



Tool & Technique Laboratory (T&T Lab.)

[CS-3096]

Individual Work

Lab. No:4 , Date: 7/02/2023 , Day: 4

Topic: Python basics

| | | | |
|------------------|-----------------|-----------------|--------|
| Roll Number: | 20051939 | Branch/Section: | CSE-17 |
| Name in Capital: | Shashikant shah | | |

Program No: (4.1)

Original Program:

Python program to interchange first and second elements in a list.

Modified Program Title:

Python program to interchange first and last elements in a list.

Input/Output Screenshots:

RUN-1:

```
Original list--> [6, 7, 9, 11, 22]
[22, 7, 9, 11, 6]
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques Laboratory\lab 4\lab 4>
```

RUN-2

```
Laboratory\lab 4\lab 4\q1.py
Original list--> [12, 35, 26, 17, 28]
[28, 35, 26, 17, 12]
PS C:\Users\KIIT\Desktop\6th sem\Tools and Technique
```

Source code

```
def swapList(newList):
    size = len(newList)
    temp = newList[0]
    newList[0] = newList[size - 1]
    newList[size - 1] = temp

    return newList

newList = [12, 35, 9, 56, 24]
print("Original list-->",newList)
print(swapList(newList))
```

Conclusion/Observation

We have successfully interchanged first and last number.

Program No: (4.2)

Original Program:

Sum of all numbers in the list.

Modified Program Title:

Multiply all numbers in the list

Input/Output Screenshots:

RUN-1:

```

C:\Users\KIIT\Desktop\ben
Laboratory\lab 4\lab 4\q2.py
98
7986
PS C:\Users\KIIT\Desktop\6th
Ln 8, Col 18  Tab Size: 4  UTF-

```

RUN-2

```

Laboratory\lab 4\lab 4\q2.py
216
7986
PS C:\Users\KIIT\Desktop\6th
Ln 9, Col 21  Tab Size: 4  UTF-

```

Source code

```
def multiplyList(myList):
```

```

    result = 1
    for x in myList:
        result = result * x
    return result

```

```

list1 = [9, 2, 5]
list2 = [11, 22, 33]
print(multiplyList(list1))
print(multiplyList(list2))

```

Conclusion/Observation

We have successfully calculated product of all the numbers in the set.

Program No: (4.3)**Original Program:**

program to count Even numbers in a List.

Modified Program Title:

program to count Even and Odd numbers in a List

Input/Output Screenshots:**RUN-1:**

```
Laboratory\lab 4\lab 4\q3.py"
Even numbers in the list: 5
Odd numbers in the list: 2
PS C:\Users\KIIT\Desktop\6th sem\Tools an
Ln 10, Col 10, Tab Size: 4, UTF-8, CR LF, {}
```

RUN-2

```
Laboratory\lab 4\lab 4\q3.py"
Even numbers in the list: 5
Odd numbers in the list: 5
PS C:\Users\KIIT\Desktop\6th sem\Too
```

Source code

```
list1 = [9, 8, 4, 11, 6, 56, 2]
```

```
even_count, odd_count = 0, 0
```

```
for num in list1:
```

```
    if num % 2 == 0:
        even_count += 1
```

```
    else:
        odd_count += 1
```

```
print("Even numbers in the list: ", even_count)
```

```
print("Odd numbers in the list: ", odd_count)
```

Conclusion/Observation

We have successfully calculated number of odd and even numbers

Program No: (4.4)

Original Program:

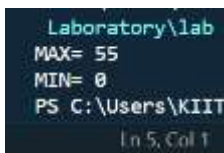
Maximum in a Set.

Modified Program Title:

Maximum and Minimum in a Set.

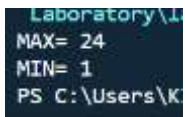
Input/Output Screenshots:

RUN-1:



```
Laboratory\lab
MAX= 55
MIN= 0
PS C:\Users\KIIT
Ln 5, Col 1
```

RUN-2



```
Laboratory\lab
MAX= 24
MIN= 1
PS C:\Users\KIIT
```

Source code

```
sets = set([8, 16, 24, 1, 25, 3, 10, 65, 55])
print("MAX=",max(sets))
print("MIN=",min(sets))
```

Conclusion/Observation

We have successfully printed max and min number in a set.

Program No: (4.5)**Original Program:**

Program to accept a vowel.

