FR. CONCEICAO RODRIGUES COLLEGE OF ENGINEERIG
Department of Computer Engineering

Experiment 3 - Python Programs on Control statements and functions

1. Course Details:

Academic Year	2023 - 24	Estimated Time	Experiment No. 3 – 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.:	09-02-2024	**Date of Submission. :**	27-02 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 3 is to understand the basic concepts of Python Programming. Students will be able to demonstrate how to create Tuple, Sets , dictionary, functions using Python language. Students will be able to apply various operations in functions using control structure .

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

Python Lab 3 (Functions: Strings, List, Tuples, Dictionaries, While loop, for loop, if else)
1. Write a function named same that takes a string as input, and simply returns that string.

2. Write a function called subtract three that takes an integer or any number as input and returns that number minus three.

3. Write a function named intro that takes a string as input. Given the string "**Becky**" as input, the function should return: "Hello, my name is Becky and I love SI 106."

4. Write a function named total that takes a list of integers as input and returns the total value of all those integers added together.

5. Write a while loop that is initialized at 0 and stops at 15. If the counter is an even number, append the counter to a list called eve_nums.

6. Below is a for loop that works. Underneath the for loop, rewrite the problem so that it does the same thing, but using a while loop instead of a for loop. Assign the accumulated total in the while loop code to the variable sum2. Once complete, sum2 should equal sum1.
sum1 = 0
lst = [65, 78, 21, 33]
for x in lst:
sum1 = sum1 + x

7. Write two functions, one called addit and one called mult. addit takes one number as an input and adds 5. mult takes one number as an input and multiplies that input by whatever is returned by addit, and then returns the result.

8. Create a dictionary that keeps track of the USA's Olympic medal count. Each key of the dictionary should be the type of medal (gold, silver, or bronze) and each key's value should be the number of that type of medal the USA's won. Currently, the USA has 33 gold medals, 17 silver, and 12 bronze. Create a dictionary saved in the variable medals that reflects this information.
(Take the key and values from the user using input function)

9. Provided is a list of tuples. Create another list called to check that contains the third element of every tuple.
lst_tups = [('Articuno', 'Moltres', 'Zaptos'), ('Beedrill', 'Metapod', 'Charizard', 'Venasaur', 'Squirtle'), ('Oddish', 'Poliwag', 'Diglett','Bellsprout'), ('Ponyta', "Farfetch'd", "Tauros", 'Dragonite'),('Hoothoot', 'Chikorita', 'Lanturn', 'Flaaffy', 'Unown', 'Teddiursa', 'Phanpy'), ('Loudred', 'Volbeat', 'Wailord', 'Seviper', 'Sealeo')]

10. Below, is provided a list of strings called nums. Write a function called last_char that takes a string as input and returns only its last character. Use this function to sort the list nums by the last digit of each number, from highest to lowest, and save this as a new list called nums_sorted.
nums = ['1450', '33', '871', '19', '14378', '32', '1005', '44', '8907', '16']

Functions Challenge 1:
Head-to-Head Tic-Tac-Toe App
Description:
You are responsible for writing a program that will allow two users to play a game of tic tac toe. Your program should follow the standard rules in which two players alternate turns putting their pieces, X or O, on a board. If a player has three pieces in a row, either vertically, horizontally, or diagonally, they are declared the winner. You will represent the tic tac toe board using the integers 1 through 9 for the 9 spaces on the board. An empty spot on the

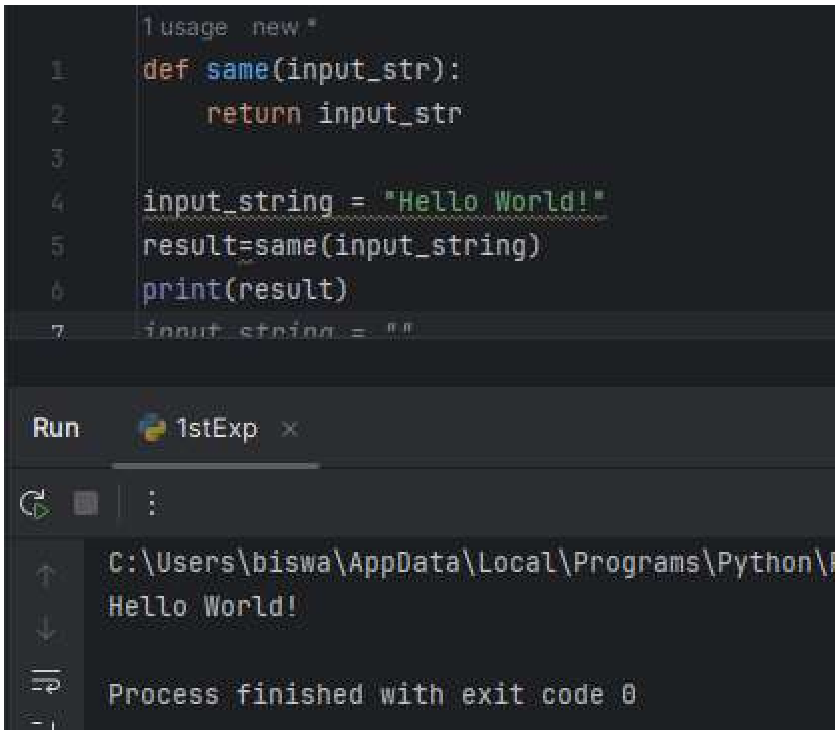
board will be represented by an underscore "_". For example, if a player would like to put a piece in the centre of the board, they would enter 5 as their move.

