**EXPERIMENT 1**

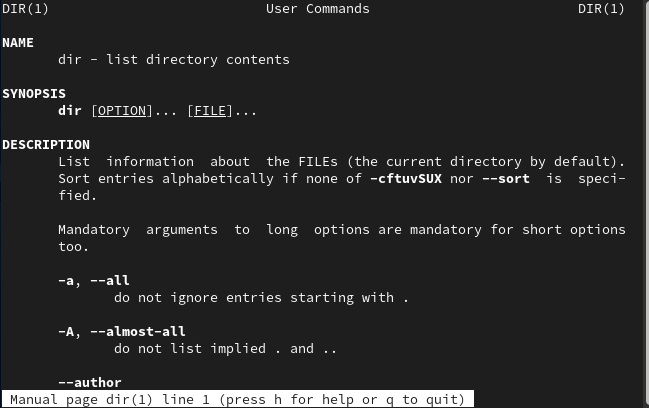
**Aim: Study of Unix general purpose utility command list.**

**man command:** Displays help of the following command keyword.

i/p:

****

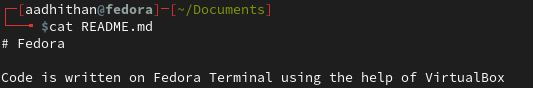
o/p:

****

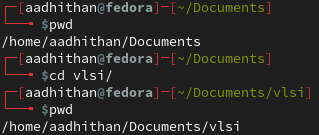
**who command:** Displays the current users of the system.



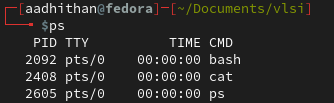
**cat command:** Displays the content of the file in the shell terminal.



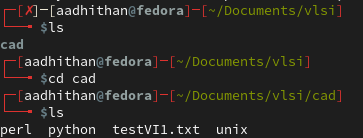
**cd command:** Navigates across directories:



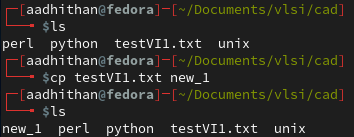
**ps command:** Lists the current working processes



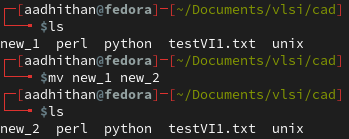
**ls command:** lists the content of the directory



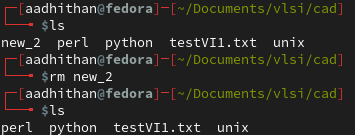
**cp command:** copies the file



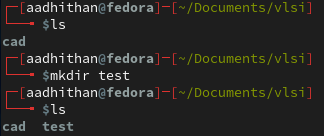
**mv command:** moves the file to new destination



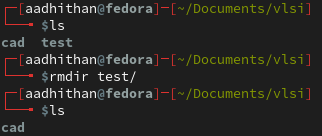
**rm command:** removes/deletes the file



**mkdir command:** creates a new directory



**rmdir command:** deletes an existing directory



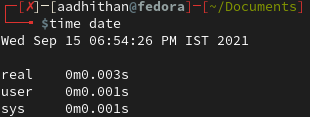
**echo command:** prints the text that follows the command



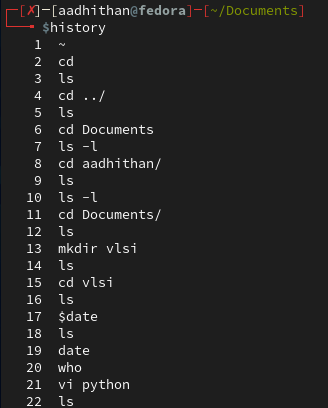
**date command:** prints the current date and time



**time command:** displays the time required to execute a command



**history command:** Displays the history of commands



**pwd command:** prints current working directory

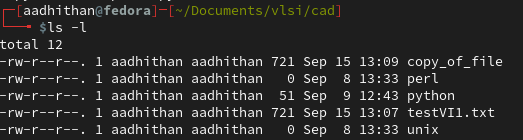
****

**shutdown command:** schedules shutdown to happen in 2 minutes

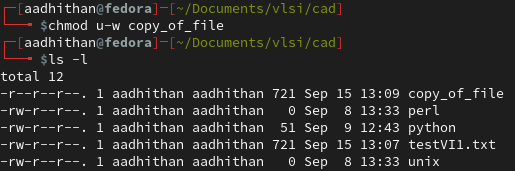
****

**chmod command:** used to change preferences of a file

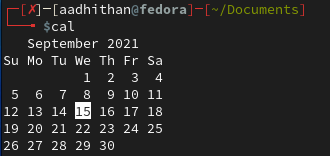
Before changing user preferences:



