**1 .Consider a list (list = []). You can perform the following commands:**

insert i e: Insert integer at position .

print: Print the list.

remove e: Delete the first occurrence of integer .

append e: Insert integer at the end of the list.

sort: Sort the list.

pop: Pop the last element from the list.

reverse: Reverse the list.

Initialize your list and read in the value of followed by lines of commands where each command will be of the types listed above. Iterate through each command in order and perform the corresponding operation on your list.

if \_name\_ == '\_main\_':

N = int(input())

lists = []

for i in range(N):

a = list(map(str,input().split( )))

lists.append(a)

arr = []

for x in lists:

if x[0] == "insert":

i = int(x[1])

e = int(x[2])

arr.insert(i,e)

elif x[0] == "print":

print(arr)

elif x[0] == "remove":

e = int(x[1])

arr.remove(e)

elif x[0] == "append":

e = int(x[1])

arr.append(e)

elif x[0] == "sort":

arr.sort()

elif x[0] == "pop":

arr.pop()

elif x[0] == "reverse":

arr.reverse()

**2 . Program make a simple calculator**

# This function adds two numbers

def add(x, y):

return x + y

# This function subtracts two numbers

def subtract(x, y):

return x - y

# This function multiplies two numbers

def multiply(x, y):

return x \* y

# This function divides two numbers

def divide(x, y):

return x / y

print("Select operation.")

print("1.Add")

print("2.Subtract")

print("3.Multiply")

print("4.Divide")

while True:

# take input from the user

choice = input("Enter choice(1/2/3/4): ")

# check if choice is one of the four options

if choice in ('1', '2', '3', '4'):

num1 = float(input("Enter first number: "))

num2 = float(input("Enter second number: "))

if choice == '1':

print(num1, "+", num2, "=", add(num1, num2))

elif choice == '2':

print(num1, "-", num2, "=", subtract(num1, num2))

elif choice == '3':

print(num1, "\*", num2, "=", multiply(num1, num2))

elif choice == '4':

print(num1, "/", num2, "=", divide(num1, num2))

# check if user wants another calculation

# break the while loop if answer is no

next\_calculation = input("Let's do next calculation? (yes/no): ")

if next\_calculation == "no":

break

else:

print("Invalid Input")

**3.Write a program to concatenate, reverse and slice a string?**

**Concatenation**

def reversed\_string(text):

result = ""

for char in text:

result = char + result

return result

reversed\_string("Hello, World!")

def reversed\_string(text):

'!dlroW ,olleH'

**Reverse**

def reversed\_string(text):

result = ""

index = len(text) - 1

while index >= 0:

result += text[index]

index -= 1

return result

reversed\_string("Hello, World!")

'!dlroW ,olleH'

**Slicing**

greeting = "Hello, World!"

for char in greeting[::-1]:

print(char)

!

d

l

r

o

W

,

o

l

l

e

H

greeting[::-1]

'!dlroW ,olleH'

**4 .Why is python is a popular programming language?**

**Python is easy to learn**

It uses a simplified syntax with an emphasis on natural language, for a much easier learning curve for beginners. And, because Python is free to use and is supported by an extremely large ecosystem of libraries and packages, it's often the first-choice language for new developers

**5 .What are the other frameworks that can be used with python?**

The frameworks provide easiness while developing an application (or software) to developers. They offer automatic implementation of redundant tasks, reduce development time, and focus significantly on application logic rather than a common element.

Python comes with many frameworks, and each framework has its advantages and disadvantages.

All we need to analyze our project requirements to choose the appropriate frameworks.

We will discuss the following Python frameworks.

Bottle

CherryPy

AIOHTTP

Dash

Falcon

Flask

Giotto

Django

Growler

Uvloop

Sanic

MorePath

Picnic

Pylons Framework

Pyramid

TurboGears

Web2py

Hug

CubicWeb

**6 .Full form of WSGI?**

The Web Server Gateway Interface (WSGI, pronounced whiskey or WIZ-ghee) is a simple calling convention for web servers to forward requests to web applications or frameworks written in the Python programming language. The current version of WSGI, version 1.0.