

ASSIGNMENT-12-09-22

1) 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.

Answer:

```
print("Enter Number of commands: ",end=" ")
```

```
Number = int(input())
```

```
list=[];
```

```
for i in range(0,Number):
```

```
    print("Enter the command:")
```

```
    cmd=input().split();
```

```
    if cmd[0] == "insert":
```

```
        list.insert(int(cmd[1]),int(cmd[2]))
```

```
        print("Inserted")
```

```
    elif cmd[0] == "print":
```

```
        print(list)
```

```
    elif cmd[0] == "remove":
```

```
        list.remove(int(cmd[1]))
```

```
        print("removed")
```

```
    elif cmd[0] == "append":
```

```
        list.append(int(cmd[1]))
```

```
        print("appended")
```

```
elif cmd[0] == "sort":  
    list.sort();  
    print("sorted")  
elif cmd[0] == "pop":  
    list.pop();  
    print("popped")  
else:  
    list.reverse();
```

2) Write a Calculator program in Python?

Answer:

```
print("Select Any one operation.")  
print("1.Addition")  
print("2.Subtraction")  
print("3.Multiplication")  
print("4.Division")  
print("5.Remainder(modulus)")  
while True:  
    operation = input("Enter any one option from - 1/2/3/4/5 : ")  
    if operation in ('1', '2', '3', '4', '5'):  
        num1 = float(input("Enter first number: "))  
        num2 = float(input("Enter second number: "))  
        if operation == '1':  
            print(num1, "+", num2, "=", num1+num2)  
        elif operation == '2':  
            print(num1, "-", num2, "=", num1-num2)  
        elif operation == '3':
```

```
    print(num1, "*", num2, "=", num1*num2)
elif operation == '4':
    print(num1, "/", num2, "=", num1/num2)
else:
    print(num1, "%", num2, "=", num1%num2)
next_calculation = input("Calculate again? (yes/no): ")
if ((next_calculation == "no") or (next_calculation == "NO") or (next_calculation == "No")):
    print("completed calculation")
    break
else:
    print("Invalid Input")
```

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

Answer:

```
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")
```

4) Why is Python a popular programming language?

Answer:

- Python is efficient, fast, and reliable.
- Python has ease of writing.
- Python empowers custom automation.
- Python's numerous libraries and frameworks.
- Emphasis on code readability.
- Python has shorter codes.
- Python offers versatile web-development solutions
- Python is well suited to data science and analytics.

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

Answer:

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

Q6) Full form of WSGI?

Answer:

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.