Changing preferences:



**cal command:** Displays the calendar



**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

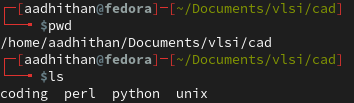
**finger command:** gets details about a domain

**logout command:** allows to programmatically logout from the session

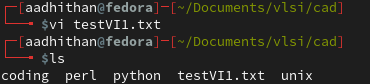
**EXPERIMENT 2: File Handling**

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

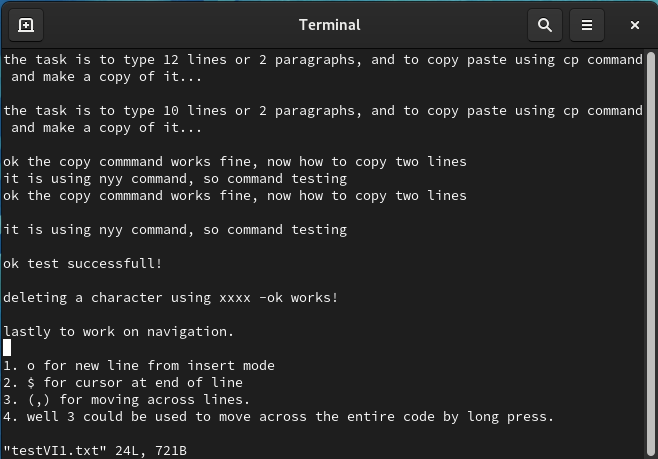
cad folder before the creation of file:

****

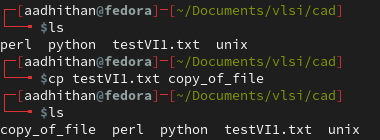
cad folder after the creation of file:



The text file created:

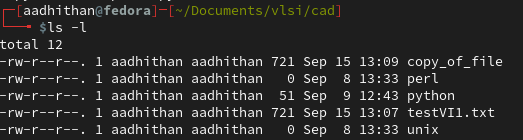
****

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

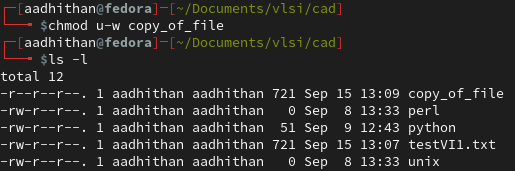
****

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

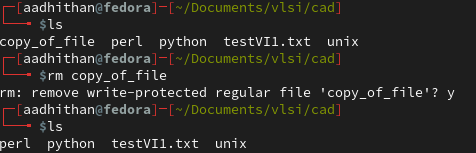
Before changing user preferences:



Changing preferences:



Deleting the file:



**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:

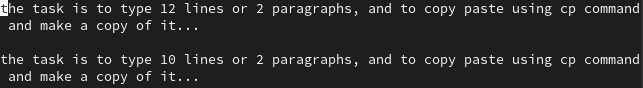
1. o for new line from insert mode

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:



Deleting in the file:

1. x deletes the letter the cursor is pointing

2. dd deletes the line

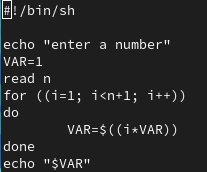
3. ndd deletes n number of lines from the cursor position

**EXPERIMENT 3: Shell Scripting**

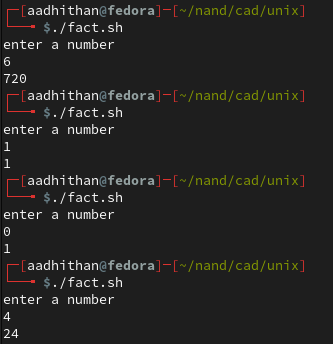
1. **Write a Shell Script program to find factorial of a number.**

