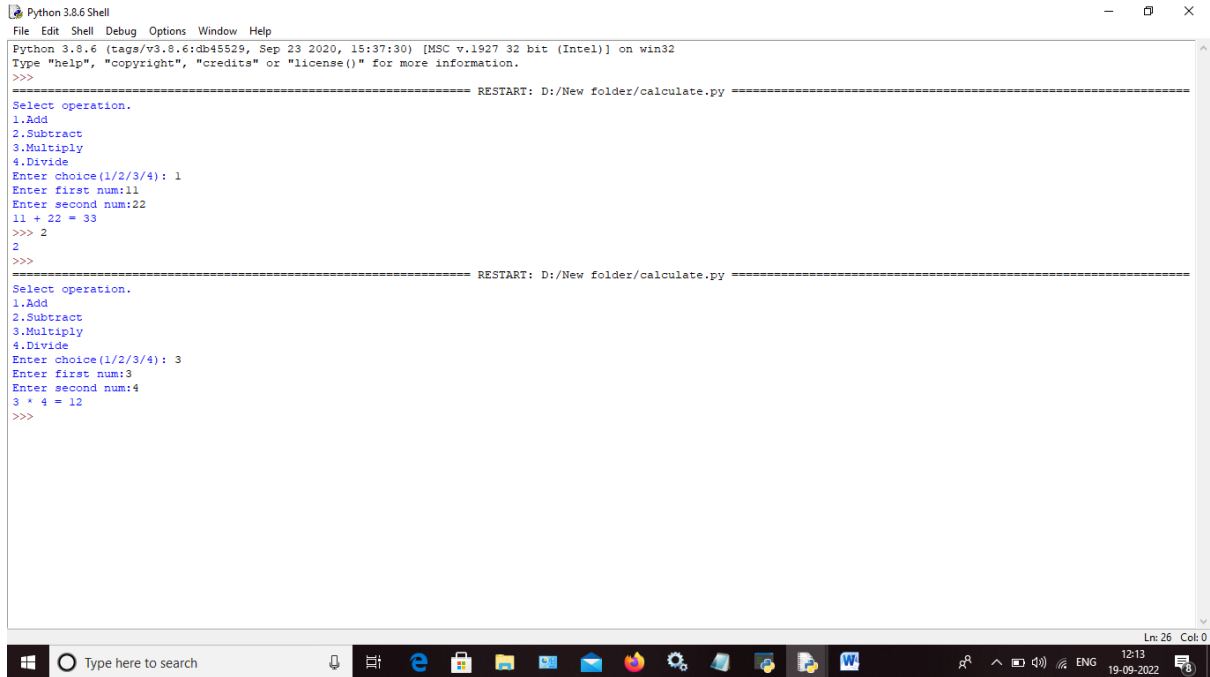


1. Write a Calculated Program

```
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")  
  
choice = input("Enter choice(1/2/3/4): ")  
  
num1=int(input("Enter first num:"))  
  
num2=int(input("Enter second num:"))  
  
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))  
  
else:  
    print("Invalid input")
```

Output



```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:37:30) [MSC v.1927 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/New folder/calculate.py =====
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 1
Enter first num:11
Enter second num:22
11 + 22 = 33
>>> 2
2
>>>
===== RESTART: D:/New folder/calculate.py =====
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 3
Enter first num:3
Enter second num:4
3 * 4 = 12
>>>
```

2.List program (append,pop,insert,sort,reverse,remove,print)

```
a=[1,2,3,4]
```

```
a.append(66)
```

```
a.pop()
```

```
a.insert(2,10)
```

