

Assignment -1
Python Programming

Assignment Date	19 September 2022
Student Name	M.Saraswathi
Student Roll Number	820419104062
Maximum Marks	2 Marks

Question-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.

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();
```

The screenshot shows a Python IDE with two windows. The left window, titled 'list1.py - E:\New folder\list1.py (3.9.10)', contains the following code:

```
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()
```

The right window, titled 'IDLE Shell 3.9.10*', shows the output of the program:

```
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\New folder\list1.py =====
30
insert 0 9
insert 1 32
insert 2 11
insert 3 7
insert 4 19
insert 5 3
print
[9, 32, 11, 7, 19, 3]
remove 32
print
[9, 11, 7, 19, 3]
append 44
print
[9, 11, 7, 19, 3, 44]
sort
print
[3, 7, 9, 11, 19, 44]
pop
print
[3, 7, 9, 11, 19]
reverse
print
[19, 11, 9, 7, 3]
```

Question-2:

Write a Calculator program in Python?

Solution:

```
def add(x, y):
    return x + y

def subtract(x, y):
    return x - y

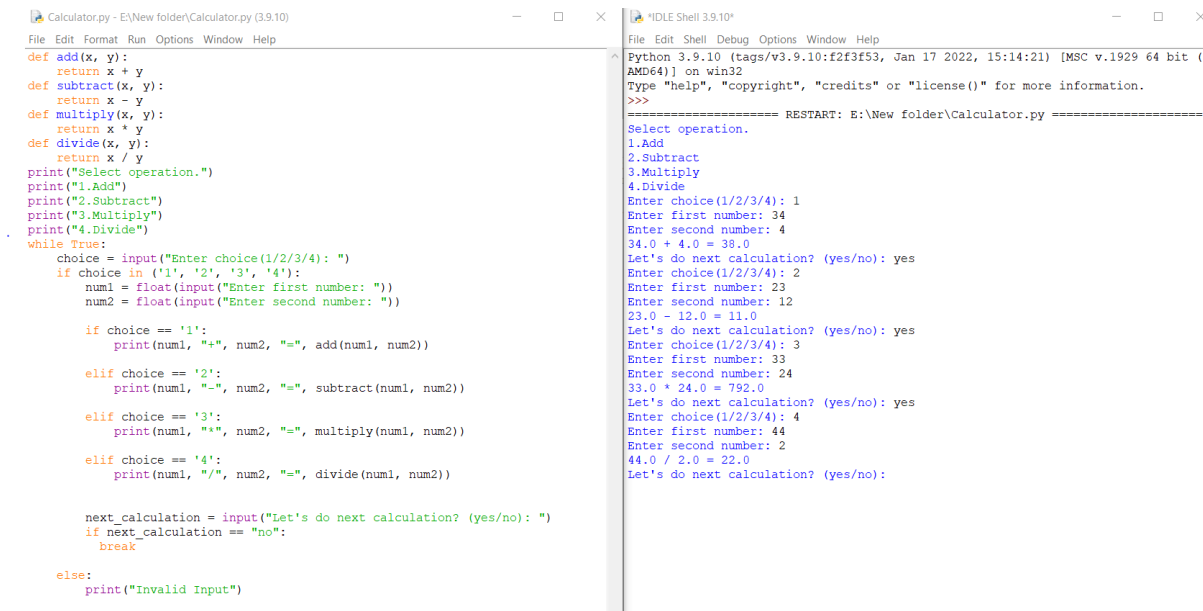
def multiply(x, y):
    return x * y

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")
```



The screenshot shows a Python IDE with two windows. The left window, titled 'Calculator.py - E:\New folder\Calculator.py (3.9.10)', contains the source code for a calculator. The code defines four functions: `add`, `subtract`, `multiply`, and `divide`. It then uses a `while True` loop to prompt the user to select an operation (1 for Add, 2 for Subtract, 3 for Multiply, 4 for Divide). For each operation, it prompts for two numbers and performs the calculation. It also asks if the user wants to do the next calculation. If the user enters anything other than 1-4, it prints 'Invalid Input' and breaks the loop. The right window, titled 'IDLE Shell 3.9.10*', shows the execution of the program. It displays the menu, the user's choices, the calculations performed (e.g., `34.0 + 4.0 = 38.0`), and the user's responses to the 'next calculation?' prompts.