**Flowchart:**

**Code:**

****

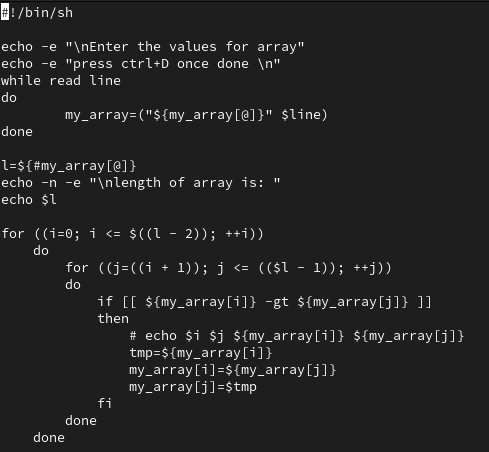
**Output:**

****

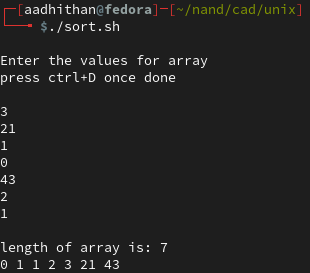
1. **Write a Shell Script program to sort an array in ascending order.**

**Flowchart:**

**Code:**

****

**Output:**

****

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

**Code:**

****

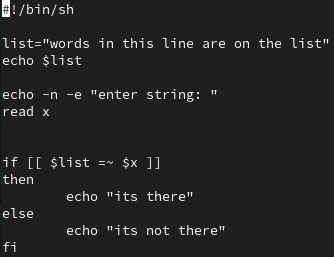
**Output:**

****

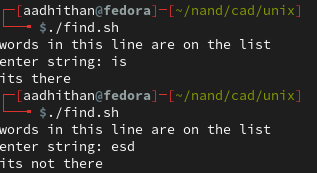
1. **Write a Shell Script program to search whether element is present in the list or not.**

