FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERIG
Department of Computer Engineering

Experiment 2 - Python Programs on Tuples, Dictionary and sets

1. Course Details:

Academic Year	2023 - 24	Fst mated Time	Experiment No. 2 – 02 Hours
Course & Semester	S.E. (COMP) – Sem. IV	Subject Name	Python Programming Lab
Module No.	01	Chapter Title	Python Basics
Experiment Type	Software Performance	Subject Code	CSL405

Name of Student	Prakash P. Biswas	**Roll No.**	9947
Date of Performance.:	02/02/2024	**Date of Submission. :**	10/02/2024
CO Mapping	**CSL405.1: Demonstrate basic concepts of python such as control statements, basic data structures, functions and oops in python. (Comprehension)**		

Timeline (2)	Preparedness (2)	Effort (2)	Result (2)	Documentation (2)	Total (10)

2. Aim & Objective of Experiment

To implement following programs in Python.

Objective of experiment 2 is to understand the basic concepts of Python Programming. Students will be able to demonstrate how to create Tuple, Sets and dictionary using Python language. Students will be able to apply various operations like slicing on strings and list, control statements, various functions like append, sort, remove, delete etc. on Tuple, Sets and dictionary.

Pre-Requisite: Any programming language like C, C++
Tools: Python IDE

Python Lab 2(List, Tuples, Dictionary and sets)

1. Given two digits of integer x find the digits of the number.(Any Two methods)

2. Given an arbitrary integer count the number of digits. (without using any loops)

3. Take a list of numbers. Make two list of square of odd and even numbers.

4. Given a list week containing names of the days of the week and a strings check if the string is a day of the week. We should be able to check for any of the forms sat, Sat, SAT.

5. Declare a tuple x= (1,2,3). Change the values of the tuple to x=(1,2,'apple','orange')

6. Swap two integer values a=67, b=77, without using a third variable.

7. Make a python dictionary with key names as 'txt', 'png','py' and value names as 'text file', 'image file', 'python file' etc.

e. g. mydict={'Jan': 1….}

update the dictionary with key as 'doc' and value as 'Word File'.

8. Define two sets in python and find the UNION, INTERSECTION,DIFFERENCE and SYMMETRIC DIFFERENCE between the two.

9. Input a date string like indate = '6 Feb, 2008' or '8 Jul 2020' comma may or may not be present. Make a tuple from the string as e.g. tupdate = (2008, 2, 6)

10. One of the following 10 statements generates an error. Which one? And Why?

```
x = [1,"abcd", 2,"efgh", [3, 4]] # Statement 1
y = x [0:50] # Statement 2
z = y # Statement 3
w = x # Statement 4
x[1] = x[1] + 'd' # Statement 5
y[2] = 4 # Statement 6
x[1][1] = 'y' # Statement 7
 z[0] = 0 # Statement 8
w[4][0] = 1000 # Statement 9
a = (x[4][1] == 4) # Statement 10
```

Dictionaries Challenge 1: Thesaurus App

Description:

Write a program that simulates a thesaurus. Your program will present a user with a list of words that your thesaurus contains. Based on the users choice, you will randomly present them with a synonym for their chosen word. Lastly, your program will display all of the potential synonyms for each word in the thesaurus.

Step By Step Guide:

- Create a dictionary called "thesaurus"
  - You must have a minimum of four keys in the dictionary.
  - Each key should be a **string** of a word of your choice.
  - The associated values for each key should be a **list** containing five synonyms for the key.
- For example, my dictionary includes:

"hot":['balmy', 'summery', 'tropical', 'boiling', 'scorching'],
"cold":['chilly', 'cool', 'freezing', 'frigid', 'polar'],
"happy":['content', 'cheery', 'merry', 'jovial', 'jocular'],

"sad":['unhappy', 'downcast', 'miserable', 'glum', 'melancholy'],
- Print a welcome message.
- Print what words are in your thesaurus.
    - This is represented by the keys of your dictionary
- Ask the user what word they would like to get a synonym for.
- If the users choice is in the thesaurus:
    - Chose a random synonym from the list containing the synonyms for the word.
        - Type import random as the first line of your program.
    - Display the information.
- Else, the word is not in the dictionary:
    - Inform the user.
- Ask the user if they would like to see the whole thesaurus.
    - This is represented by both the keys and values.
- If yes:
    - Display the whole dictionary following the format below.
- Else:
    - Print a goodbye message
- Use at least 2 comments to describe sections of your code.
- "Chunk" your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below.

