

Q1) Consider a list (list = []). You can perform the following commands:
insert i e: Insert integer *e* at position *i* .
print: Print the list.
remove e: Delete the first occurrence of integer *e* .
append e: Insert integer *e* 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 *N* followed by *N* 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.

SOLUTION:

```
if __name__ == '__main__':  
    N = int(input())  
    L=[];  
    for i in range(0,N):  
        cmd=input().split();  
        if cmd[0] == "insert":  
            L.insert(int(cmd[1]),int(cmd[2]))  
        elif cmd[0] == "append":  
            L.append(int(cmd[1]))  
        elif cmd[0] == "pop":  
            L.pop();  
        elif cmd[0] == "print":  
            print(L)  
        elif cmd[0] == "remove":  
            L.remove(int(cmd[1]))  
        elif cmd[0] == "sort":  
            L.sort();  
        else:
```

```
L.reverse();
```

Q2) Write a Calculator program in Python?

SOLUTION:

```
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")
```

Q3) Write a program to concatenate, reverse and slice a string?

SOLUTION:

```
def concat(x, y):  
    return x + y  
def reverse(s):  
    str = ""  
    for i in s:  
        str = i + str  
    return str  
def slicing(w, x, y):  
    num = 0  
    num = slice(x, y)
```

```
        return w[num]

print("Select operation.")

print("1.Concatenate")

print("2.Reverse")

print("3.Slice")

while True:

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

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

        if choice == '1':

            str1 = input("Enter string1: ")

            str2 = input("Enter string2: ")

            print("After concatenation", concat(str1, str2))

        elif choice == '2':

            str3 = input("Enter string1: ")

            print("After reversing", reverse(str3))

        elif choice == '3':

            str3 = input("Enter string1: ")

            # num1 = input("Enter starting index: ")

            # num2 = input("Enter stopping index: ")

            # num3 = input("Enter increment: ")
```

```
print(str3[1:3])

next_operation = input("Let's do next operation? (yes/no): ")

if next_operation == "no":

    break

else:

    print("Invalid Input")
```

Q4) Why is Python a popular programming language?

SOLUTION:

- Python is easy to learn
- Python has an active, supportive community
- Python is flexible
- Python offers versatile web-development solutions
- Python is well suited to data science and analytics
- Python is efficient, fast, and reliable
- Python is widely used with IoT Technology
- Python empowers custom automation
- Python is the academic language

Q5) What are the other Frameworks that can be used with python?

SOLUTION:

- AIOHTTP
- Bottle
- CherryPy
- CubicWeb
- Dash
- Django
- Falcon
- Giotto

Q6) Full form of WSGI?

SOLUTION:

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.