```
Calculator.py - E:\New folder\Calculator.py (3.9.10)
File Edit Format Run Options Window Help
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")

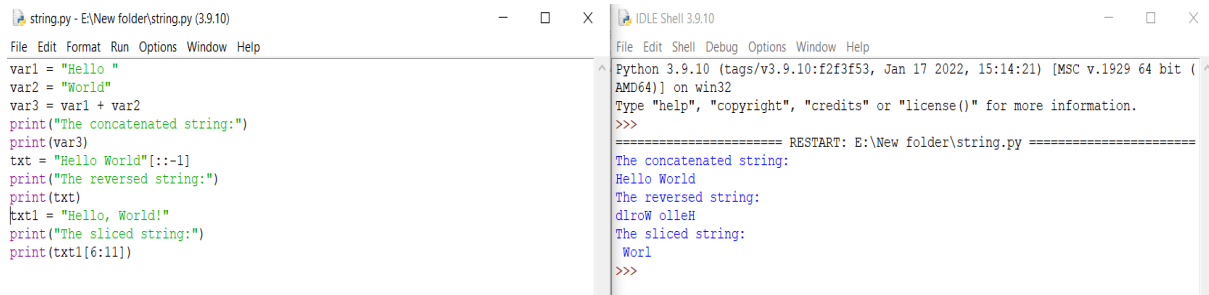
IDLE Shell 3.9.10*
File Edit Shell Debug Options Window Help
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: E:\New folder\Calculator.py =====
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 1
Enter first number: 34
Enter second number: 4
34.0 + 4.0 = 38.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 2
Enter first number: 23
Enter second number: 12
23.0 - 12.0 = 11.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 3
Enter first number: 33
Enter second number: 24
33.0 * 24.0 = 792.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 4
Enter first number: 44
Enter second number: 2
44.0 / 2.0 = 22.0
Let's do next calculation? (yes/no):
```

Question-3:

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

Solution:

```
var1 = "Hello "
var2 = "World"
var3 = var1 + var2
print("The concatenated string:")
print(var3)
txt = "Hello World"[::-1]
print("The reversed string:")
print(txt)
txt1 = "Hello, World!"
print("The sliced string:")
print(txt1[6:11])
```

A screenshot of a Python IDE with two windows. The left window, titled 'string.py - E:\New folder\string.py (3.9.10)', contains the following Python code:

```
var1 = "Hello "  
var2 = "World"  
var3 = var1 + var2  
print("The concatenated string:")  
print(var3)  
txt = "Hello World"[:-1]  
print("The reversed string:")  
print(txt)  
txt1 = "Hello, World!"  
print("The sliced string:")  
print(txt1[6:11])
```

The right window, titled 'IDLE Shell 3.9.10', shows the output of the script after execution:

```
Python 3.9.10 (tags/v3.9.10:f2f3f53, Jan 17 2022, 15:14:21) [MSC v.1929 64 bit (AMD64)] on win32  
Type "help", "copyright", "credits" or "license()" for more information.  
>>>  
===== RESTART: E:\New folder\string.py =====  
The concatenated string:  
Hello World  
The reversed string:  
dlroW olleH  
The sliced string:  
World  
>>>
```

Question-4:

Why is Python a popular programming language?

One of the main reasons it is popular as a programming language is that it's a great way to learn how to code. It reads almost like plain English and has many features that allow you to write complex tasks very simply. Python is a high-level programming language. Python is very easy to learn the language as compared to other languages like C, C#, Javascript, Java, etc. It is very easy to code in python language.

Question-5:

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

Django
Pyramid
Circuits
Flask
Cherrypy
TurboGears
Web2py

Question-6:

Full form of WSGI?

WSGI refers to Web Server Gateway Interface. WSGI plays a vital role at the time when deploy Django or Flask application.