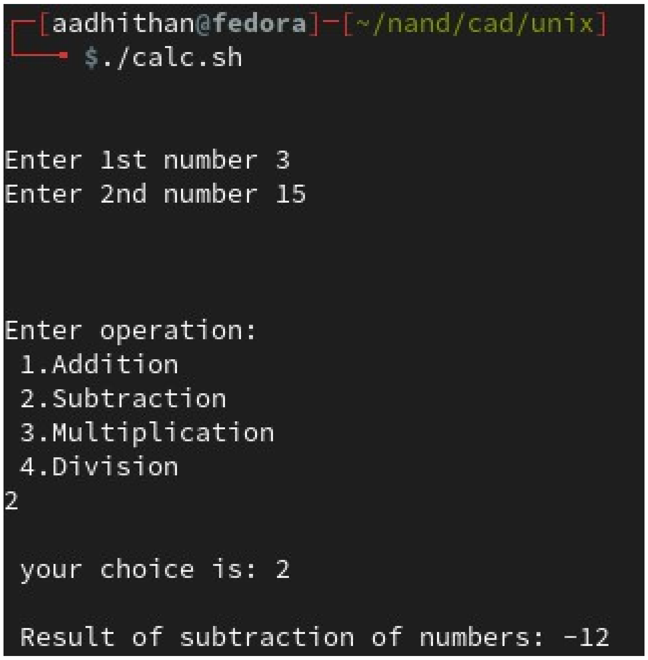
Output:



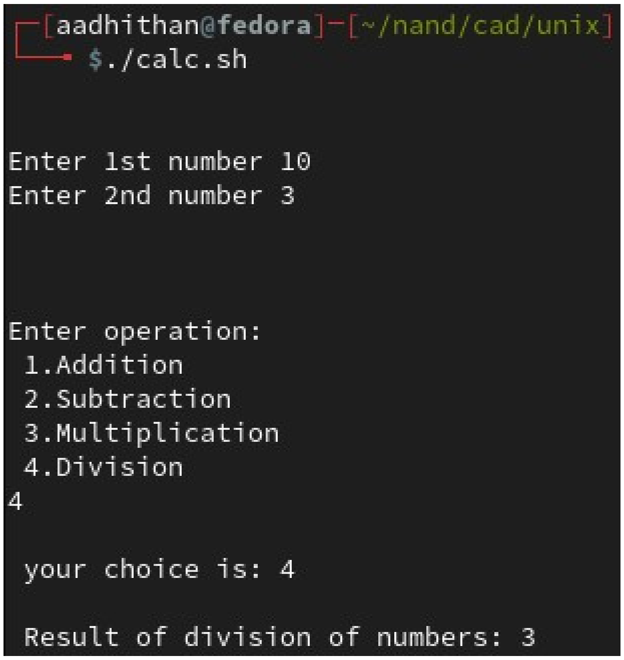
```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./calc.sh


Enter 1st number 2
Enter 2nd number 4



Enter operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division
1

 your choice is: 1

 Result of addition of numbers: 6
```



```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./calc.sh


Enter 1st number 3
Enter 2nd number 10



Enter operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division
3

 your choice is: 3

 Result of multiplication of numbers: 30
```



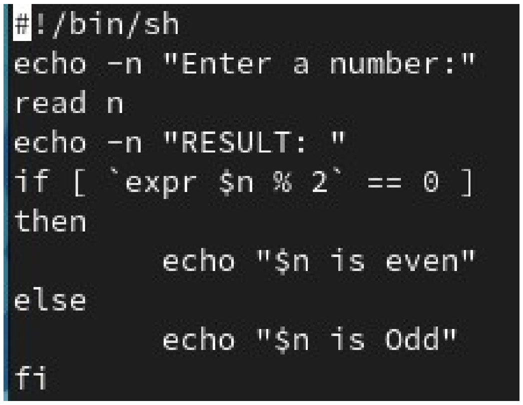
```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./calc.sh


Enter 1st number 3
Enter 2nd number 15



Enter operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division
2

 your choice is: 2

 Result of subtraction of numbers: -12
```



```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./calc.sh


Enter 1st number 10
Enter 2nd number 3



Enter operation:
 1.Addition
 2.Subtraction
 3.Multiplication
 4.Division
4

 your choice is: 4

 Result of division of numbers: 3
```

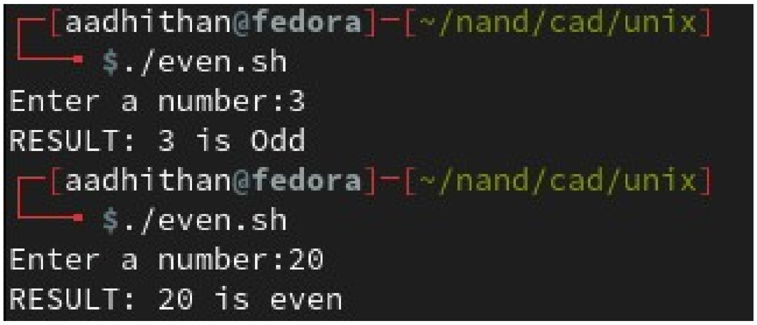
b) Write a Shell Script program to check whether the given number is even or odd.

Code:



```
#!/bin/sh
echo -n "Enter a number:"
read n
echo -n "RESULT: "
if [ `expr $n % 2` == 0 ]
then
        echo "$n is even"
else
        echo "$n is Odd"
fi
```

Output:



```
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./even.sh
Enter a number:3
RESULT: 3 is Odd
┌─[aadhithan@fedora]─[~/nand/cad/unix]
└──╼ $./even.sh
Enter a number:20
RESULT: 20 is even
```

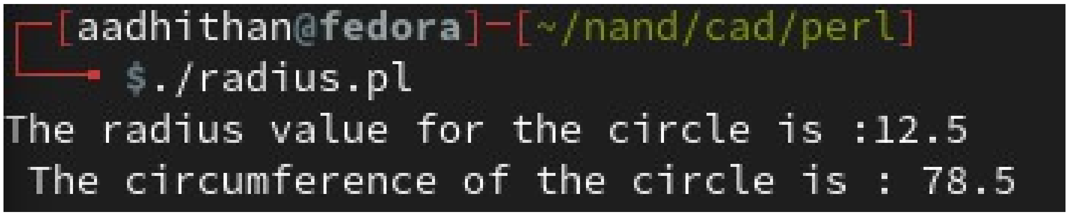
EXPERIMENT 5

Write a Perl program that computes the circumference of a circle with a radius of 12.5 units.

Code:

```perl
#!/usr/bin/perl

$radius = 12.5;
print "The radius value for the circle is :$radius \n ";
$circumference = (2 * 3.14 * $radius);
print "The circumference of the circle is : $circumference \n";
```

Output:



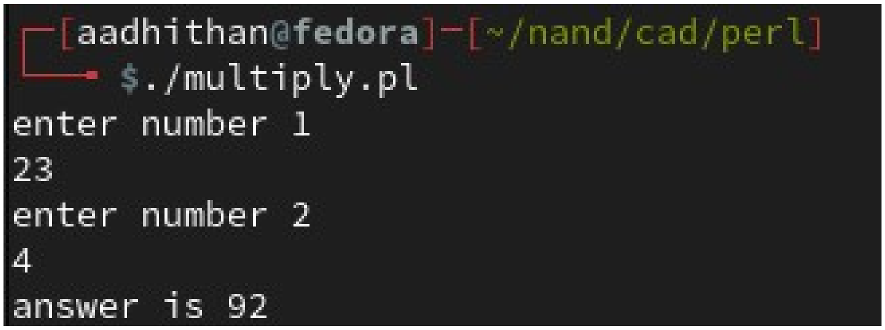
```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./radius.pl
The radius value for the circle is :12.5
 The circumference of the circle is : 78.5
```

Write a Perl program to take in two numbers and prints out the result of the two numbers multiplied.

Code:

```perl
#!/usr/bin/perl

print "enter number 1 \n";
$n1 = <STDIN>;
print "enter number 2 \n";
$n2 = <STDIN>;

$s = $n1*$n2;
print "answer is $s \n";
```

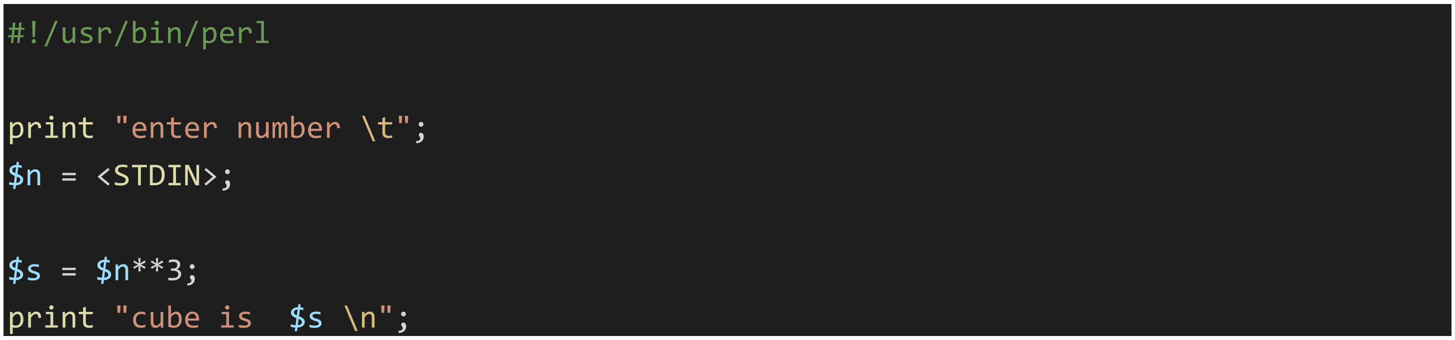
Output:



```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./multiply.pl
enter number 1
23
enter number 2
4
answer is 92
```

