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

1. insert i e: Insert integer e at position *i*.
2. print: Print the list.
3. remove e: Delete the first occurrence of integer e.
4. append e: Insert integer e at the end of the list.
5. sort: Sort the list.
6. pop: Pop the last element from the list.
7. reverse: Reverse the list.

Initialize your list and read in the value of *n* followed by *n* lines of commands where each command will be of the 7 types listed above. Iterate through each command in order and perform the corresponding operation on your list.

**list=[1,3,1,5,2,4,1]**

**#To insert**

**list.insert(5,6)**

**print("The inserted list:",list)**

**#To remove first occurence**

**list.remove(1)**

**print("The first occurence removed list:",list)**

**#To insert integer at the end**

**list.append(7)**

**print("The appended list:",list)**

**#To sort the list**

**list.sort()**

**print("The sorted list:",list)**

**#To pop the last element**

**list.pop()**

**print("The popped list:",list)**

**#To reverse**

**list.reverse()**

**print("The reversed list:",list)**

**Output:**

**The inserted list: [1, 3, 1, 5, 2, 6, 4, 1]**

**The first occurence removed list: [3, 1, 5, 2, 6, 4, 1]**

**The appended list: [3, 1, 5, 2, 6, 4, 1, 7]**

**The sorted list: [1, 1, 2, 3, 4, 5, 6, 7]**

**The popped list: [1, 1, 2, 3, 4, 5, 6]**

**The reversed list: [6, 5, 4, 3, 2, 1, 1]**

2. write a calculator program in python

**def add(x, y):**

**return x + y**

**def subtract(x, y):**

**return x - y**

**def multiply(x, y):**

**return x \* y**

**def divide(x, y):**

**return x / y**

**print("Select operation.")**

**print("1.Add")**

**print("2.Subtract")**

**print("3.Multiply")**

**print("4.Divide")**

**while True:**

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

**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))**

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

**if next\_calculation == "no":**

**break**

**else:**

**print("Invalid Input")**

**Output:**

**Select operation.**

**1.Add**

**2.Subtract**

**3.Multiply**

**4.Divide**

**Enter choice(1/2/3/4): 3**

**Enter first number: 15**

**Enter second number: 14**

**15.0 \* 14.0 = 210.0**

**Let's do next calculation? (yes/no): no**

3.write a program to concatenate , reverse and slice a string in python

**str1="Hello"**

**str2="World"**

**print ("String 1:",str1)**

**print ("String 2:",str2)**

**str=str1+str2**

**print("Concatenated two different strings:",str)**

**reverse\_String = ""**

**count = len(str)**

**while count > 0:**

**reverse\_String += str[ count - 1 ]**

**count = count - 1**

**print ("The reversed string using a while loop is : ",reverse\_String)**

**sliced\_str = slice(6)**

**print(str[sliced\_str])**

**output:**

**String 1: Hello**

**String 2: World**

**Concatenated two different strings: HelloWorld**

**The reversed string using a while loop is : dlroWolleH**

**HelloW**

4. why python is popular programming language

**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?

**Types of Python Framework**

**1. Full-Stack Framework**

### 2. Microframework

### 3. Asynchronous Framework

**There are various frameworks of python like:**

* **Bottle**
* **Flask**
* **Django**
* **Web2py**
* **AIOHTTP**
* **CherryPy**
* **Dash**
* **Falcon**
* **Growler**
* **UvLoop**
* **Pyramid**
* **Sanic**
* **CubicWeb**
* **TurboGears**
* **Hug**
* **MorePath**

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.**