Example Output:

```
Tic-Tac-Toe
~~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~~
Tic-Tac-Toe
~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 5
Tic-Tac-Toe
~~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~~
Tic-Tac-Toe
~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 5
That spot has already been chosen. Try again.
O: Where would you like to place your piece (1 - 9): 15
That is not a spot on the board. Try again.
O: Where would you like to place your piece (1 - 9): 3
Tic-Tac-Toe
~~~~~~~~~~
|| 1 || 2 || 3 ||
~~~~~~~~~~
|| 4 || 5 || 6 ||
~~~~~~~~~~
|| 7 || 8 || 9 ||
~~~~~~~~~~
Tic-Tac-Toe
~~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~~
|| _ || X || _ ||
```

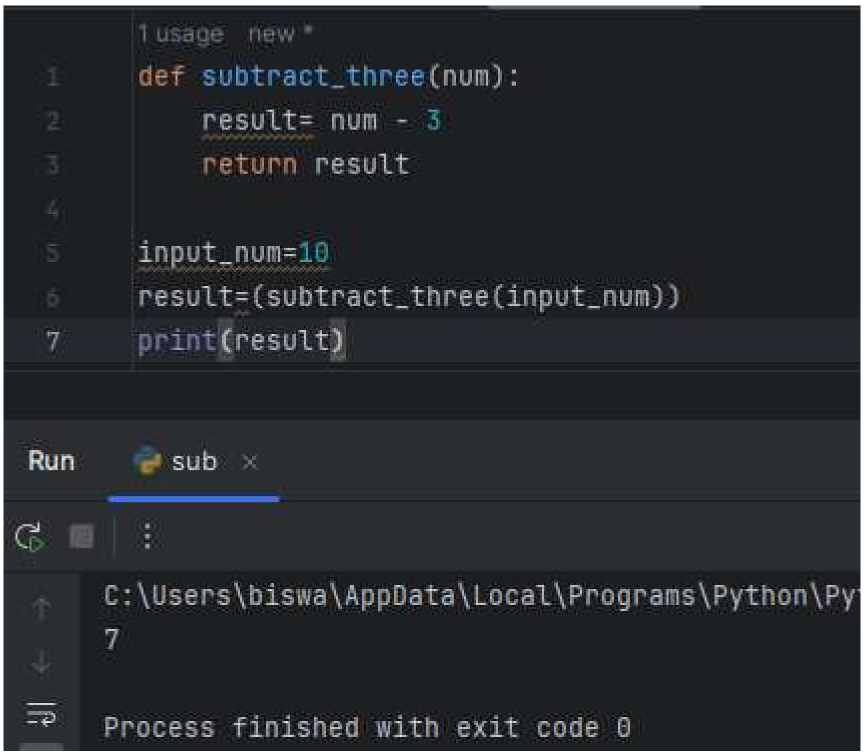
```
~~~~~~~~~~
| | _ | | _ | | _ | |
~~~~~~~~~~
```

Post Lab:
While Loops Challenge 29: Guess My Word App
Description:
You are responsible for writing a program that plays a word guessing game with a user. Your program will provide a category of words to the user and a string of dashes "-----" that represent the length of the word. The user will guess the word and with each incorrect guess, your program will reveal a letter at random, "-a---". Upon guessing the word correctly, your program will then inform the user how many guesses they took.