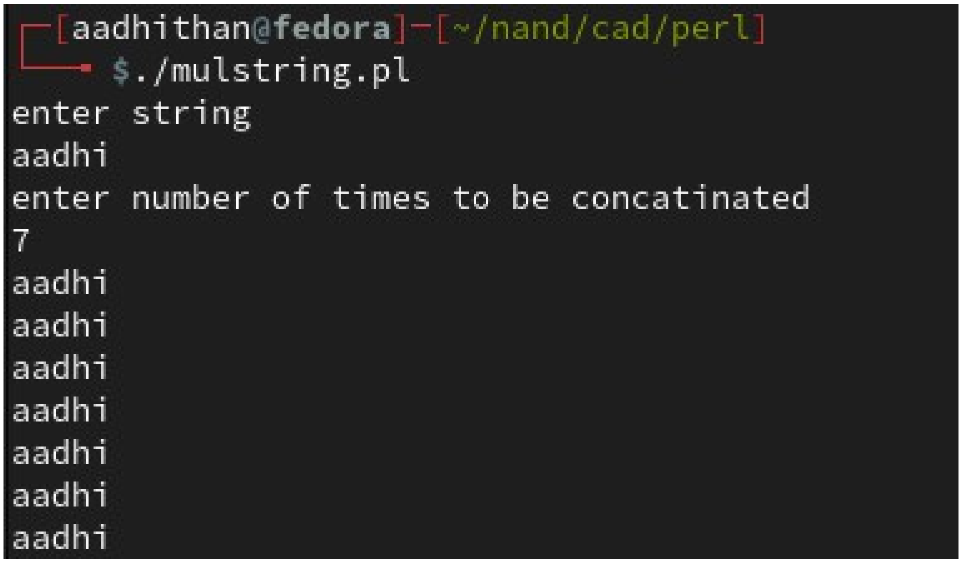
Write a Perl program that reads in a string and a number, and then prints out the string the number of times requested.

Code:



```
#!/usr/bin/perl

print "enter number \t";
$n = <STDIN>;

$s = $n**3;
print "cube is  $s \n";
```

Output:



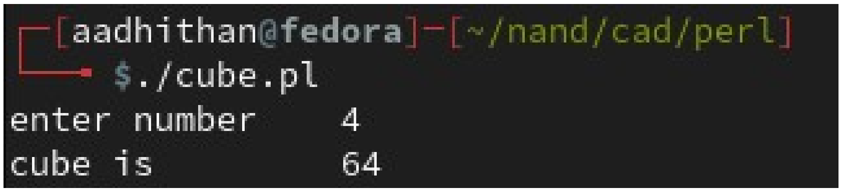
```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./mulstring.pl
enter string
aadhi
enter number of times to be concatinated
7
aadhi
aadhi
aadhi
aadhi
aadhi
aadhi
aadhi
```

Write a Perl program that prints the cube of a number.

Code:

```perl
#!/usr/bin/perl

print "enter number \t";
$n = <STDIN>;

$s = $n**3;
print "cube is  $s \n";
```

Output:



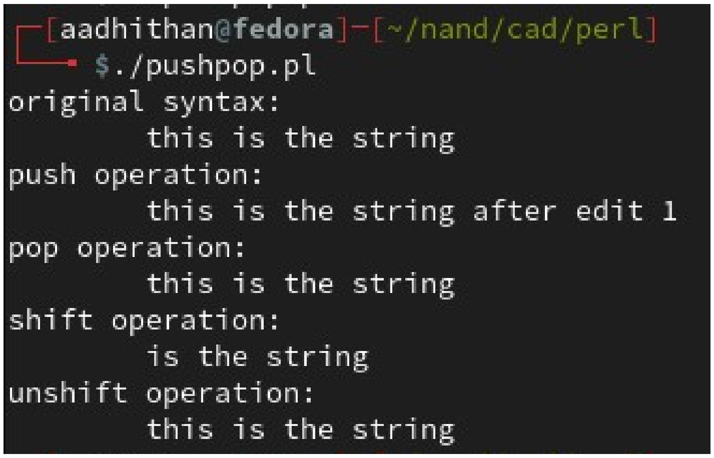
```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./cube.pl
enter number    4
cube is         64
```

Write a code to explore String operators.

Code:

```perl
!/usr/bin/perl

@s = ('this', 'is', 'the', 'string');
print "original syntax: \n \t";
print "@s \n";

push(@s,'after edit 1');
print "push operation: \n \t";
print "@s \n";

pop(@s);
print "pop operation: \n \t";
print "@s \n";

shift(@s);
print "shift operation: \n \t";
print "@s \n";

unshift(@s, 'this');
print "unshift operation: \n \t";
print "@s \n";
```

Output:



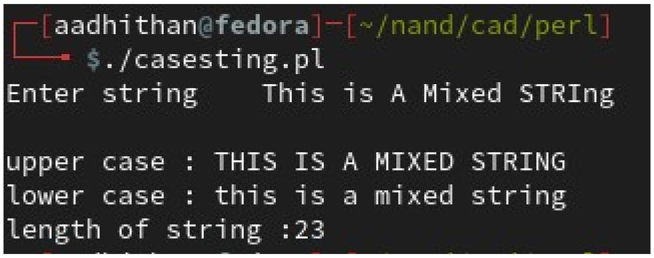
```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./pushpop.pl
original syntax:
        this is the string
push operation:
        this is the string after edit 1
pop operation:
        this is the string
shift operation:
        is the string
unshift operation:
        this is the string
```

Write a Perl program with UC(), LC() and length() functions.

Code:

```
#!/usr/bin/perl

print "Enter string  \t";
$s = <STDIN>;

print("\nupper case : " ,uc($s));
print("lower case : " ,lc($s));
print("length of string :", length($s), "\n");
```

Output:



```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./casesting.pl
Enter string    This is A Mixed STRIng

upper case : THIS IS A MIXED STRING
lower case : this is a mixed string
length of string :23
```

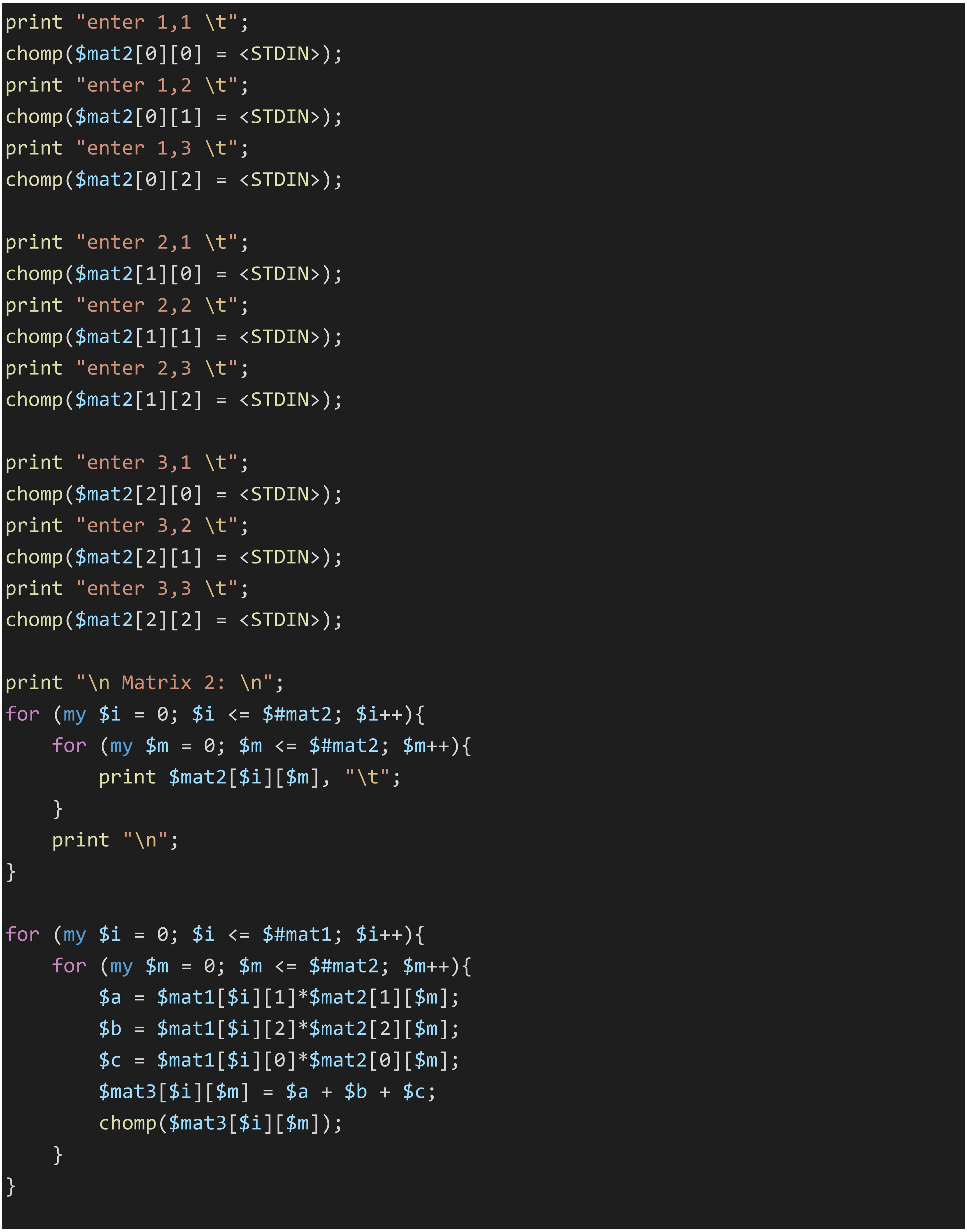
EXPERIEMENT 6

Write a Perl program to multiply two matrices.