Modified Program Title:

Program to accept the strings which contains all vowels

Input/Output Screenshots:**RUN-1:**

```

PS C:\Users\KIIT\Desktop\6th sem\Tools ar
Laboratory\lab 4\lab 4\q5.py"
Enter string: aeiou
Accepted
PS C:\Users\KIIT\Desktop\6th sem\Tools ar
Ln 10, Col 1 Tab Size: 4 UTF-8 CRLF { } P

```

RUN-2

```

Laboratory\lab 4\lab 4\q5.py"
Enter string: shashikant
Not Accepted
PS C:\Users\KIIT\Desktop\6th sem\Tools ar
Ln 12, Col 24 Tab Size: 4 UTF-8 CRLF { } P

```

Source code

```

def check(string):
    string = string.lower()

    vowels = set("aeiou")

    s = set({})

    for char in string:
        if char in vowels:
            s.add(char)
        else:
            pass

    if len(s) == len(vowels):
        print("Accepted")
    else:
        print("Not Accepted")

if __name__ == "__main__":
    string = str(input("Enter string: "))
    check(string)

```

Conclusion/Observation

Successfully accepted a string of all vowels

Program No: (4.6)

Original Program:

Check if a given string is containing any 0 or 1.

Modified Program Title:

Check if a given string is binary string or not

Input/Output Screenshots:

RUN-1:

```
PS C:\Users\KIIT\Desktop\6th sem\Tools
Laboratory\lab 4\lab 4\q6.py"
Enter your string: shashikant
No
PS C:\Users\KIIT\Desktop\6th sem\Tools
Ln 8, Col 43  Tab Size: 4  UTF-8  CRLF
```

RUN-2

```
Laboratory\lab 4\lab 4\q6.py"
Enter your string: 0123
No
PS C:\Users\KIIT\Desktop\6th sem\Tools a
Laboratory\lab 4\lab 4\q6.py"
Enter your string: 1010101
Yes
PS C:\Users\KIIT\Desktop\6th sem\Tools a
Ln 7, Col 1  Tab Size: 4  UTF-8  CRLF
```

Source code

```
def check(string):

    p = set(string)

    s = {'0', '1'}

    if s == p or p == {'0'} or p == {'1'}:
        print("Yes")
    else:
        print("No")

if __name__ == "__main__":

    string = str(input("Enter your string: "))

    check(string)
```

Conclusion/Observation

We have successfully checked if the string is binary or not.

Program No: (4.7)**Original Program:**

program to count number of alphabets using sets in a given string.

Modified Program Title:

program to count number of vowels using sets in given string

Input/Output Screenshots:**RUN-1:**

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
Laboratory\lab 4\lab 4\q7.py"
Enter string:shashikant
No. of vowels : 3
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
are Ln 5, Col 5 Tab Size: 4 UTF-8 CRLF Python 3.10
```

RUN-2

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
Laboratory\lab 4\lab 4\q7.py"
Enter string:aerio
No. of vowels : 5
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techni
Ln 7, Col 5 Tab Size: 4 UTF-8 CRLF Python 3.10.9
```

Source code

```
def vowel_count(str):
    count = 0
    vowel = set("aeiouAEIOU")
    for alphabet in str:
        if alphabet in vowel:
            count = count + 1
    print("No. of vowels :", count)
string = str(input("Enter string:"))
vowel_count(string)
```

Conclusion/Observation

We have successfully counted total vowels

Program No: (4.8)

Original Program:

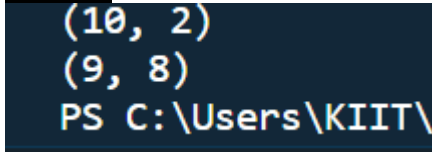
Swap two numbers in a tuple.

Modified Program Title:

Swap two tuples in Python

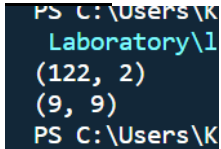
Input/Output Screenshots:

RUN-1:



```
(10, 2)
(9, 8)
PS C:\Users\KIIT\
```

RUN-2:



```
PS C:\Users\K
Laboratory\1
(122, 2)
(9, 9)
PS C:\Users\K
```

Source code

```
tuple1 = (10, 2)
tuple2 = (9, 8)
tuple1, tuple2 = tuple2, tuple1
print(tuple2)
print(tuple1)
```

Conclusion/Observation

We have successfully swapped two tuples.