Which letter would you like to count the occurrences of: h
Guido, your message has 7 h's in it.

Post Lab:

Dictionaries Challenge 2: Database Admin Program

Description:

Write a program that will simulate logging into a database and prompting a user to change their password. All usernames and passwords to the database will be stored in a dictionary. Upon entering the correct credentials, your program will prompt the user to enter a new password that is a minimum of eight characters long. If the new password meets the criteria, it will be accepted, otherwise the new password will be rejected. If the user who logged in is the admin, a list of all usernames and passwords will be displayed.

Step By Step Guide:

- Print a welcome message.
- Create a dictionary called "log_on_information".
    - The key-value pairs in this dictionary will be username:password.
    - Store 5 keys.
        - Each key should be a string representing a username.
        - One username must be "admin00".
    - Store 5 values.
        - Each value should be a string representing a password at least 8

characters long.

- Get user input for their username.
- Follow the following conditional logic to control user interaction.
- If the username is in the database, get user input for their password.
    - If they enter the password correct, greet them with a message.
        - If the user that logged in is the admin00, display the whole dictionary.
        - Else, if the user who logged in is not the admin, ask the user if they would like to change their password.
- If the user wants to change their password, ask them for their new password and inform them that the password must be 8 characters long.
    - If the password is not 8 characters long, do not accept.
    - Else, If the password is 8 characters long or more, accept and display the username and new password.
- Else, if the user does not want to change their password, print a goodbye message.
    - Else, if the user enters their password incorrectly, inform the user.
- Else, If the username is not in the database, inform the user.
- Use at least 2 comments to describe sections of your code.
- "Chunk" your code so that is readable.
- Use appropriate and informative variable names.
- Format your output as below

In [1]:
```
x=15

digit1=(x//10)
digit2=x%10

print( digit1)
print(digit2)
```

```
1
5
```

In [2]:
```
x=25

digit_str=str(x)
digit1=int(digit_str[0])
digit2=int(digit_str[1])

print( digit1)
print(digit2)
```

```
2
5
```

In [3]:
```
x=98643

num_digits=len(str(abs(x)))

print("Number of digits:", num_digits)
```

```
Number of digits: 5
```

In [11]:
```
l= [1, 2, 3, 4, 5, 6, 7, 8]

l1= []
l2= []

foriinl:
    ifi%2==0:
        l1.append(i*i)
    else:
        l2.append(i*i)

print("Even sq", l1)
print("Odd sq", l2)
```

```
Even sq [4, 16, 36, 64]
Odd sq [1, 9, 25, 49]
```

In [19]:
```
week= ["mon", "tue", "wed", "thu", "fri", "sat", "sun"]

defis_day(str):

    lower=str.lower()
    week_lower= [day.lower() fordayinweek]


    returnlowerinweek_lower


input="SAt"
result=is_day(input)

ifresult:
    print(f"{input_day} is a day of the week.")
else:
    print(f"{input_day} is not a valid day of the week.")
```

```
SAt is a day of the week.
```

In [20]:
```
x= (1, 2, 3)
x= (x[0], x[1], 'apple', 'orange')

print(x)
```

```
(1, 2, 'apple', 'orange')
```

In [21]:
```
a=67
b=77

a=a+b
b=a-b
a=a-b

print("After swapping: a =", a, "b =", b)
```

```
After swapping: a = 77 b = 67
```

In [24]:
```
file_types= {
    'txt': 'Text File',
    'png': 'Image File',
    'py': 'Python File'
}
print("Original Dictionary:", file_types)

file_types['doc'] ='Word File'

print(file_types)
```

```
Original Dictionary: {'txt': 'Text File', 'png': 'Image File', 'py
': 'Python File'}
{'txt': 'Text File', 'png': 'Image File', 'py': 'Python File', 'doc
': 'Word File'}
```