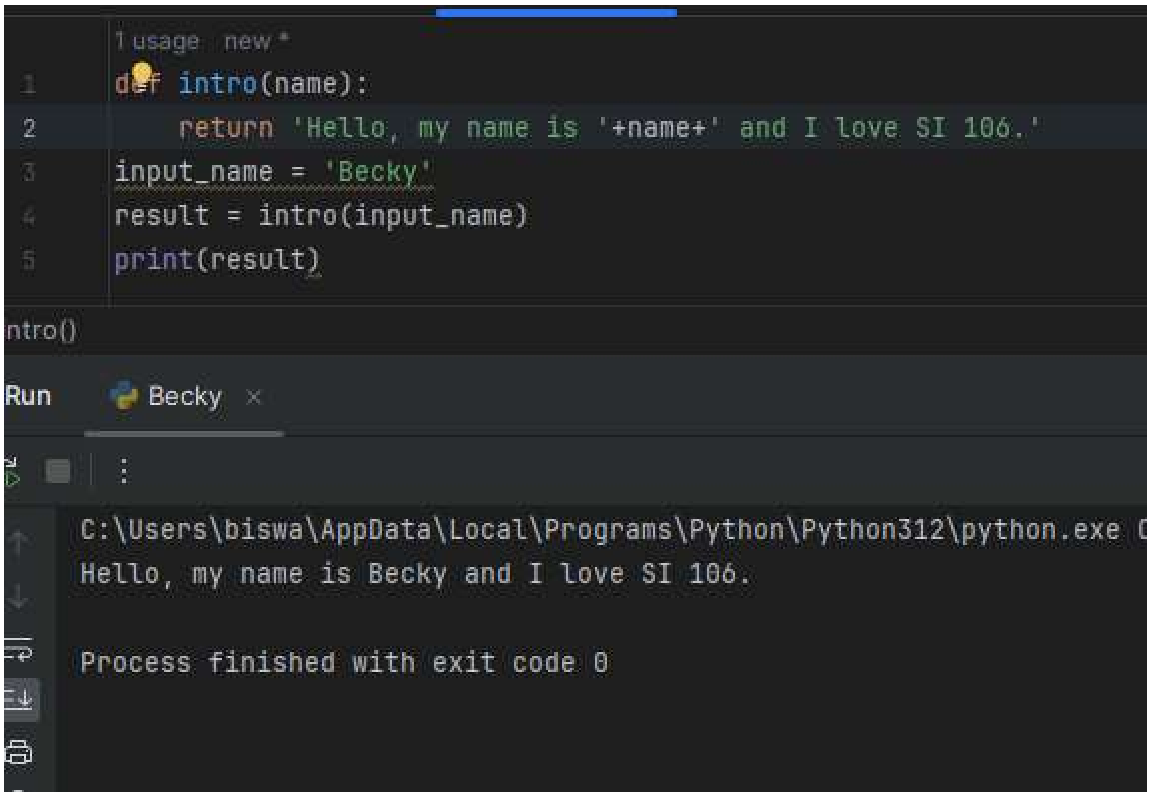
```python
def same(input_str):
    return input_str

input_string = "Hello World!"
result=same(input_string)
print(result)
input_string = ""
```

```
C:\Users\biswa\AppData\Local\Programs\Python\
Hello World!

Process finished with exit code 0
```



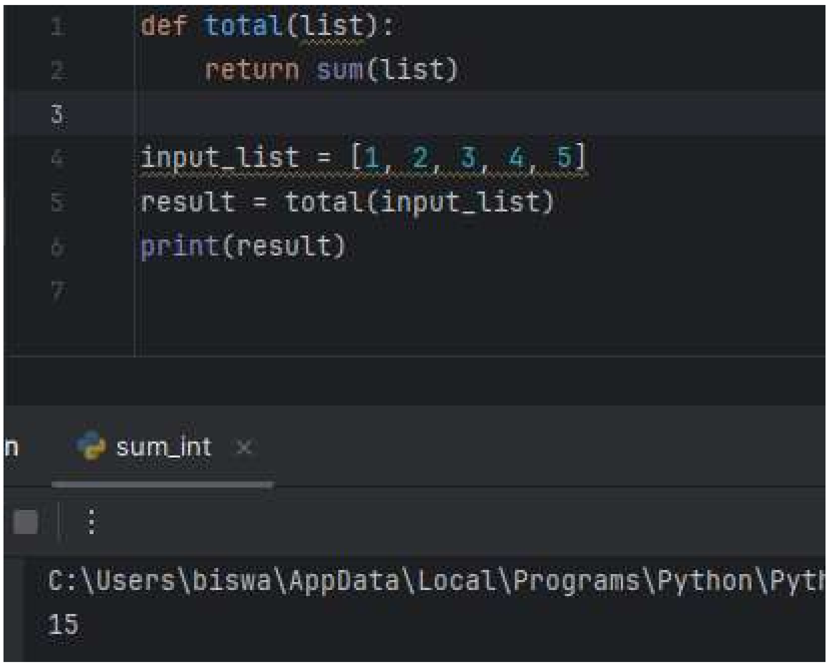
```python
def subtract_three(num):
    result= num - 3
    return result

input_num=10
result=(subtract_three(input_num))
print(result)
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Py
7

Process finished with exit code 0
```