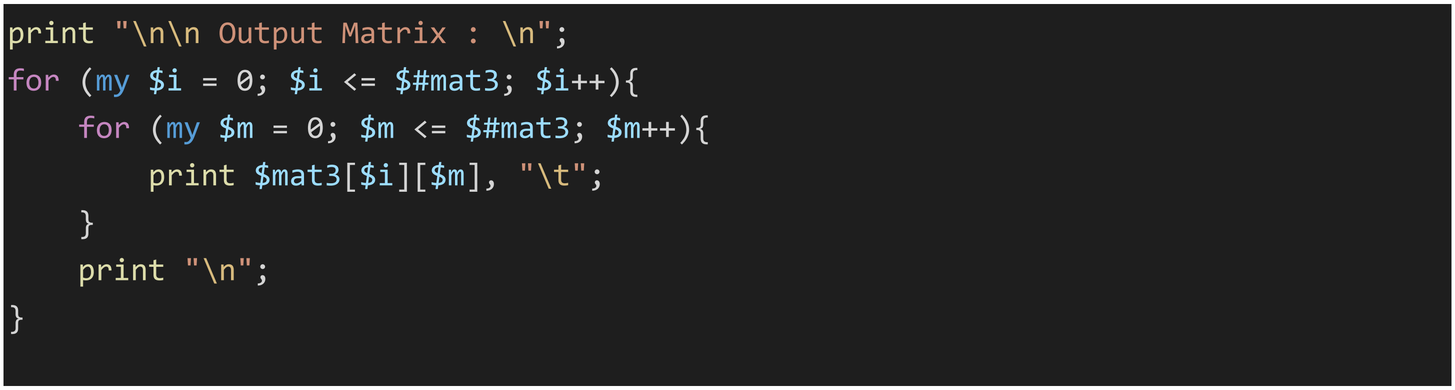
Flowchart:

Code:

```perl
#!/usr/bin/perl

my @mat1=([0,0,0],[0,0,0],[0,0,0]);
my @mat2=([0,0,0],[0,0,0],[0,0,0]);
my @mat3=([0,0,0],[0,0,0],[0,0,0]);

print " values of matrix 1: \n";
print "enter 1,1 \t";
chomp($mat1[0][0] = <STDIN>);
print "enter 1,2 \t";
chomp($mat1[0][1] = <STDIN>);
print "enter 1,3 \t";
chomp($mat1[0][2] = <STDIN>);

print "enter 2,1 \t";
chomp($mat1[1][0] = <STDIN>);
print "enter 2,2 \t";
chomp($mat1[1][1] = <STDIN>);
print "enter 2,3 \t";
chomp($mat1[1][2] = <STDIN>);

print "enter 3,1 \t";
chomp($mat1[2][0] = <STDIN>);
print "enter 3,2 \t";
chomp($mat1[2][1] = <STDIN>);
print "enter 3,3 \t";
chomp($mat1[2][2] = <STDIN>);

print " \n Matrix 1: \n";;
for (my $i = 0; $i <= $#mat1; $i++){
    for (my $m = 0; $m <= $#mat1; $m++){
        print $mat1[$i][$m], "\t";
    }
    print "\n";
}


print "\n values of matrix 2: \n";
```

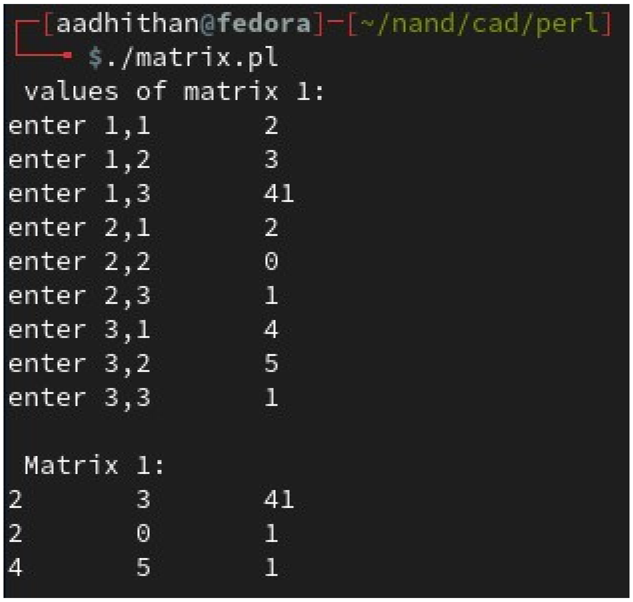


```
print "enter 1,1 \t";
chomp($mat2[0][0] = <STDIN>);
print "enter 1,2 \t";
chomp($mat2[0][1] = <STDIN>);
print "enter 1,3 \t";
chomp($mat2[0][2] = <STDIN>);

print "enter 2,1 \t";
chomp($mat2[1][0] = <STDIN>);
print "enter 2,2 \t";
chomp($mat2[1][1] = <STDIN>);
print "enter 2,3 \t";
chomp($mat2[1][2] = <STDIN>);

print "enter 3,1 \t";
chomp($mat2[2][0] = <STDIN>);
print "enter 3,2 \t";
chomp($mat2[2][1] = <STDIN>);
print "enter 3,3 \t";
chomp($mat2[2][2] = <STDIN>);

print "\n Matrix 2: \n";
for (my $i = 0; $i <= $#mat2; $i++){
    for (my $m = 0; $m <= $#mat2; $m++){
        print $mat2[$i][$m], "\t";
    }
    print "\n";
}

for (my $i = 0; $i <= $#mat1; $i++){
    for (my $m = 0; $m <= $#mat2; $m++){
        $a = $mat1[$i][1]*$mat2[1][$m];
        $b = $mat1[$i][2]*$mat2[2][$m];
        $c = $mat1[$i][0]*$mat2[0][$m];
        $mat3[$i][$m] = $a + $b + $c;
        chomp($mat3[$i][$m]);
    }
}
```

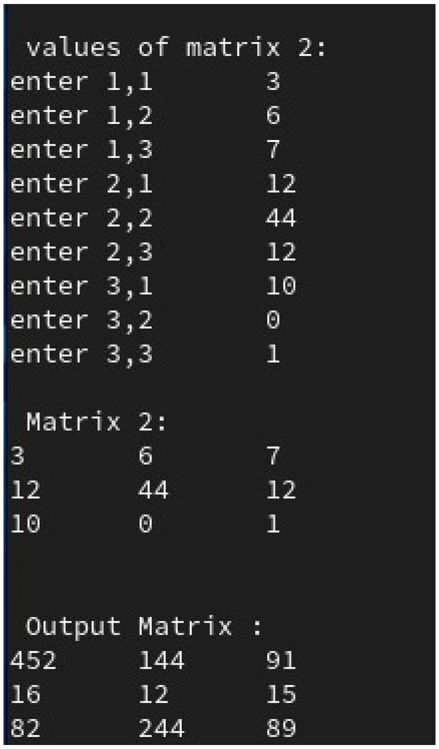


```perl
print "\n\n Output Matrix : \n";
for (my $i = 0; $i <= $#mat3; $i++){
    for (my $m = 0; $m <= $#mat3; $m++){
        print $mat3[$i][$m], "\t";
    }
    print "\n";
}
```

Output:



```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./matrix.pl
 values of matrix 1:
enter 1,1       2
enter 1,2       3
enter 1,3       41
enter 2,1       2
enter 2,2       0
enter 2,3       1
enter 3,1       4
enter 3,2       5
enter 3,3       1

 Matrix 1:
2       3       41
2       0       1
4       5       1
```