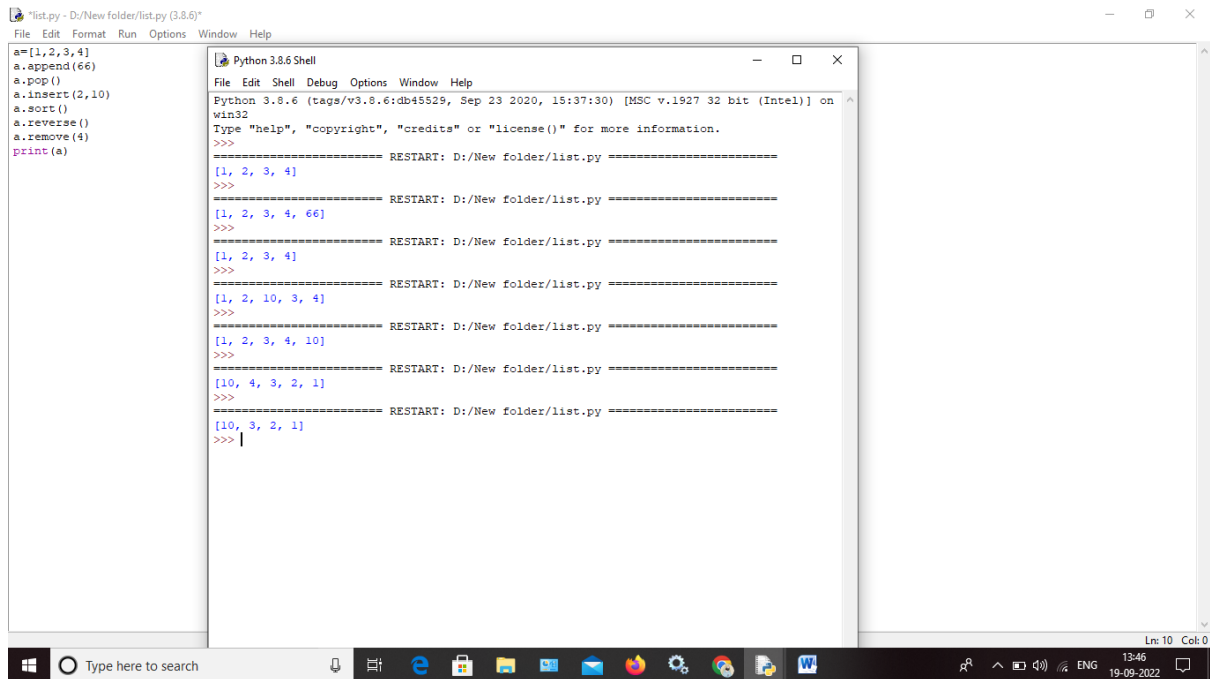
```
a.sort()
```

```
a.reverse()
```

```
a.remove(4)
```

```
print(a)
```

output:



The screenshot shows a Python IDE with a script on the left and its output in a shell window on the right. The script performs a series of list operations: initial list [1, 2, 3, 4], append 66, pop, insert 10 at index 2, sort, reverse, and remove 4. The shell window shows the script being restarted multiple times, with the output of the `print(a)` statement after each restart, demonstrating the state of the list after each operation.

```
list.py - D:/New folder/list.py (3.8.6)
File Edit Format Run Options Window Help

a=[1,2,3,4]
a.append(66)
a.pop()
a.insert(2,10)
a.sort()
a.reverse()
a.remove(4)
print(a)
```

```
Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:37:30) [MSC v.1927 32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/New folder/list.py =====
[1, 2, 3, 4]
>>>
===== RESTART: D:/New folder/list.py =====
[1, 2, 3, 4, 66]
>>>
===== RESTART: D:/New folder/list.py =====
[1, 2, 3, 4]
>>>
===== RESTART: D:/New folder/list.py =====
[1, 2, 10, 3, 4]
>>>
===== RESTART: D:/New folder/list.py =====
[1, 2, 3, 4, 10]
>>>
===== RESTART: D:/New folder/list.py =====
[10, 4, 3, 2, 1]
>>>
===== RESTART: D:/New folder/list.py =====
[10, 3, 2, 1]
>>> |
```

3.String(Concatenate,Slice,Reverse)

```
a=("Amani")
```

```
b=("ramzan")
```

```
c=a+b
```

```
print(c)
```