Program No: (4.9)

Original Program:

Print the tuple

Modified Program Title:

Reverse the tuple

Input/Output Screenshots:

RUN-1:

```
PS C:\Users\KIIT\Desktop\6th sem\Tools
Laboratory\lab 4\lab 4\q9.py"
(5, 4, 3, 2, 1)
PS C:\Users\KIIT\Desktop\6th sem\Tools
```

RUN-2:

```
PS C:\Users\KIIT\Desktop\
Laboratory\lab 4\lab 4\q
(7, 6, 5, 2, 1)
PS C:\Users\KIIT\Desktop\
In 4, Col 1: Spaces: 4
```

Source code

```
tuple1 = (1, 2, 3, 4, 5)
tuple1 = tuple1[::-1]
print(tuple1)
```

Conclusion/Observation

We have successfully reversed the tuple

Program No: (4.10)

Original Program:

Sort the tuple

Modified Program Title:

Sort a tuple of tuples by 2nd item

Input/Output Screenshots:

RUN-1:

```
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques
Laboratory\lab 4\lab 4\q10.py"
(('c', 1), ('a', 2), ('b', 7), ('d', 9))
PS C:\Users\KIIT\Desktop\6th sem\Tools and Techniques
Ln 4, Col 1  Spaces: 4  UTF-8  CRLF  { } Python  3.10.9
```

RUN-2:

```
In [30]: runfile('C:/Users/Aryan Raj/Desktop/t&tLab/lab4/untitled10.py', wdir='C:/Users/Aryan Raj/
Desktop/t&tLab/lab4')
(('c', 1), ('a', 2), ('b', 7), ('d', 9))
```

Source code

```
tuple1 = (('a', 2), ('b', 7), ('c', 1), ('d', 9))
tuple1 = tuple(sorted(list(tuple1), key=lambda x: x[1]))
print(tuple1)
```

Conclusion/Observation

We have sorted in the required order.

Program No: (4.11)

Original Program:

Print two Dictionaries in sorted order

Modified Program Title:

Merging two Dictionaries

Input/Output Screenshots:

RUN-1:

```
In [31]: runfile('C:/Users/Aryan Raj/Desktop/t&tlab/lab4/untitled11.py', wdir='C:/Users/Aryan Raj/Desktop/t&tlab/lab4')
None
After merging= {'d': 6, 'c': 4, 'a': 10, 'b': 8}
```

RUN-2:

```
In [32]: runfile('C:/Users/Aryan Raj/Desktop/t&tlab/lab4/untitled11.py', wdir='C:/Users/Aryan Raj/Desktop/t&tlab/lab4')
None
After merging= {'d': 3, 'c': 4, 'a': 1, 'b': 88}
```

Source code

```
def Merge(dict1, dict2):
    return(dict2.update(dict1))
```

```
dict1 = {'a': 1, 'b': 88}
dict2 = {'d': 3, 'c': 4}
```

```
print(Merge(dict1, dict2))
```

```
print("After merging=" ,dict2)
```

Conclusion/Observation

We have merged two dictionaries.

Program No: (4.12)

Original Program:

Print all keys

Modified Program Title:

Key with maximum unique values

Input/Output Screenshots:

RUN-1:

```
In [33]: runfile('C:/Users/Aryan Raj/Desktop/t&tlab/Lab4/untitled12.py', wdir='C:/Users/Aryan Raj/Desktop/t&tlab/Lab4')
The original dictionary is : {'Gfg': [5, 7, 5, 4, 5], 'is': [6, 7, 4, 3, 3], 'Best': [9, 9, 6, 5, 5]}
Key with maximum unique values : is
```

Source code

```
test_dict = {"Gfg" : [5, 7, 5, 4, 5],
             "is" : [6, 7, 4, 3, 3],
             "Best" : [9, 9, 6, 5, 5]}

print("The original dictionary is : " + str(test_dict))

max_val = 0
max_key = None
for sub in test_dict:
    if len(set(test_dict[sub])) > max_val:
        max_val = len(set(test_dict[sub]))
        max_key = sub

print("Key with maximum unique values : " + str(max_key))
```

Conclusion/Observation

We have printed keys with max unique values.

Program No: (4.13)

Original Program:

Remove words from Dictionary

Modified Program Title:

Replace words from Dictionary

Input/Output Screenshots:

RUN-1:

```
In [35]: runfile('C:/Users/Aryan Raj/Desktop/t&tlab/lab4/untitled13.py', wdir='C:/Users/Aryan Raj/Desktop/t&tlab/lab4')
The original string is : Hello how are you i am fine
Replaced Strings : Hello how replaced1 you i am replaced2
```

Source code

```
test_str = 'Hello how are you i am fine'

print("The original string is : " + str(test_str))

lookp_dict = {"are" : "replaced1", "fine" : "replaced2"}

temp = test_str.split()
res = []
for wrd in temp:
    res.append(lookp_dict.get(wrd, wrd))
res = ' '.join(res)

print("Replaced Strings : " + str(res))
```

Conclusion/Observation

We have replaced values from dictionary