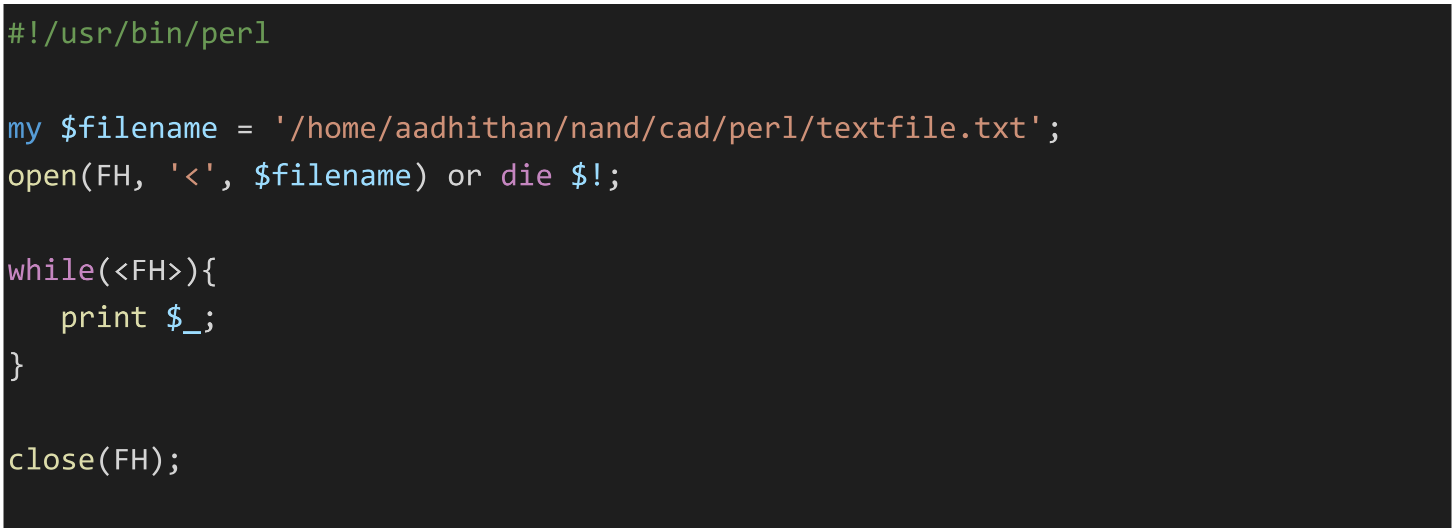


```
 values of matrix 2:
enter 1,1       3
enter 1,2       6
enter 1,3       7
enter 2,1       12
enter 2,2       44
enter 2,3       12
enter 3,1       10
enter 3,2       0
enter 3,3       1

 Matrix 2:
3       6       7
12      44      12
10      0       1


 Output Matrix :
452     144     91
16      12      15
82      244     89
```

EXPERIMENT 7

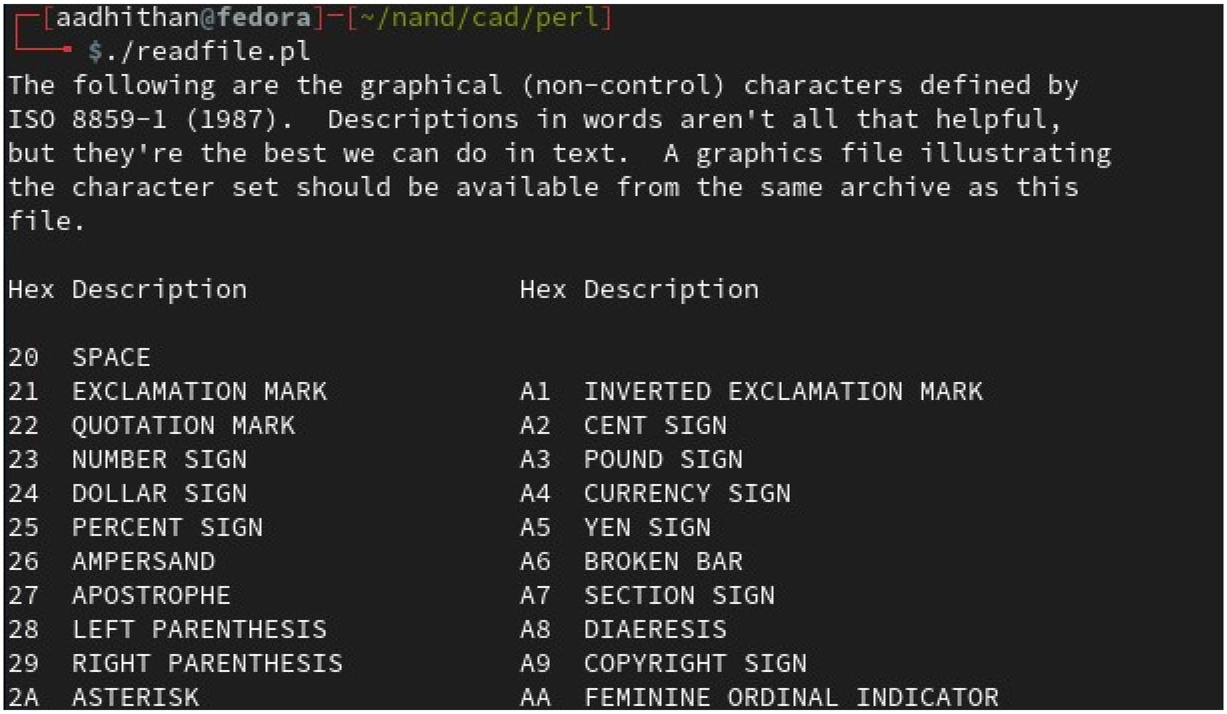
Write a perl program to read all files of a text file.

Code:



```perl
#!/usr/bin/perl

my $filename = '/home/aadhithan/nand/cad/perl/textfile.txt';
open(FH, '<', $filename) or die $!;

while(<FH>){
   print $_;
}

close(FH);
```

Output:

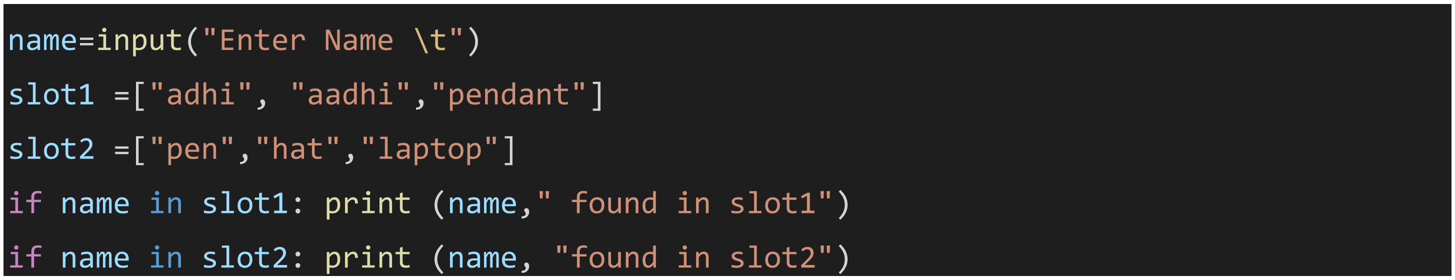


```
┌─[aadhithan@fedora]─[~/nand/cad/perl]
└──╼ $./readfile.pl
The following are the graphical (non-control) characters defined by
ISO 8859-1 (1987).  Descriptions in words aren't all that helpful,
but they're the best we can do in text.  A graphics file illustrating
the character set should be available from the same archive as this
file.

Hex Description                 Hex Description

20  SPACE
21  EXCLAMATION MARK            A1  INVERTED EXCLAMATION MARK
22  QUOTATION MARK              A2  CENT SIGN
23  NUMBER SIGN                 A3  POUND SIGN
24  DOLLAR SIGN                 A4  CURRENCY SIGN
25  PERCENT SIGN                A5  YEN SIGN
26  AMPERSAND                   A6  BROKEN BAR
27  APOSTROPHE                  A7  SECTION SIGN
28  LEFT PARENTHESIS            A8  DIAERESIS
29  RIGHT PARENTHESIS           A9  COPYRIGHT SIGN
2A  ASTERISK                    AA  FEMININE ORDINAL INDICATOR
```

EXPERIMENT 8

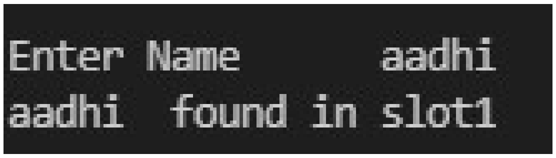
String Manipulation: to find whether an element entered by the user is present in list or not

Code:



```
name=input("Enter Name \t")
slot1 =["adhi", "aadhi","pendant"]
slot2 =["pen","hat","laptop"]
if name in slot1: print (name," found in slot1")
if name in slot2: print (name, "found in slot2")
```

Output:



