Python Assignment - 1

Name	Nithin S
Roll No	SSNCE195001071
Date	15 September 2022
Team ID	PNT20222TMID53089
Project Name	Project - Personal Expense Tracker App

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.

```
N = int(input())
lists = []
for i in range(N):
  a = list(map(str,input().split( )))
  lists.append(a)
arr = []
for x in lists:
  if x[0] == "insert":
    i = int(x[1])
```

```
  e = int(x[2])
  arr.insert(i,e)
elif x[0] == "print":
  print(arr)
elif x[0] == "remove":
  e = int(x[1])
  arr.remove(e)
elif x[0] == "append":
  e = int(x[1])
  arr.append(e)
elif x[0] == "sort":
  arr.sort()
elif x[0] == "pop":
  arr.pop()
elif x[0] == "reverse":
  arr.reverse()
```

```
16
insert 0 11
print
append 9
print
remove 11
print
insert 1 8
print
pop
print
append 5
print
sort
print
reverse
print
[11]
[11, 9]
[9]
[9, 8]
[9]
[9, 5]
[5, 9]
[9, 5]
```

Write a Calculator program in Python?

```
# Program make a simple calculator
# This function adds two numbers
def add(x, y):
    return x + y


# This function subtracts two numbers
def subtract(x, y):
    return x - y


# This function multiplies two numbers
def multiply(x, y):
    return x * y


# This function divides two numbers
def divide(x, y):
    return x / y


print("Select operation.")
print("1.Add")
print("2.Subtract")
print("3.Multiply")
print("4.Divide")

while True:
    # take input from the user
    choice = input("Enter choice(1/2/3/4): ")

    # check if choice is one of the four options
    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))

        # check if user wants another calculation
        # break the while loop if answer is no
        next_calculation = input("Let's do next
calculation? (yes/no): ")
        if next_calculation == "no":
            break

    else:
        print("Invalid Input")
```

```
Select operation.
1.Add
2.Subtract
3.Multiply
4.Divide
Enter choice(1/2/3/4): 1
Enter first number: 12
Enter second number: 23
12.0 + 23.0 = 35.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 2
Enter first number: 23
Enter second number: 34
23.0 - 34.0 = -11.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 3
Enter first number: 34
Enter second number: 45
34.0 * 45.0 = 1530.0
Let's do next calculation? (yes/no): yes
Enter choice(1/2/3/4): 4
Enter first number: 45
Enter second number: 56
45.0 / 56.0 = 0.8035714285714286
Let's do next calculation? (yes/no): no
```

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

```
#Program to concatenate, reverse and slice a string
def reverse1(a):
    return a[::-1]


def concatenate(a, c):
    a+=c
    return a


def slice(a, l, h):
    return a[l:h]


print("Enter string")
b = input()
print("Select operation.")
print("1.concatrenate")
print("2.reverse")
print("3.slice")
while True:
    # take input from the user
    choice = input("Enter choice(1/2/3): ")

    # check if choice is one of the four options
    if choice in ('1', '2', '3'):

        if choice == '1':
            print("Enter character")
            temp = input()
            b = concatenate(b,temp)
            print(b)

        elif choice == '2':
            b = reverse1(b)
            print(b)

        elif choice == '3':
```

```
            print("Enter lower limit")
            l = int(input())
            print("Enter higher limit")
            h = int(input())
            b = slice(b,l,h)
            print(b)

        # check if user wants another calculation
        # break the while loop if answer is no
        next_calculation = input("continue? (yes/no): ")
        if next_calculation == "no":
            break

    else:
        print("Invalid Input")
```

```
Enter string
varsini
Select operation.
1.concatrenate
2.reverse
3.slice
Enter choice(1/2/3): 1
Enter character
s
varsinis
continue? (yes/no): yes
Enter choice(1/2/3): 2
sinisrav
continue? (yes/no): yes
Enter choice(1/2/3): 3
Enter lower limit
2
Enter higher limit
4
ni
continue? (yes/no): no
```

Why is python a popular programming language?

- easy to learn
- open source
- has a wide range of in built libraries and frameworks

- The Language is Extensively used in Data Science
- is backed up by popular companies like Google, Amazon, Facebook
- the syntax is easy to learn and remember

- it has a mature and supportive community
- Python can be used in ML tool
- The Language is Extensively used in Data Science

- Excellent Pay and Immense Career Scope
- Features like Simplicity, Library Support, Versatility, etc. have made it extensively popular.
- Python is platform-independent and highly versatile, it is used to automate different kinds of applications.

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

Python frameworks:

Django - Django is a free and open-source full-stack python framework, it includes all the necessary features by default.

It follows the DRY principle, which says don't repeat yourselves. Django uses its ORM mappers to map objects to database tables.

Bottle -

Bottle is a micro-framework which is originally meant for building APIs , bottle implements everything in a single source file. It has no dependencies whatsoever apart from the python standard library.

CherryPy -

CherryPy is an open-source framework. It follows the minimalist approach in building web applications. It makes

building web applications similar to writing an object oriented program.

Web2Py-

Web2Py is open source, scalable and a full-stack framework .

Pyramid -

Pyramid is a small, fast, down-to-earth Python web framework. It is developed as part of the Pylons Project. It is licensed under a BSD-like license. It makes real-

world web application development and deployment more fun, more predictable and more productive.

Dash -

Dash as an open source library for creating interactive web-based visualizations. The plotly team created Dash – an open source framework that leverages Flask, React.js and plotly.js to build custom data visualization apps.

CubicWeb -

Full-stack framework Developed and curated by Logilab, CubicWeb is a free-to-use, semantic, open-source, Python-based web framework. Based on the data model, CubicWeb requires to have the same defined in order to develop a functional application.

Falcon -

Microframework Aimed at rapidly building web APIs, Falcon is another widely used Python framework. Unlike other Python frameworks that require loading a lot of dependencies for building HTTP APIs, Falcon allows developers to build a cleaner design that enables HTTP and REST architectures.

Giotto -

Full-stack framework Based on the Model View Controller pattern, Giotto is an application framework for Python. In order to allow web designers, web developers, and system admins to work independently, Giotto separates Model, View, and Controller elements in order.

Growler -

Asynchronous framework Inspired by the NodeJS and the Express/Connect frameworks, Growler is a micro web framework written atop the Python's asyncio library.

Hug -

Microframework The Hug is designed to allow Python developers to develop an API once and then use it anywhere they wish. The Python framework simplifies API development by means of offering multiple interfaces. It is labeled as the fastest web framework for Python 3.

MorePath -

Microframework Labeled as the "Super Powered Python Web Framework," MorePath ensures minimal setup footprint. It is designed specifically for getting most of the typical use cases up and running ASAP, including the common Python data structures being induced into RESTful Web Services.

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