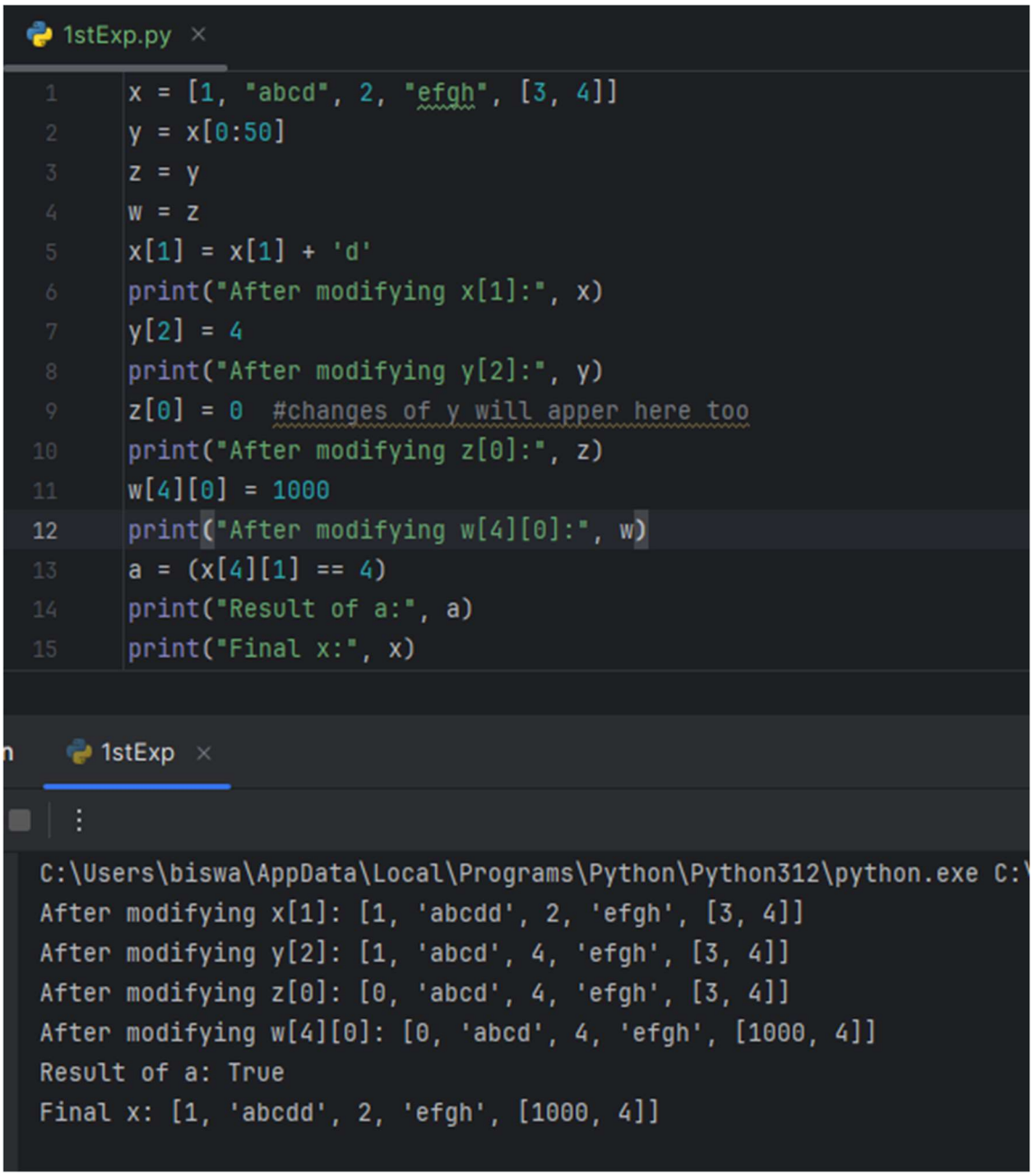
In [26]:
```
set1= {1, 2, 3, 4, 5}
set2= {4, 5, 6, 7, 8}

union_set=set1.union(set2)
print("Union:", union_set)


intersection_set=set1.intersection(set2)
print("Intersection:", intersection_set)

difference_set=set1.difference(set2)
print("Difference:", difference_set)

symmetric_difference_set=set1.symmetric_difference(set2)
print("Symmetric Difference:", symmetric_difference_set)
```

```
Union: {1, 2, 3, 4, 5, 6, 7, 8}
Intersection: {4, 5}
Difference: {1, 2, 3}
Symmetric Difference: {1, 2, 3, 6, 7, 8}
```

In [38]:
```
import datetime
def make_tuple(indate):
    indate = indate.replace(",", "").split()
    month_abbreviations = ['Jan', 'Feb',
'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug',
'Sep', 'Oct', 'Nov', 'Dec']
    try:
        month_number =
month_abbreviations.index(indate[1][:3]) + 1
        return tuple(int(part) for part in
[indate[2], month_number, indate[0]])
    except ValueError:
        raise ValueError(f"Invalid month
abbreviation: {indate[1][:3]}")
indate = '6 Feb, 2008'
try:
    tupdate = make_tuple(indate)
    print(tupdate)
except ValueError as e:
```

```
(2008, 2, 6)
```