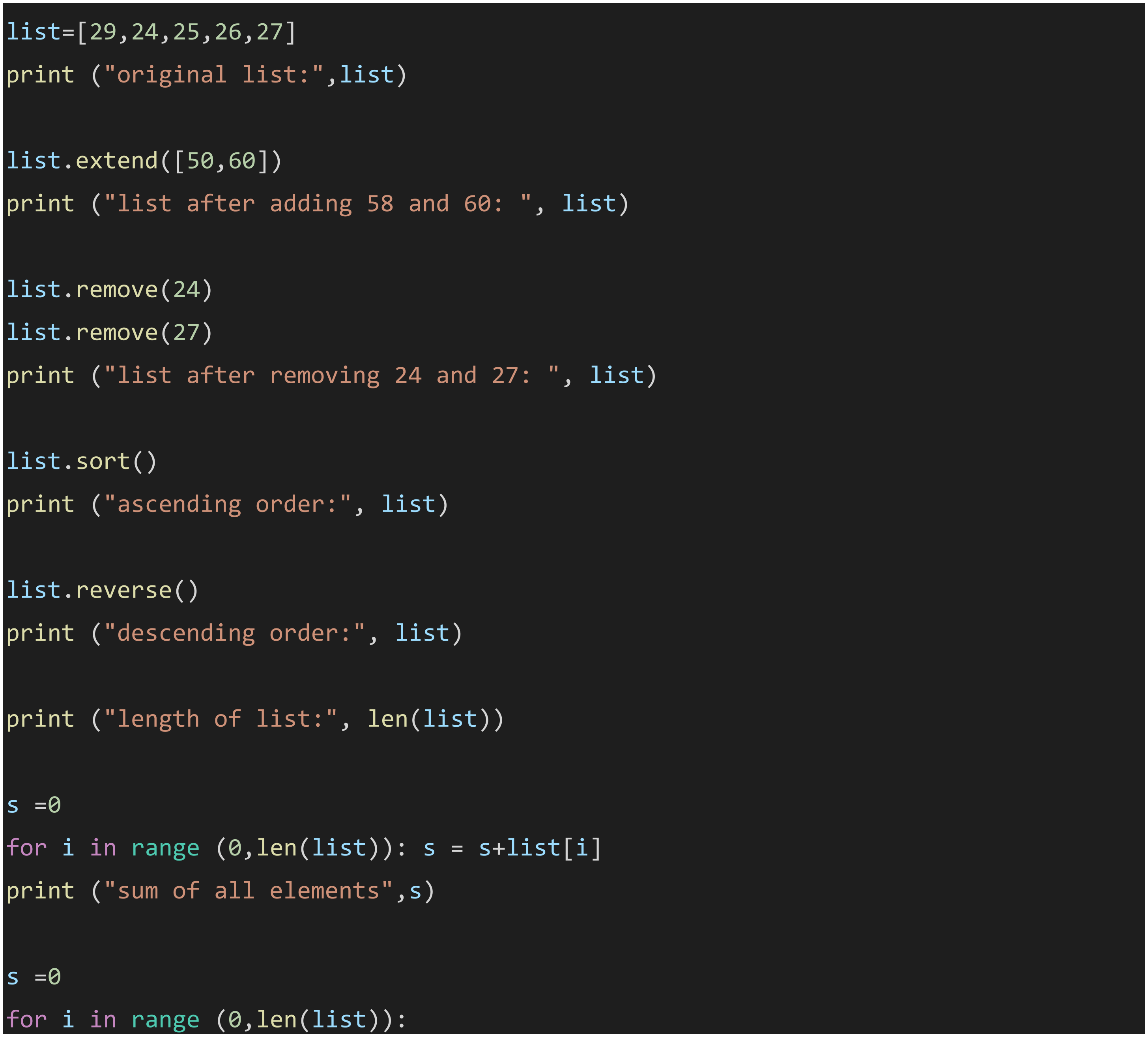
```
Enter Name      aadhi
aadhi  found in slot1
```

EXPERIMENT 9

Make a list and perform the following functions:

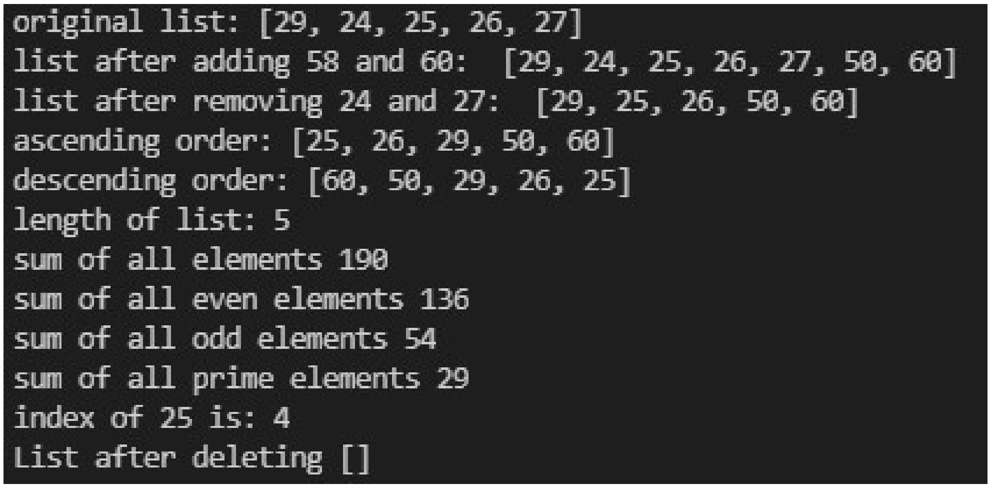
a. Reverse
b. Sort
c. Append
d. Removing an element
e. Determination of length of the list
f. Sum of odd and even elements
g. Sum of prime elements

Code:



```
list=[29,24,25,26,27]
print ("original list:",list)

list.extend([50,60])
print ("list after adding 58 and 60: ", list)

list.remove(24)
list.remove(27)
print ("list after removing 24 and 27: ", list)

list.sort()
print ("ascending order:", list)

list.reverse()
print ("descending order:", list)

print ("length of list:", len(list))

s =0
for i in range (0,len(list)): s = s+list[i]
print ("sum of all elements",s)

s =0
for i in range (0,len(list)):
    if(list[i]%2 ==0) :s = s+list[i]
print ("sum of all even elements",s)

s =0
for i in range (0,len(list)):
    if(list[i]%2 ==1) :s = s+list[i]
print ("sum of all odd elements",s)

c =0
s =0
list2 =[]
for i in range (0,len(list)):
    for a in range (2,list[i]):
        if (list[i]%a == 0): c = c+1
    if (c==0):list2.append(list[i])
    c =0
for b in range (0, len(list2)): s = s+list2[b]
print ("sum of all prime elements",s)

index= list.index(25)
print ("index of 25 is:",index)
del list[:]
print ("List after deleting",list)
```

Output:

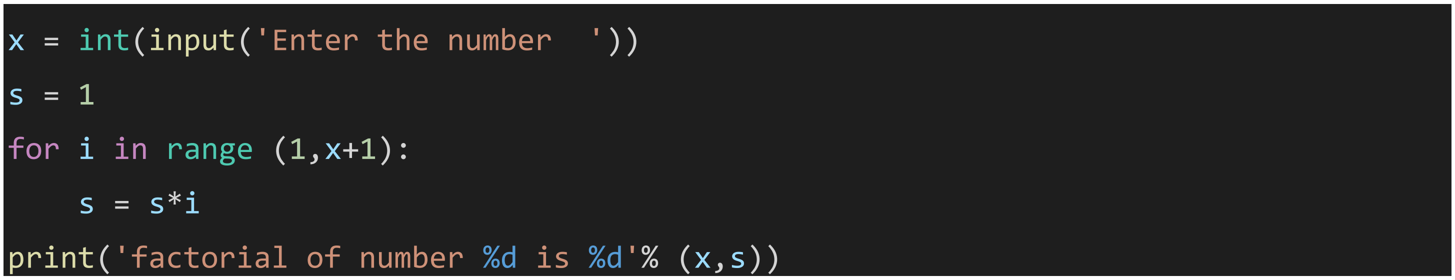


```
original list: [29, 24, 25, 26, 27]
list after adding 58 and 60:  [29, 24, 25, 26, 27, 50, 60]
list after removing 24 and 27:  [29, 25, 26, 50, 60]
ascending order: [25, 26, 29, 50, 60]
descending order: [60, 50, 29, 26, 25]
length of list: 5
sum of all elements 190
sum of all even elements 136
sum of all odd elements 54
sum of all prime elements 29
index of 25 is: 4
List after deleting []
```

EXPERIEMENT 10

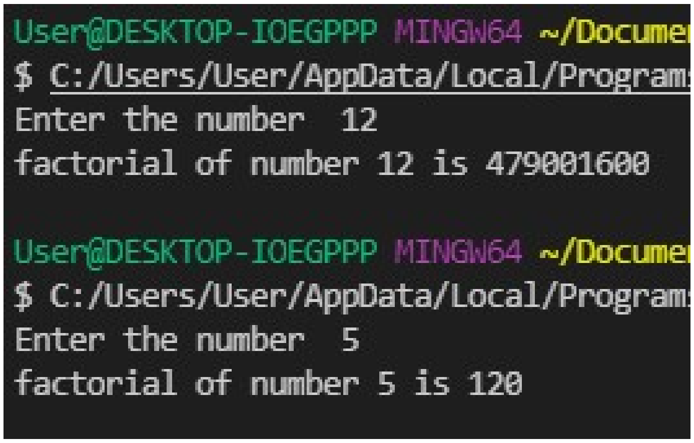
Calculation of factorial of a number

Code:



```
x = int(input('Enter the number  '))
s = 1
for i in range (1,x+1):
    s = s*i
print('factorial of number %d is %d'% (x,s))
```

Output:



```
User@DESKTOP-IOEGPPP MINGW64 ~/Docume
$ C:/Users/User/AppData/Local/Program
Enter the number  12
factorial of number 12 is 479001600

User@DESKTOP-IOEGPPP MINGW64 ~/Docume
$ C:/Users/User/AppData/Local/Program
Enter the number  5
factorial of number 5 is 120
```

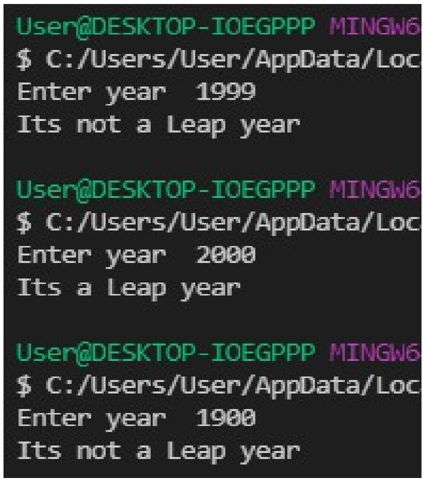
WAP to check whether the year entered by the user is leap year or not