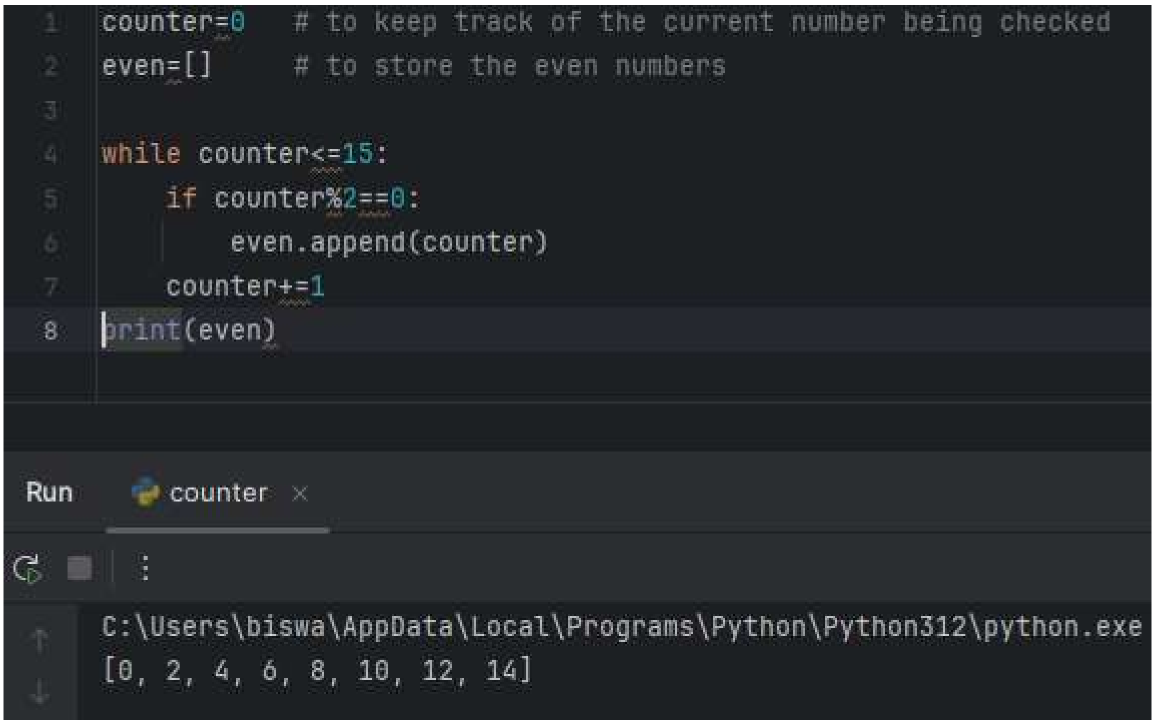
```python
def intro(name):
    return 'Hello, my name is '+name+' and I love SI 106.'
input_name = 'Becky'
result = intro(input_name)
print(result)
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C
Hello, my name is Becky and I love SI 106.

Process finished with exit code 0
```



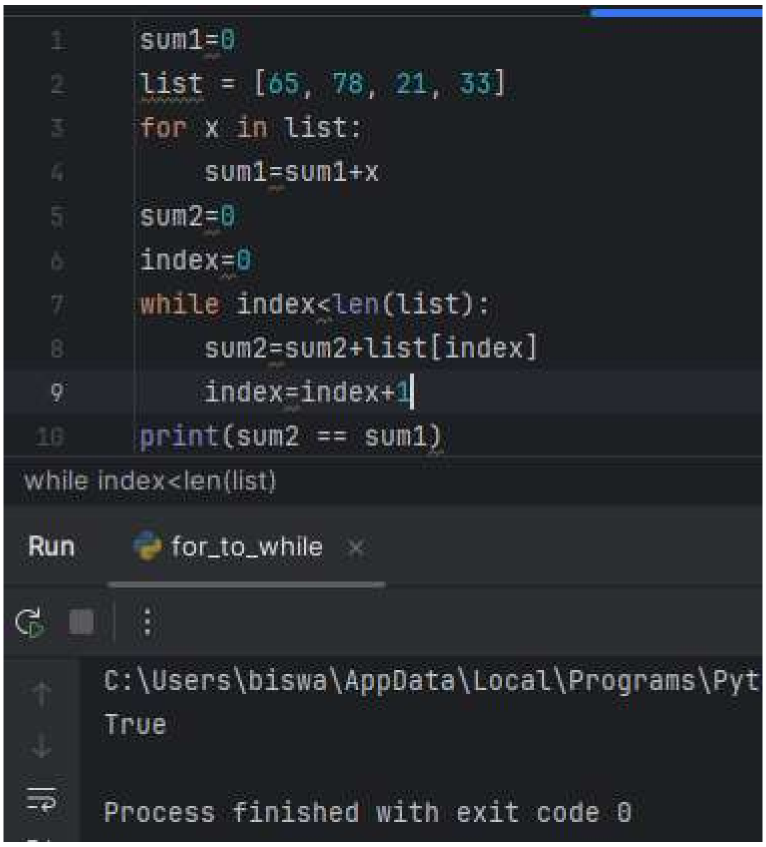
```python
def total(list):
    return sum(list)

input_list = [1, 2, 3, 4, 5]
result = total(input_list)
print(result)