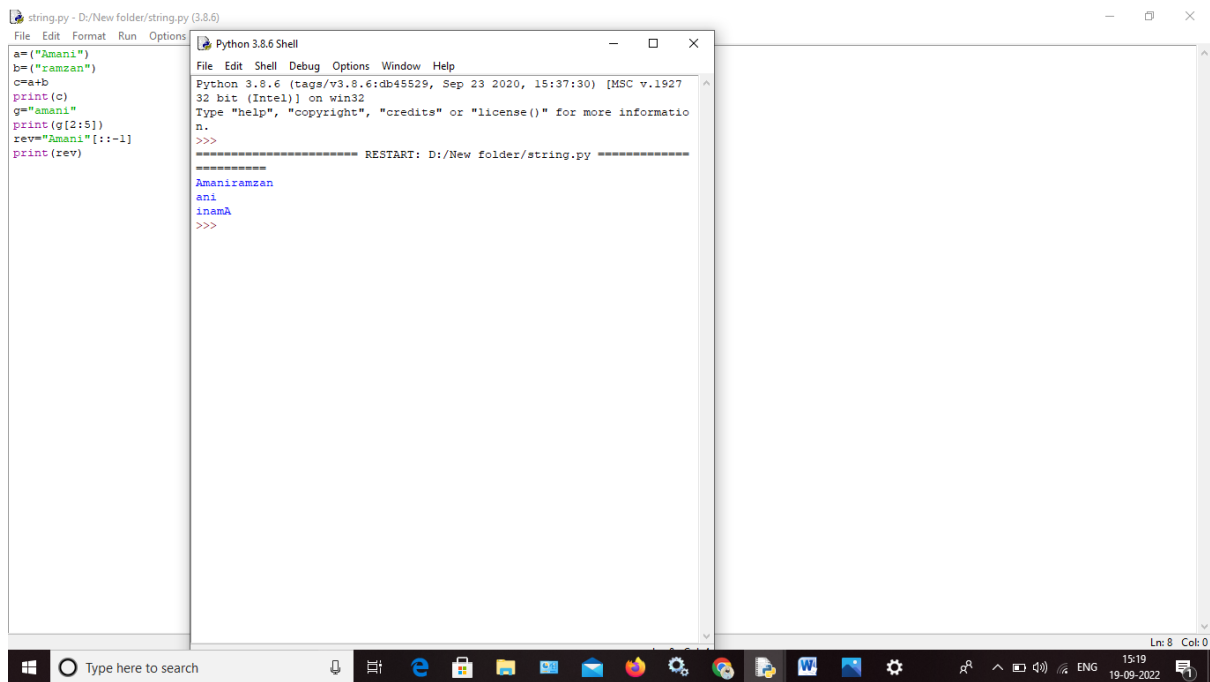
```
g="amani"
```

```
print(g[2:5])
```

```
rev="Amani"[::-1]
```

```
print(rev)
```

output:



```
string.py - D:/New folder/string.py (3.8.6)
File Edit Format Run Options
a="Amani"
b="ramzan"
c=a+b
print(c)
g="amani"
print(g[2:5])
rev="Amani"[::-1]
print(rev)

Python 3.8.6 Shell
File Edit Shell Debug Options Window Help
Python 3.8.6 (tags/v3.8.6:db45529, Sep 23 2020, 15:37:30) [MSC v.1927
32 bit (Intel)] on win32
Type "help", "copyright", "credits" or "license()" for more informatio
n.
>>>
===== RESTART: D:/New folder/string.py =====
Amaniramzan
ani
inamA
>>>
```

4.why is python a 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?

- Bottle.
- Flask.
- Django.
- Web2py.
- AIOHTTP.
- CherryPy.
- Dash.
- Falcon.

6.Full form of WSGI?

WSGI stands for "**Web Server Gateway Interface**". It is used to forward requests from a web server (such as Apache or NGINX) to a backend Python web application or framework.