Flowchart:

Code:

```
x = input('Enter year  ')
try :
    x = int(x)
    if x%4 == 0 and x%100 != 0:
        print('Its a Leap year')
    elif x%400 == 0:
        print('Its a Leap year')
    else:
        print('Its not a Leap year')
except:
    print('Year entered is invalid')
```

Output:



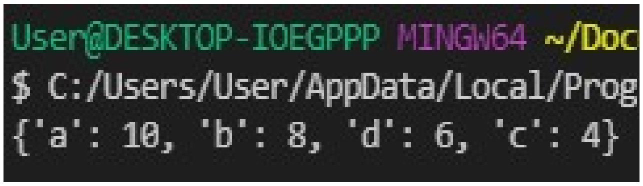
```
User@DESKTOP-IOEGPPP MINGW6
$ C:/Users/User/AppData/Loc
Enter year  1999
Its not a Leap year

User@DESKTOP-IOEGPPP MINGW6
$ C:/Users/User/AppData/Loc
Enter year  2000
Its a Leap year

User@DESKTOP-IOEGPPP MINGW6
$ C:/Users/User/AppData/Loc
Enter year  1900
Its not a Leap year
```

WAP to merge dictionary using update() method

Code:

```
def Merge(dict1, dict2):
    return(dict1.update(dict2))
dict1 = {'a': 10, 'b': 8}
dict2 = {'d': 6, 'c': 4}
Merge(dict1, dict2)
print(dict1)
```

Output:

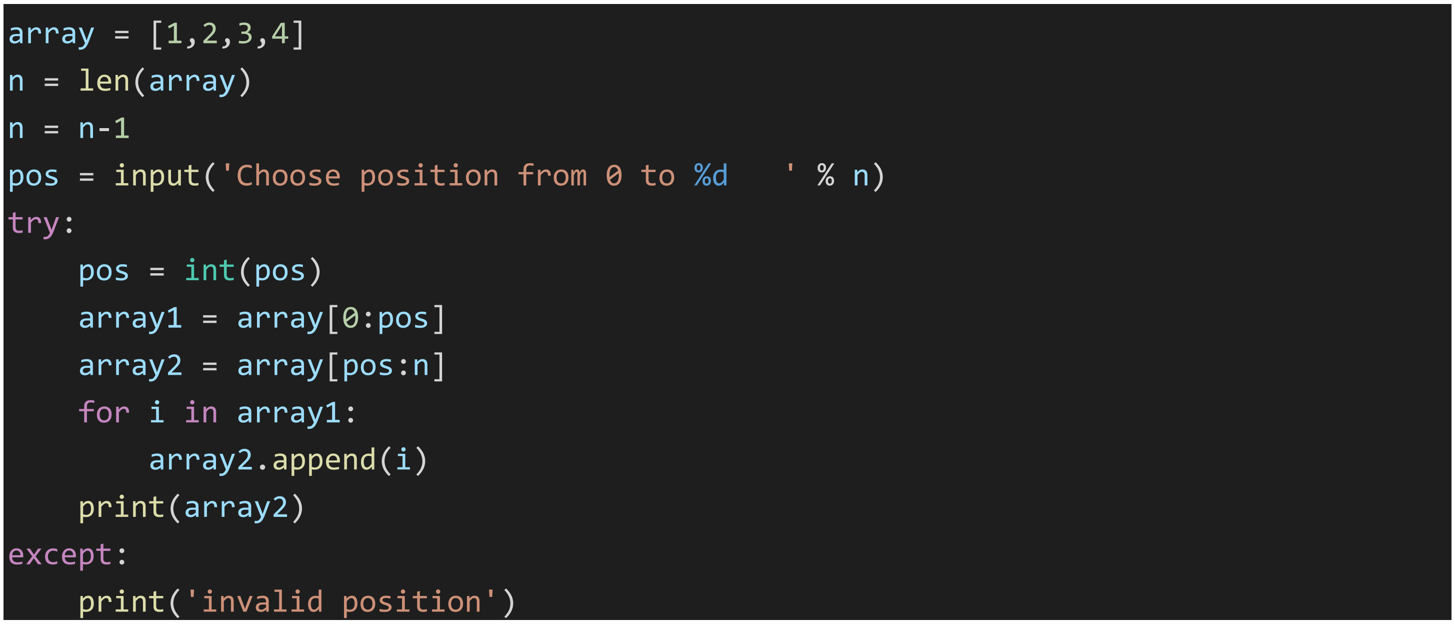


```
User@DESKTOP-IOEGPPP MINGW64 ~/Doc
$ C:/Users/User/AppData/Local/Prog
{'a': 10, 'b': 8, 'd': 6, 'c': 4}
```

EXPERIEMENT 11

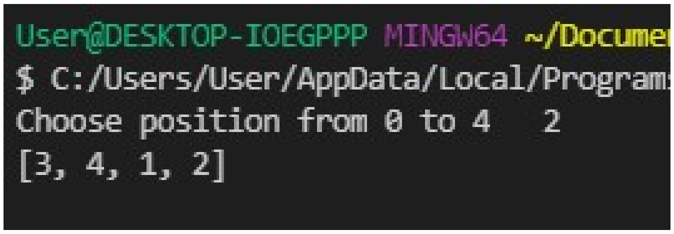
WAP to split the array and add the first two elements to the end

Code:



```
array = [1,2,3,4]
n = len(array)
n = n-1
pos = input('Choose position from 0 to %d   ' % n)
try:
    pos = int(pos)
    array1 = array[0:pos]
    array2 = array[pos:n]
    for i in array1:
        array2.append(i)
    print(array2)
except:
    print('invalid position')
```

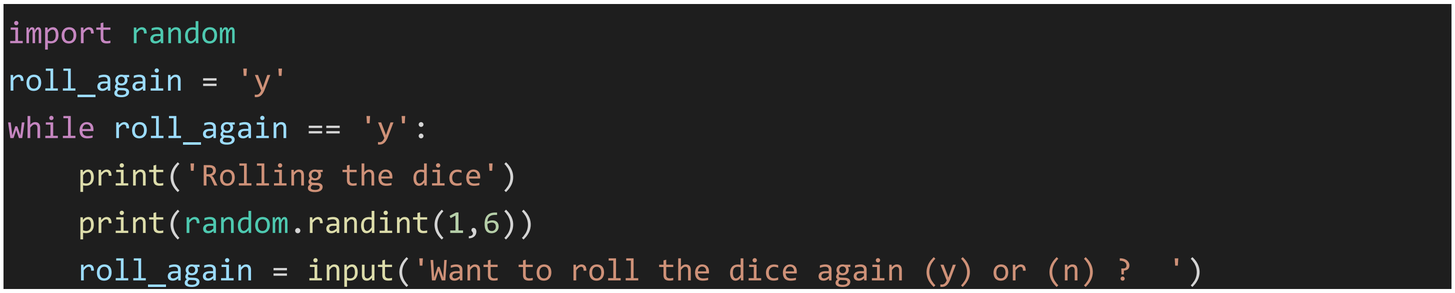
Output:



```
User@DESKTOP-IOEGPPP MINGW64 ~/Documen
$ C:/Users/User/AppData/Local/Program
Choose position from 0 to 4   2
[3, 4, 1, 2]
```

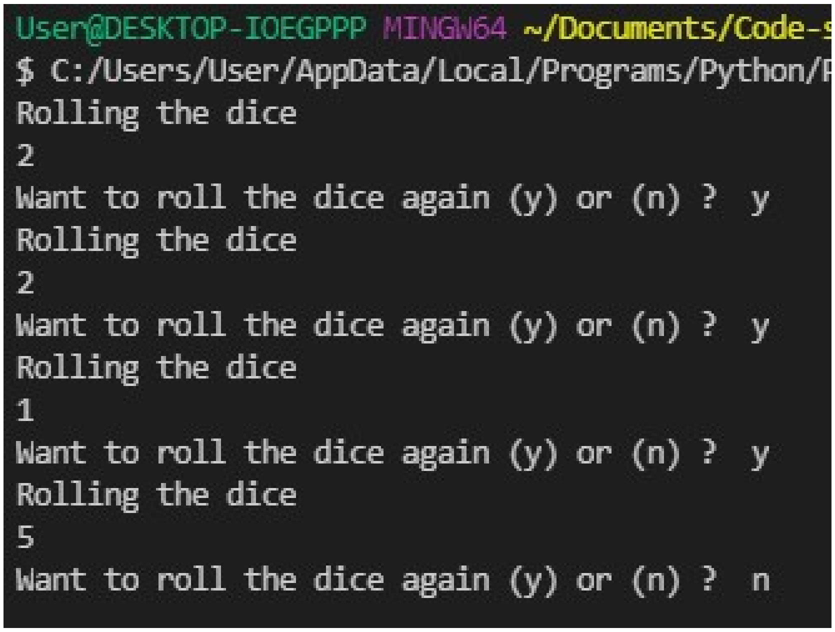
WAP to roll the dice. Use the random module

Code:



```
import random
roll_again = 'y'
while roll_again == 'y':
    print('Rolling the dice')
    print(random.randint(1,6))
    roll_again = input('Want to roll the dice again (y) or (n) ?  ')
```