```

```
C:\Users\biswa\AppData\Local\Programs\Python\Pyth
15
```



```python
counter=0    # to keep track of the current number being checked
even=[]      # to store the even numbers

while counter<=15:
    if counter%2==0:
        even.append(counter)
    counter+=1
print(even)
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe
[0, 2, 4, 6, 8, 10, 12, 14]
```

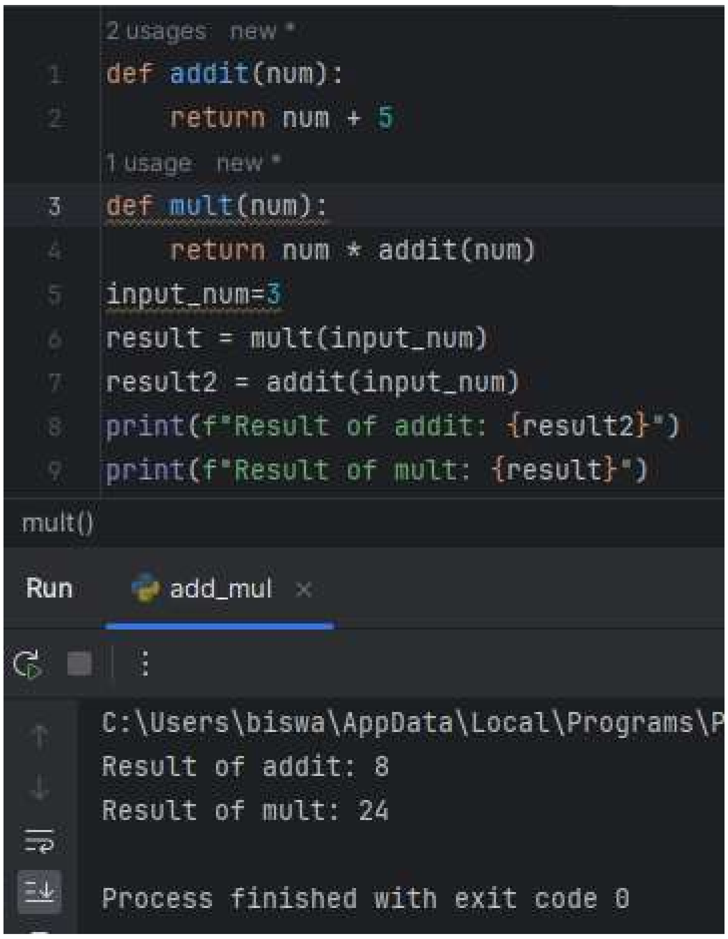


```python
sum1=0
list = [65, 78, 21, 33]
for x in list:
    sum1=sum1+x
sum2=0
index=0
while index<len(list):
    sum2=sum2+list[index]
    index=index+1
print(sum2 == sum1)
```

while index<len(list)

Run for_to_while

```
C:\Users\biswa\AppData\Local\Programs\Pyt
True

Process finished with exit code 0
```