*


1stExp.py
x = [1, "abcd", 2, "efgh", [3, 4]]
y = x[0:50]
z = y
w = z
x[1] = x[1] + 'd'
print("After modifying x[1]:", x)
y[2] = 4
print("After modifying y[2]:", y)
z[0] = 0 #changes of y will apper here too
print("After modifying z[0]:", z)
w[4][0] = 1000
print("After modifying w[4][0]:", w)
a = (x[4][1] == 4)
print("Result of a:", a)
print("Final x:", x)
1stExp
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C:
After modifying x[1]: [1, 'abcdd', 2, 'efgh', [3, 4]]
After modifying y[2]: [1, 'abcd', 4, 'efgh', [3, 4]]
After modifying z[0]: [0, 'abcd', 4, 'efgh', [3, 4]]
After modifying w[4][0]: [0, 'abcd', 4, 'efgh', [1000, 4]]
Result of a: True
Final x: [1, 'abcdd', 2, 'efgh', [1000, 4]]

*

```python
import random
thesaurus = {
    "hot": ['balmy', 'summery', 'tropical', 'boiling', 'scorching'],
    "cold": ['chilly', 'cool', 'freezing', 'frigid', 'polar'],
    "happy": ['content', 'cheery', 'merry', 'jovial', 'jocular'],
    "sad": ['unhappy', 'downcast', 'miserable', 'glum', 'melancholy']
}
print("Welcome to the Thesaurus App!")
print("Words in the Thesaurus:")
for word in thesaurus.keys():
    print("-", word)      # To Display words in the thesaurus

user_word = input("Enter a word to get a synonym: ").lower()    # User se word liye

if user_word in thesaurus:       # Check the word is in the thesaurus
    random_synonym = random.choice(thesaurus[user_word])    #koi bhi synonym choice karega
    print(f"A synonym for '{user_word}' is: {random_synonym}")
else:
    print(f"Sorry, '{user_word}' is not in the thesaurus.")

see_whole_thesaurus = input("Would you like to see the whole thesaurus? (yes/no): ").lower()

if see_whole_thesaurus == 'yes':
    print("\nWhole Thesaurus:")
    for word, synonyms in thesaurus.items():
        print(f"{word.capitalize()}: {', '.join(synonyms)}")

print("Thank you for using the Thesaurus App! Goodbye.")
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe
Welcome to the Thesaurus App!
Words in the Thesaurus:
- hot
- cold
- happy
- sad
Enter a word to get a synonym: sad
A synonym for 'sad' is: downcast
Would you like to see the whole thesaurus? (yes/no): no
Thank you for using the Thesaurus App! Goodbye.

Process finished with exit code 0
```

*

```
print("Welcome to the Database Admin Program!")
log_on_information = {
    "Prakash": "password1","Akash": "securepass","Pronab": "abc12345", "Kartick":
"adminpass","admin00": "pass"
}
username = input("Enter your username: ")
if username in log_on_information:    # Check username is in the database
    password = input("Enter your password: ")

    if password == log_on_information[username]:      # Check password
        print(f"Hello {username}! You have successfully logged in.")

        if username == "admin00":            #Display the entire dictionary if true
            print("Here is the list of all usernames and passwords:")
            for user, passw in log_on_information.items():
                print(f"Username: {user}, Password: {passw}")
        else:
                        # Ask to change their password
            change_password = input("Would you like to change your password? (yes/no): ").lower()
            if change_password == "yes":
                new_password = input("Enter your new password (at least 8 characters): ")
                if len(new_password) >= 8:
                    log_on_information[username] = new_password
                    print(f"Password successfully changed! Updated login information:")
                    print(f"Username: {username}, New Password: {new_password}")
                else:
                    print("Password must be at least 8 characters long. Change unsuccessful.")
            else:
                print("Goodbye!")
    else:
        print("Incorrect password.")
else:
    print("Username not found")
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C
Welcome to the Database Admin Program!
Enter your username: Prakash
Enter your password: password1
Hello Prakash! You have successfully logged in.
Would you like to change your password? (yes/no): yes
Enter your new password (at least 8 characters): python123
Password successfully changed! Updated login information:
Username: Prakash, New Password: python123
```