**Flowchart:**

**Code:**

****

**Output:**

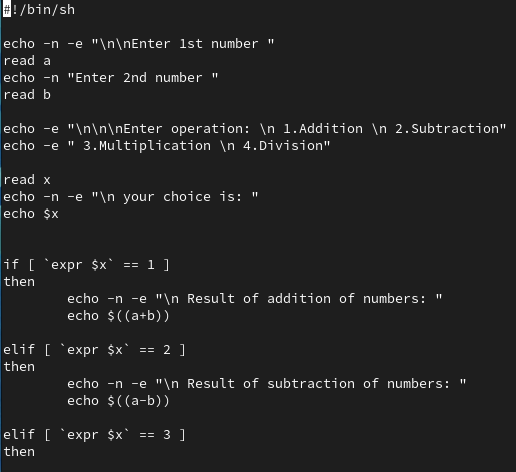
****

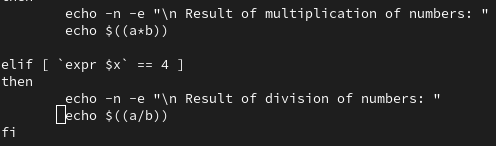
**EXPERIMENT 4: VI Editor**

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

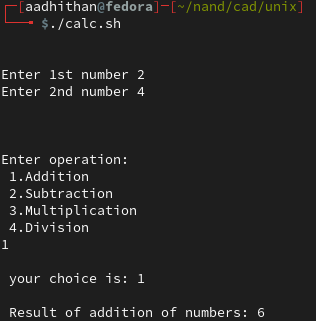
**Flowchart:**

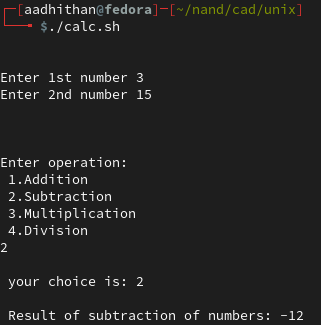
**Code:**

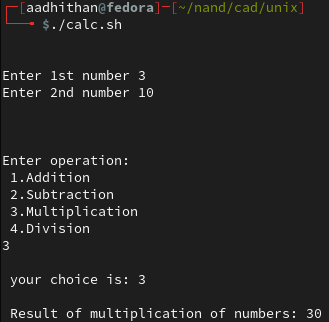
****

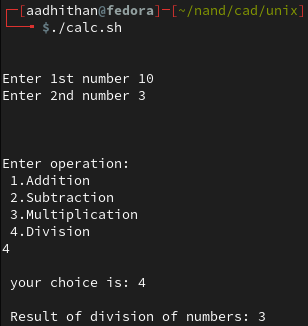
****

**Output:**

****

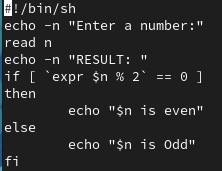
****

****

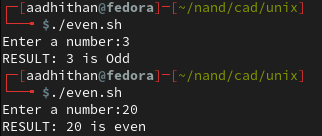
****

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

**Code:**

****

**Output:**

****

**EXPERIMENT 5**

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

**Code:**

#!/usr/bin/perl

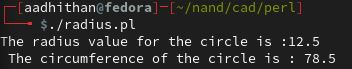
$radius = 12.5;

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:**



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

**Code:**

#!/usr/bin/perl

print "enter number 1 \n";

$n1 = <STDIN>;

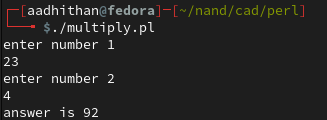
print "enter number 2 \n";

$n2 = <STDIN>;

$s = $n1\*$n2;

print "answer is $s \n";

**Output:**

****

**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

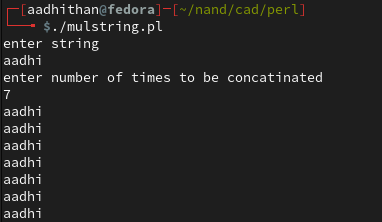
print "enter number \t";

$n = <STDIN>;

$s = $n\*\*3;

print "cube is  $s \n";

**Output:**

****

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

**Code:**

#!/usr/bin/perl

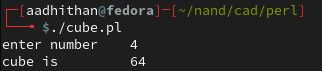
print "enter number \t";

$n = <STDIN>;

$s = $n\*\*3;

print "cube is  $s \n";

**Output:**

****

**Write a code to explore String operators.**

**Code:**

!/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";

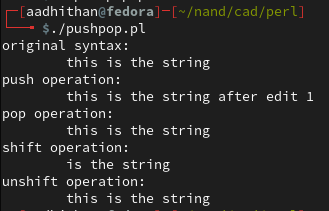
print "@s \n";

unshift(@s, 'this');

print "unshift operation: \n \t";

print "@s \n";

**Output:**

****

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

**Code:**

#!/usr/bin/perl

print "Enter string  \t";

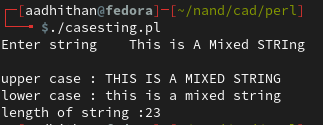
$s = <STDIN>;

print("\nupper case : " ,uc($s));

print("lower case : " ,lc($s));

print("length of string :", length($s), "\n");

**Output:**

****

**EXPERIEMENT 6**

**Write a Perl program to multiply two matrices.**

**Flowchart:**

**Code:**

#!/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]);

    }

}

print "\n\n Output Matrix : \n";

for (my $i = 0; $i <= $#mat3; $i++){

    for (my $m = 0; $m <= $#mat3; $m++){

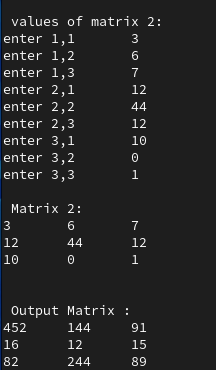
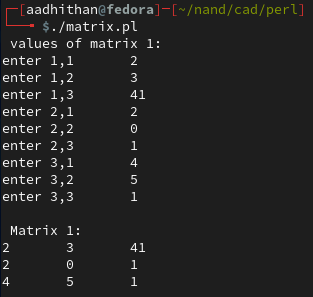
        print $mat3[$i][$m], "\t";

    }

    print "\n";

}

**Output:**

****

**EXPERIMENT 7**

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

**Code:**

#!/usr/bin/perl

my $filename = '/home/aadhithan/nand/cad/perl/textfile.txt';

open(FH, '<', $filename) or die $!;