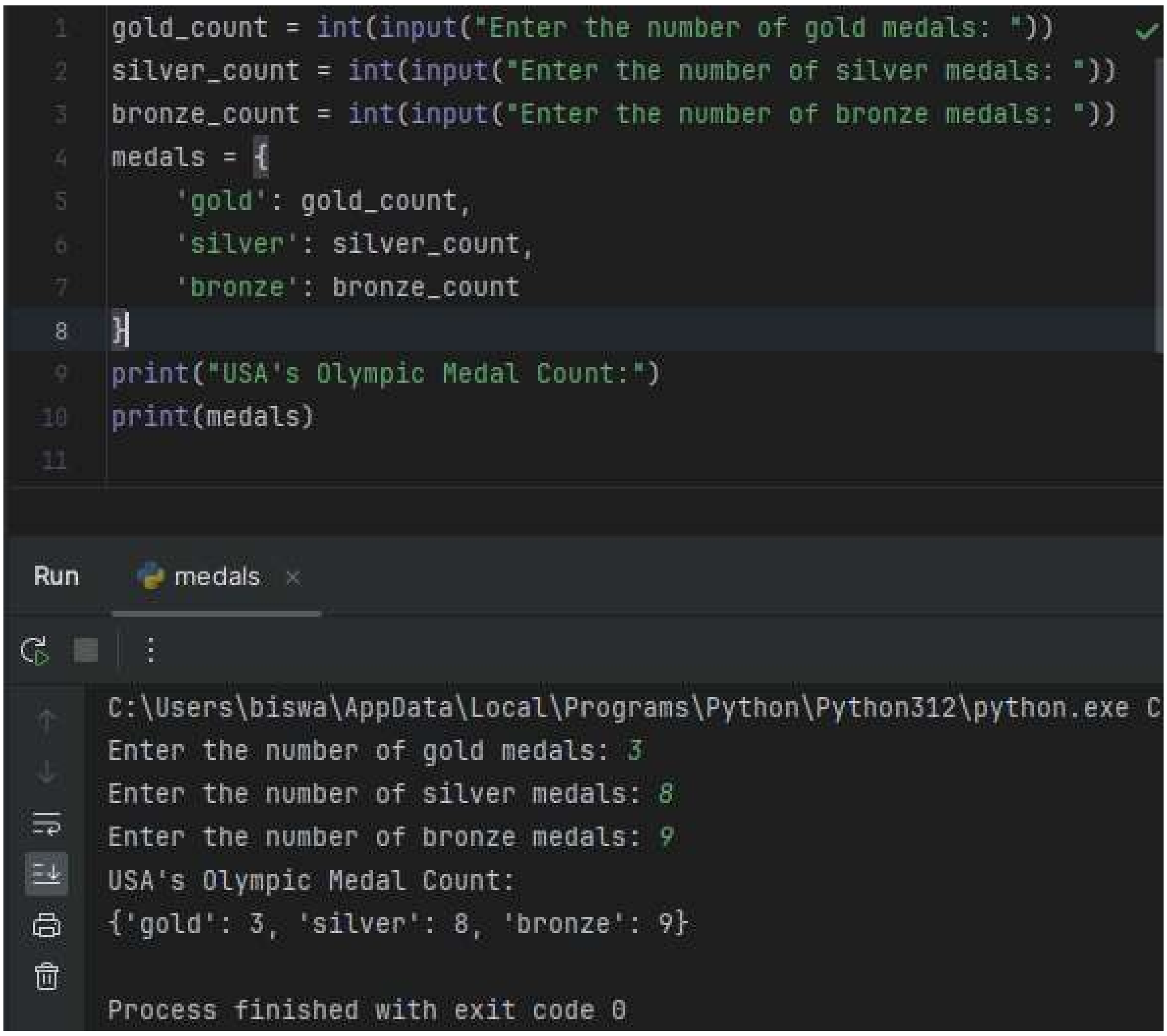


```python
2 usages   new *
def addit(num):
    return num + 5
1 usage   new *
def mult(num):
    return num * addit(num)
input_num=3
result = mult(input_num)
result2 = addit(input_num)
print(f"Result of addit: {result2}")
print(f"Result of mult: {result}")
```

mult()