Output:



```
User@DESKTOP-IOEGPPP MINGW64 ~/Documents/Code-
$ C:/Users/User/AppData/Local/Programs/Python/
Rolling the dice
2
Want to roll the dice again (y) or (n) ?  y
Rolling the dice
2
Want to roll the dice again (y) or (n) ?  y
Rolling the dice
1
Want to roll the dice again (y) or (n) ?  y
Rolling the dice
5
Want to roll the dice again (y) or (n) ?  n
```

EXPERIEMENT 12

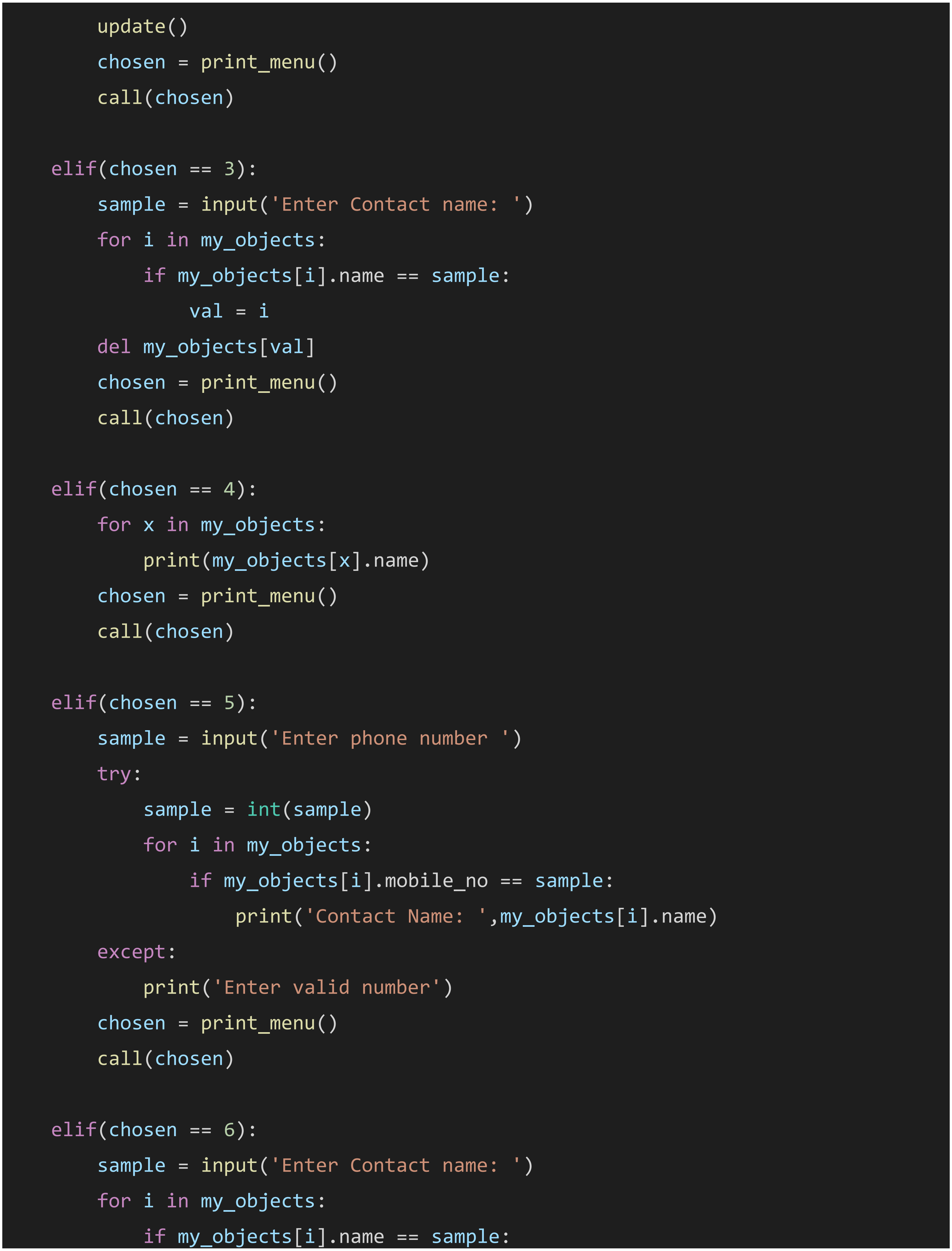
Create a directory and perform the following functions:

a. **Addition of name and details**
b. **Updating the details**
c. **Removal of content**
d. **Displaying the content**

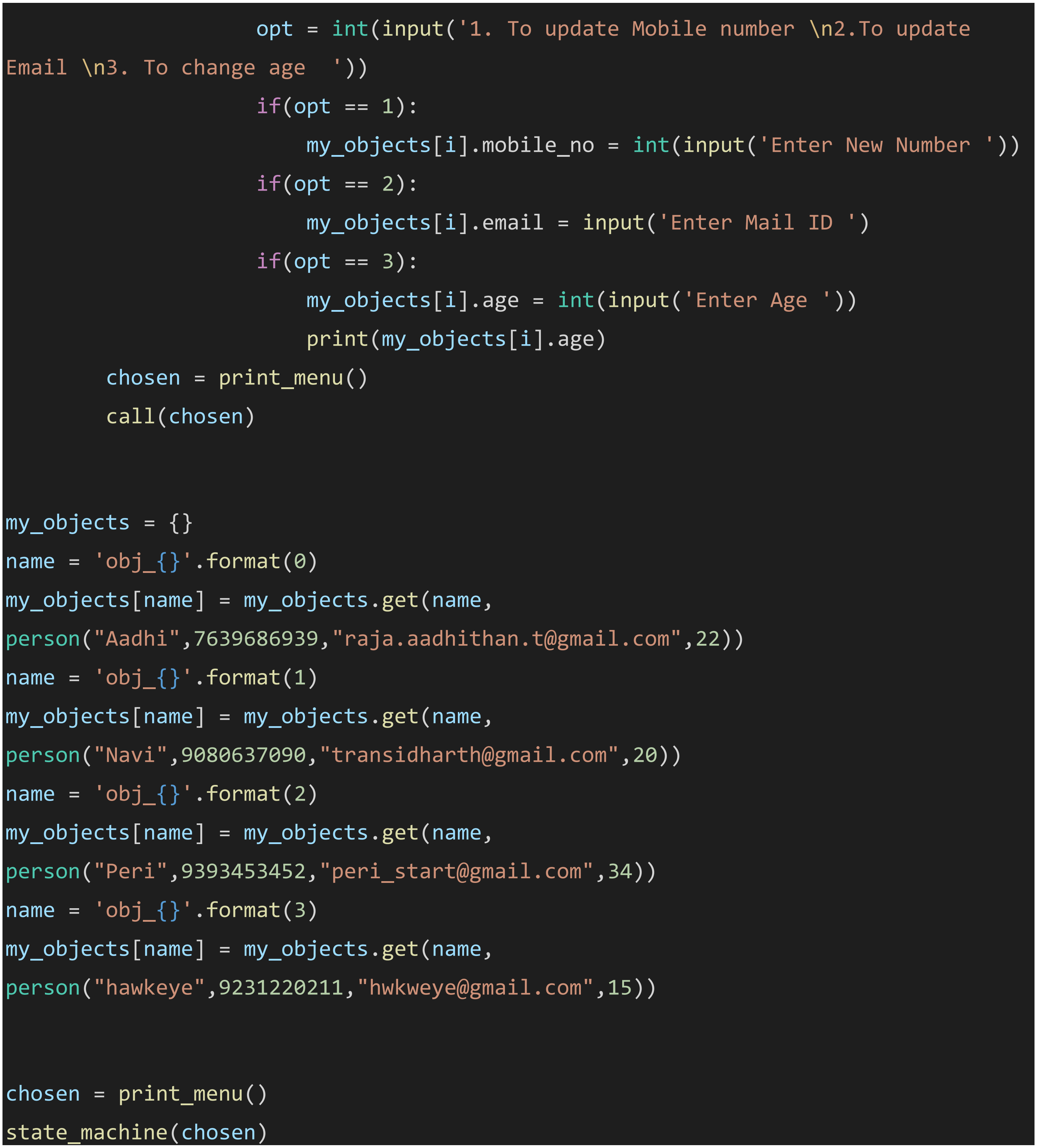
Flowchart:

Code:

```
#directory

def update():
    name = 'obj_{}'.format(len(my_objects))
    nam = input("Enter Contacts Name: ")
    num =  input("Enter Contacts Mobile no: ")
    email = input("Enter Email: ")
    age = input("Enter Age: ")
    my_objects[name] = my_objects.get(name, person(nam, num, email, age))

def print_menu():
    print('\n\n\n\n Menu: \n')
    print('1. Print Details')
    print('2. Add a Contact')
    print('3. Remove a Contact ')
    print('4. Show all contacts ')
    print('5. Lookup a Phone Number')
    print('6. Update a Contact')
    print('To quit enter any other number')
    choice = input('Enter a Choice from 1 to 6: ')
    try:
        choice = int(choice)
        if(choice >= 1 and choice < 7):
            return(choice)
        else:
            print('Quitting')
    except:
        print('Invalid choice')



class person:
    def __init__(self, name, mobile_no, email, age):
        self.name = name
        self.mobile_no = mobile_no
        self.email = email
        self.age = age
    def show(self):
        print('\t Contact :',self.name,'\n\t Mobile no:',self.mobile_no,'\n\t Email:',self.email,'\n\t Age:',self.age)

def call(chosen):
    state_machine(chosen)

def state_machine(chosen):
    if(chosen == 1):
        contact = input('Enter the name of the contact: ')
        for i in my_objects:
            if my_objects[i].name == contact:
                print('Contact Exists')
                no = input('Enter Details needed: \n\t 1. Mobile no \n\t 2. Email \n\t 3. Age \n\t 4. All \n')
                if (no == '1' or no == 'mobile_no' or no == 'Mobile_no'or no == 'Mobile'or no == 'mobile'):
                    print(my_objects[i].mobile_no)
                elif (no == '2' or no == 'email' or no == 'Email'or no == 'mail'):
                    print(my_objects[i].email)
                elif (no == '3' or no == 'age' or no == 'Age'):
                    print(my_objects[i].age)
                elif (no == '4' or no == 'all' or no == 'All'):
                    my_objects[i].show()
                else:
                    print('not a valid option')
        chosen = print_menu()
        call(chosen)


    elif(chosen == 2):
```

