24BIT196- Shreya Nair

AIM: To implement various list operations in Python, including random number generation, searching, sorting, filtering, stack and queue operations, and list comprehensions.

HARDWARE & SOFTWARE REQUIREMENTS: Hardware:16GB RAM, Intel Processor(i9), Software: Python (Version 3.x), Google Colab (Cloud-based)

SYSTEM CONFIGURATION: Operating System: Windows 11, IDE: Google Colab

THEORY: Lists in Python are mutable sequences used to store collections of items. Various operations like sorting, searching, filtering, flattening, and data structure implementations (stack & queue) can be performed using lists.

REFERENCES: Geeks for Geeks, Python Documentation: https://docs.python.org/3/

1. Create a list of 5 odd integers using random nos. Similarly create a list of 4 even integers using random nos. Replace the third element of odd integers with a list of 4 even integers. Flattern, sort and print the list. Provide appropriate message at each stage.

```
import random
odd=[random.randrange(1,50,2) for _ in range(5)]
print("Odd numbers:",odd)
even=[random.randrange(2,50,2) for _ in range(4)]
print("Even numbers:",even)
odd[2]=even
print("After replacement:",odd)
flatten_lst=[]
for i in odd:
  if isinstance(i,list):
    flatten_lst.extend(i)
  else:
    flatten_lst.append(i)
  print("Flattened:",flatten_lst)
flatten_lst.sort()
print("Sorted:",flatten_lst)
```

```
Odd numbers: [47, 21, 9, 25, 43]
Even numbers: [30, 48, 2, 34]
After replacement: [47, 21, [30, 48, 2, 34], 25, 43]
Flattened: [47]
Flattened: [47, 21]
Flattened: [47, 21, 30, 48, 2, 34]
Flattened: [47, 21, 30, 48, 2, 34, 25]
Flattened: [47, 21, 30, 48, 2, 34, 25, 43]
Sorted: [2, 21, 25, 30, 34, 43, 47, 48]
```

2. Generate 20 random integers and store them in a list. Accept a number from the user and print position of all occurrences of that number in the list.

```
import random
lst = [random.randint(1,30) for _ in range(20)]
print("List:",lst)
num = int(input("Enter a number to search: "))
positions = [i for i in range(len(lst)) if lst[i] == num]
if positions:
    print(f"The number {num} is found at positions: {positions}")
else:
    print(f"The number {num} is not found in the list.")
```

```
List: [25, 30, 22, 23, 21, 29, 2, 12, 19, 13, 19, 21, 7, 14, 28, 21, 3, 7, 1, 2]
Enter a number to search: 20
The number 20 is not found in the list.
```

3. Generate 50 random numbers in the range 1 and 30. Remove all duplicate values from the list

```
import random
lst=[random.randint(1, 30) for _ in range(50)]
print("Original:",lst)
lst1=list(set(lst))
print("Unique:",lst1)
```

```
Original: [21, 9, 25, 14, 26, 11, 26, 7, 29, 11, 30, 9, 16, 24, 9, 14, 21, 2, 7, 21, 17, 11, 18, 9, 13, 14, 3, 4, 6, 16, 27, 13, 19, 1, 23, 20, 28, 7, 23, 4, 4, 25, 16, 10, 6
Unique: [1, 2, 3, 4, 6, 7, 8, 9, 10, 11, 13, 14, 16, 17, 18, 19, 20, 21, 23, 24, 25, 26, 27, 28, 29, 30]
```

4) Generate 30 random numbers and put them in a list. Create two more lists – one containing only +ve numbers and another with –ve nos.

```
import random
lst=[random.randint(-20,20) for _ in range(30)]
print("Original:",lst)
pt=[x for x in lst if x>0]
nt=[x for x in lst if x<0]
print("Positive:",pt)
print("Negative:",nt)
```

```
Original: [0, 20, 5, -8, 17, -4, -17, -1, 16, -15, 4, 4, 0, -20, 20, 9, 16, -12, 1, 5, -15, 1, -15, 5, 0, 20, -15, -14, 19, 18]
Positive: [20, 5, 17, 16, 4, 4, 20, 9, 16, 1, 5, 1, 5, 20, 19, 18]
Negative: [-8, -4, -17, -1, -15, -20, -12, -15, -15, -15, -14]
```

5) A list contains 5 strings. Convert all these strings to uppercase.

```
lst=["monday","tuesday","wednesday","thursday","friday"]
print("Original:",lst)
upper_lst=[x.upper() for x in lst]
print("Uppercase list:", upper_lst)
```

```
Original: ['monday', 'tuesday', 'wednesday', 'thursday', 'friday']
Uppercase list: ['MONDAY', 'TUESDAY', 'WEDNESDAY', 'THURSDAY', 'FRIDAY']
```

6) Convert list of temperatures in Fahrenheit degrees to equivalent Celsius degrees.

```
lst1=[30,40,50,70,90,100]
print("Temperature in fahrenheit:",lst1)
lst2=[(x-32)*5/9 for x in lst1]
print("Temperature in celsius:",lst2)
```

```
Temperature in fahrenheit: [30, 40, 50, 70, 90, 100]
Temperature in celsius: [-1.1111111111111112, 4.444444444444445, 10.0, 21.11111111111111, 32.22222222222222, 37.77777777777778]
```

7) Write a menu-driven program to implement the stack data structure.

```
stck = []
while True:
    print("Operations:")
    print("1. Push")
    print("2. Pop")
    print("3. Display")
    print("4. Exit")
    choice = int(input("Enter choice: "))
    if choice == 1:
        element = input("Enter element to push: ")
        stck.append(element)
        print("Element pushed.")
    elif choice == 2:
        if stck:
            print("Popped element:", stck.pop())
        else:
            print("Stack is empty!")
    elif choice == 3:
        print("Stack contents:", stck)
    elif choice == 4:
        break
    else:
        print("Invalid")
```

```
Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 1
Enter element to push: 2
Element pushed.
Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 2
Popped element: 2
Operations:
1. Push
```

```
2. Pop
3. Display
4. Exit
Enter choice: 3
Stack contents: []
Operations:
1. Push
2. Pop
3. Display
4. Exit
Enter choice: 4
```

8) Write a menu-driven program to implement the Queue data structure.

```
q=[]
while True:
  print("Operations:")
  print("1. Add element:")
  print("2. Remove element:")
  print("3. Display")
  print("4. Exit")
  choice=int(input("Enter:"))
  if choice==1:
   elem=input("Enter:")
   q.append(elem)
   print(elem)
  elif choice==2:
   if q:
    print(q.pop(0))
   else:
    print("Empty")
  elif choice==3:
   if q:
    print(q)
   else:
    print("empty")
  elif choice==4:
   print("Exit")
   break
  else:
   print("Invalid")
   break
```

```
Operations:
1. Add element:
2. Remove element:
3. Display
4. Exit
Enter:5
Invalid
```

9) Take two lists of numbers. Create third list of numbers for only those numbers from first list which are not there in 2nd list (use list comprehension).

```
lst1=[3,5,7,9,2,6]
lst2=[4,5,6,7,8]

lst3=[x for x in lst1 if x not in lst2]

print(lst3)
```

```
[3, 9, 2]
```