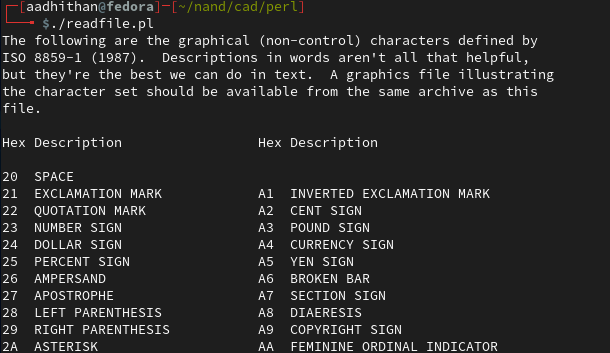
while(<FH>){

   print $\_;

}

close(FH);

**Output:**

****

**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:**

****

**EXPERIMENT 9**

**Make a list and perform the following functions:**

1. **Reverse**
2. **Sort**
3. **Append**
4. **Removing an element**
5. **Determination of length of the list**
6. **Sum of odd and even elements**
7. **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)

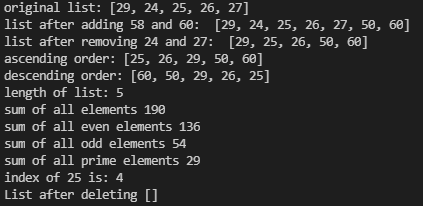
index= list.index(25)

print ("index of 25 is:",index)

del list[:]

print ("List after deleting",list)

**Output:**

****

**EXPERIEMENT 10**

**Calculation of factorial of a number**

**Code:**

x = int(input('Enter the number  '))

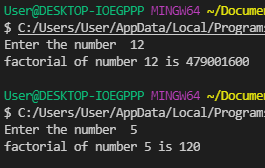
s = 1

for i in range (1,x+1):

    s = s\*i

print('factorial of number %d is %d'% (x,s))

**Output:**

****

**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')

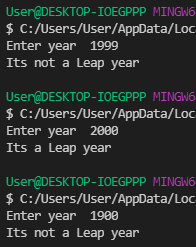
    else:

        print('Its not a Leap year')

except:

    print('Year entered is invalid')

**Output:**

****

**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:**

****

**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:

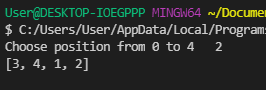
        array2.append(i)

    print(array2)

except:

    print('invalid position')

**Output:**

****

**WAP to roll the dice. Use the random module**

**Code:**

import random

roll\_again = 'y'

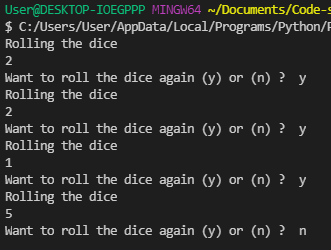
while roll\_again == 'y':

    print('Rolling the dice')

    print(random.randint(1,6))

    roll\_again = input('Want to roll the dice again (y) or (n) ?  ')

**Output:**

****

**EXPERIEMENT 12**

**Create a directory and perform the following functions:**

1. **Addition of name and details**
2. **Updating the details**
3. **Removal of content**
4. **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):

        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))

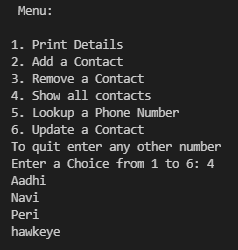
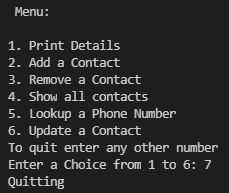
name = 'obj\_{}'.format(3)

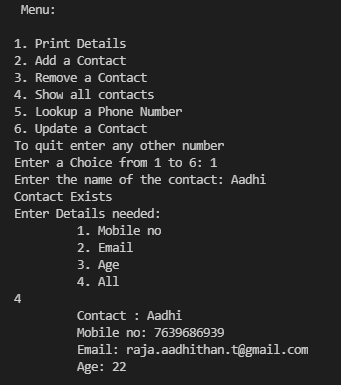
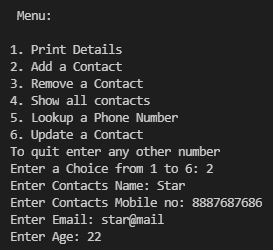
my\_objects[name] = my\_objects.get(name, person("hawkeye",9231220211,"hwkweye@gmail.com",15))

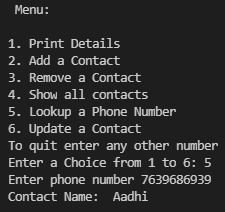
chosen = print\_menu()

state\_machine(chosen)

**Output:**

** **

** **

****