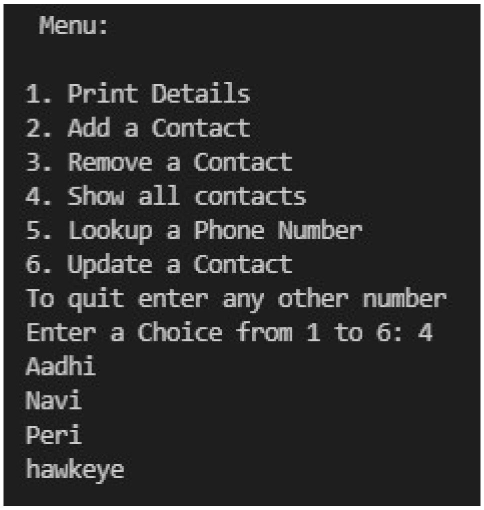


```python
        update()
        chosen = print_menu()
        call(chosen)

    elif(chosen == 3):
        sample = input('Enter Contact name: ')
        for i in my_objects:
            if my_objects[i].name == sample:
                val = i
        del my_objects[val]
        chosen = print_menu()
        call(chosen)

    elif(chosen == 4):
        for x in my_objects:
            print(my_objects[x].name)
        chosen = print_menu()
        call(chosen)

    elif(chosen == 5):
        sample = input('Enter phone number ')
        try:
            sample = int(sample)
            for i in my_objects:
                if my_objects[i].mobile_no == sample:
                    print('Contact Name: ',my_objects[i].name)
        except:
            print('Enter valid number')
        chosen = print_menu()
        call(chosen)

    elif(chosen == 6):
        sample = input('Enter Contact name: ')
        for i in my_objects:
            if my_objects[i].name == sample:
```

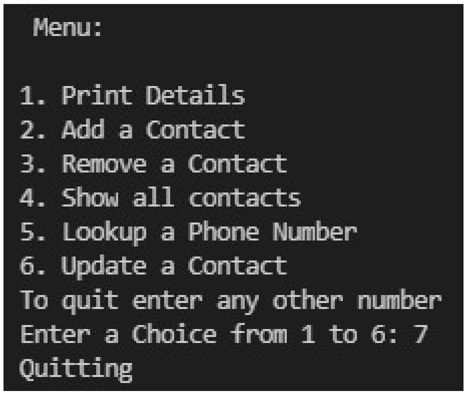


```
                    opt = int(input('1. To update Mobile number \n2.To update
Email \n3. To change age  '))
                    if(opt == 1):
                        my_objects[i].mobile_no = int(input('Enter New Number '))
                    if(opt == 2):
                        my_objects[i].email = input('Enter Mail ID ')
                    if(opt == 3):
                        my_objects[i].age = int(input('Enter Age '))
                        print(my_objects[i].age)
        chosen = print_menu()
        call(chosen)


my_objects = {}
name = 'obj_{}'.format(0)
my_objects[name] = my_objects.get(name,
person("Aadhi",7639686939,"raja.aadhithan.t@gmail.com",22))
name = 'obj_{}'.format(1)
my_objects[name] = my_objects.get(name,
person("Navi",9080637090,"transidharth@gmail.com",20))
name = 'obj_{}'.format(2)
my_objects[name] = my_objects.get(name,
person("Peri",9393453452,"peri_start@gmail.com",34))
name = 'obj_{}'.format(3)
my_objects[name] = my_objects.get(name,
person("hawkeye",9231220211,"hwkweye@gmail.com",15))


chosen = print_menu()
state_machine(chosen)
```

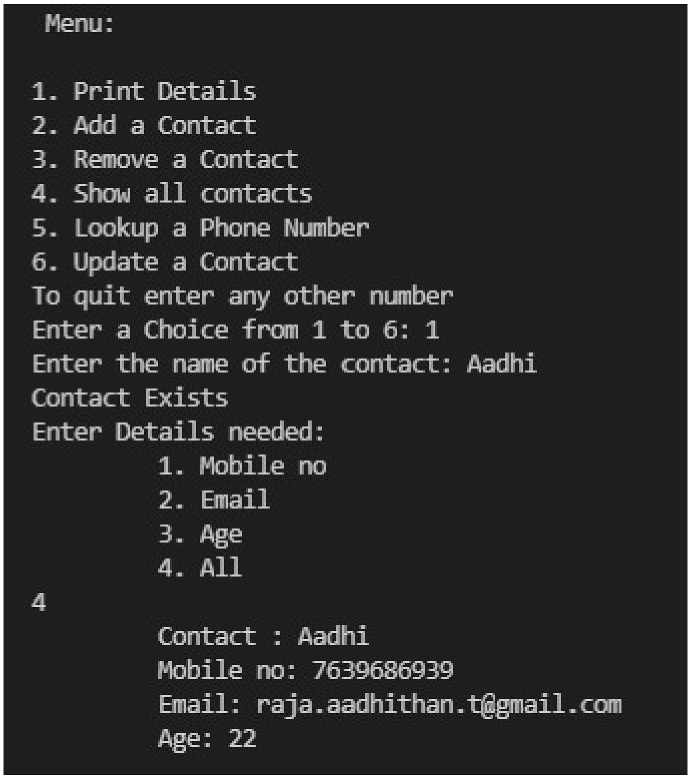
Output:



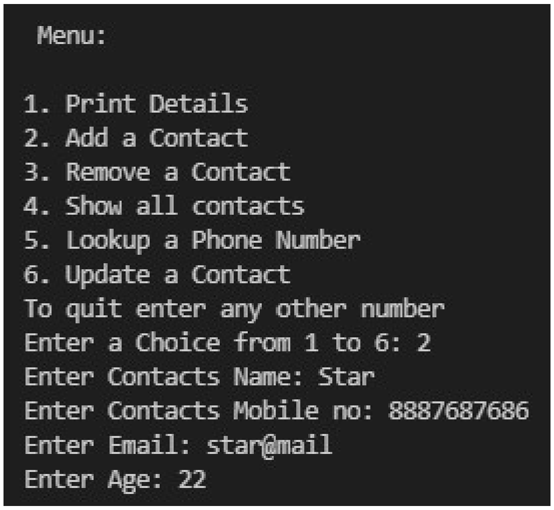
```
 Menu:

1. Print Details
2. Add a Contact
3. Remove a Contact
4. Show all contacts
5. Lookup a Phone Number
6. Update a Contact
To quit enter any other number
Enter a Choice from 1 to 6: 4
Aadhi
Navi
Peri
hawkeye
```



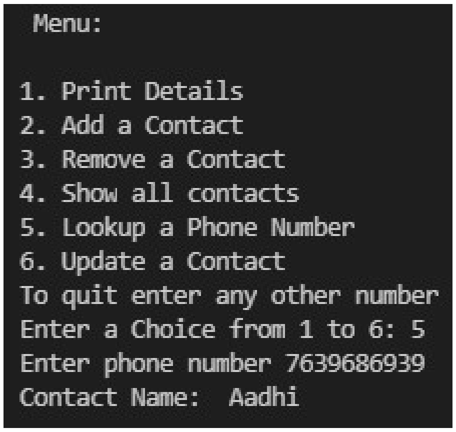
```
 Menu:

1. Print Details
2. Add a Contact
3. Remove a Contact
4. Show all contacts
5. Lookup a Phone Number
6. Update a Contact
To quit enter any other number
Enter a Choice from 1 to 6: 7
Quitting
```



```
 Menu:

1. Print Details
2. Add a Contact
3. Remove a Contact
4. Show all contacts
5. Lookup a Phone Number
6. Update a Contact
To quit enter any other number
Enter a Choice from 1 to 6: 1
Enter the name of the contact: Aadhi
Contact Exists
Enter Details needed:
         1. Mobile no
         2. Email
         3. Age
         4. All
4
         Contact : Aadhi
         Mobile no: 7639686939
         Email: raja.aadhithan.t@gmail.com
         Age: 22
```



```
 Menu:

1. Print Details
2. Add a Contact
3. Remove a Contact
4. Show all contacts
5. Lookup a Phone Number
6. Update a Contact
To quit enter any other number
Enter a Choice from 1 to 6: 2
Enter Contacts Name: Star
Enter Contacts Mobile no: 8887687686
Enter Email: star@mail
Enter Age: 22
```



```
 Menu:

1. Print Details
2. Add a Contact
3. Remove a Contact
4. Show all contacts
5. Lookup a Phone Number
6. Update a Contact
To quit enter any other number
Enter a Choice from 1 to 6: 5
Enter phone number 7639686939
Contact Name:  Aadhi
```