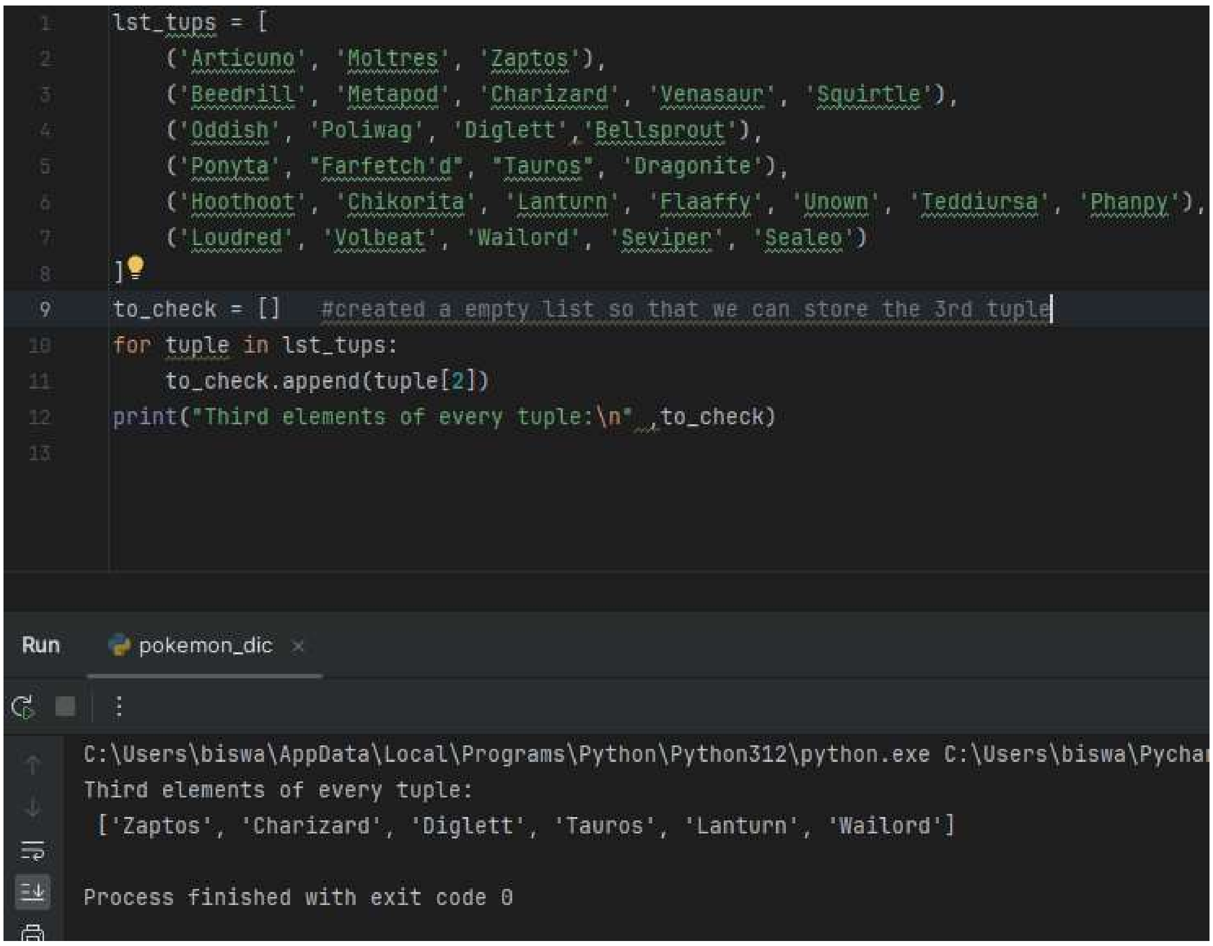
Run add_mul

```
C:\Users\biswa\AppData\Local\Programs\P
Result of addit: 8
Result of mult: 24

Process finished with exit code 0
```



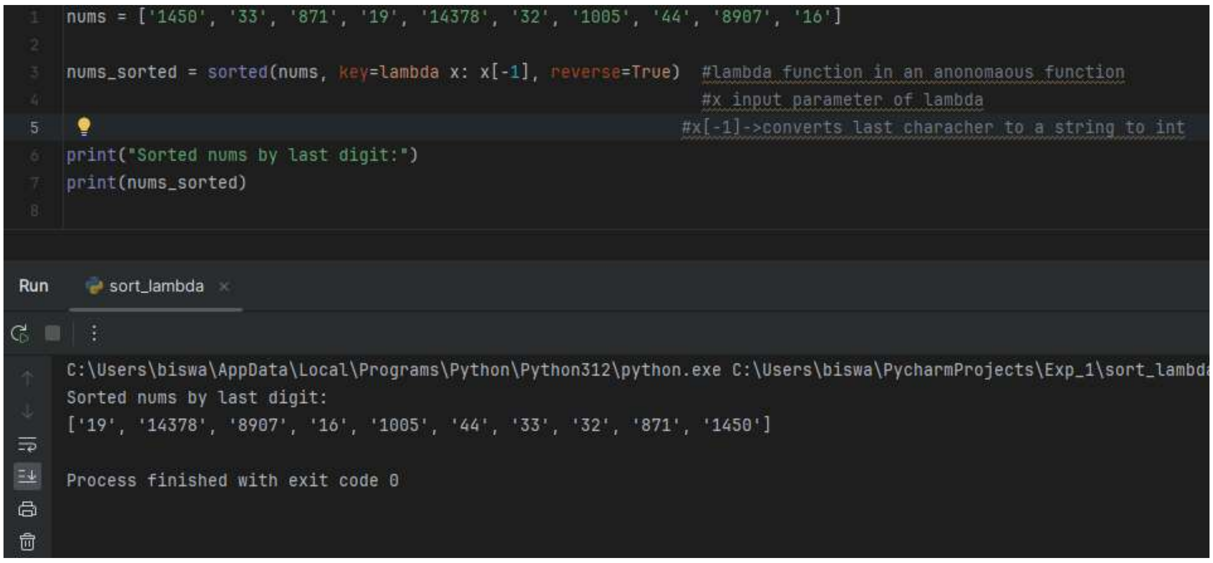
```python
gold_count = int(input("Enter the number of gold medals: "))
silver_count = int(input("Enter the number of silver medals: "))
bronze_count = int(input("Enter the number of bronze medals: "))
medals = {
    'gold': gold_count,
    'silver': silver_count,
    'bronze': bronze_count
}
print("USA's Olympic Medal Count:")
print(medals)

```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C
Enter the number of gold medals: 3
Enter the number of silver medals: 8
Enter the number of bronze medals: 9
USA's Olympic Medal Count:
{'gold': 3, 'silver': 8, 'bronze': 9}

Process finished with exit code 0
```



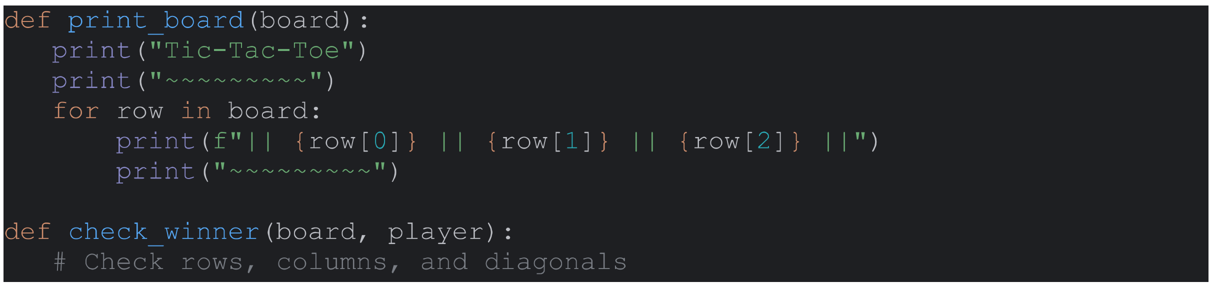
```python
lst_tups = [
    ('Articuno', 'Moltres', 'Zaptos'),
    ('Beedrill', 'Metapod', 'Charizard', 'Venasaur', 'Squirtle'),
    ('Oddish', 'Poliwag', 'Diglett','Bellsprout'),
    ('Ponyta', "Farfetch'd", "Tauros", 'Dragonite'),
    ('Hoothoot', 'Chikorita', 'Lanturn', 'Flaaffy', 'Unown', 'Teddiursa', 'Phanpy'),
    ('Loudred', 'Volbeat', 'Wailord', 'Seviper', 'Sealeo')
]
to_check = []   #created a empty list so that we can store the 3rd tuple
for tuple in lst_tups:
    to_check.append(tuple[2])
print("Third elements of every tuple:\n" ,to_check)
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C:\Users\biswa\Pychar
Third elements of every tuple:
 ['Zaptos', 'Charizard', 'Diglett', 'Tauros', 'Lanturn', 'Wailord']

Process finished with exit code 0
```



```python
nums = ['1450', '33', '871', '19', '14378', '32', '1005', '44', '8907', '16']

nums_sorted = sorted(nums, key=lambda x: x[-1], reverse=True)  #lambda function in an anonomaous function
                                                                #x input parameter of lambda
                                                             #x[-1]->converts last characher to a string to int
print("Sorted nums by last digit:")
print(nums_sorted)
```

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe C:\Users\biswa\PycharmProjects\Exp_1\sort_lambda
Sorted nums by last digit:
['19', '14378', '8907', '16', '1005', '44', '33', '32', '871', '1450']

Process finished with exit code 0
```



```python
def print_board(board):
   print("Tic-Tac-Toe")
   print("~~~~~~~~~~")
   for row in board:
       print(f"|| {row[0]} || {row[1]} || {row[2]} ||")
       print("~~~~~~~~~~")

def check_winner(board, player):
   # Check rows, columns, and diagonals
```

```python
    for i in range(3):
        if all(board[i][j] == player for j in range(3)) or all(board[j][i]
== player for j in range(3)):
            return True
    if all(board[i][i] == player for i in range(3)) or all(board[i][2 - i]
== player for i in range(3)):
        return True
    return False

def tic_tac_toe():
    board = [["_" for _ in range(3)] for _ in range(3)]
    player_turn = True  # True for player X, False for player O

    while True:
        print_board(board)

        player = "X" if player_turn else "O"
        move = input(f"{player}: Where would you like to place your piece (1
- 9): ")

        try:
            move = int(move)
            if 1 <= move <= 9:
                row, col = (move - 1) // 3, (move - 1) % 3

                if board[row][col] == "_":
                    board[row][col] = player
                    if check_winner(board, player):
                        print_board(board)
                        print(f"{player} is the winner!")
                        break
                    elif all(board[i][j] != "_" for i in range(3) for j in
range(3)):
                        print_board(board)
                        print("It's a tie!")
                        break
                    else:
                        player_turn = not player_turn
                else:
                    print("That spot has already been chosen. Try again.")
            else:
                print("That is not a spot on the board. Try again.")
        except ValueError:
            print("Invalid input. Please enter a number between 1 and 9.")

if __name__ == "__main__":
    tic_tac_toe()
```

OUTPUT:

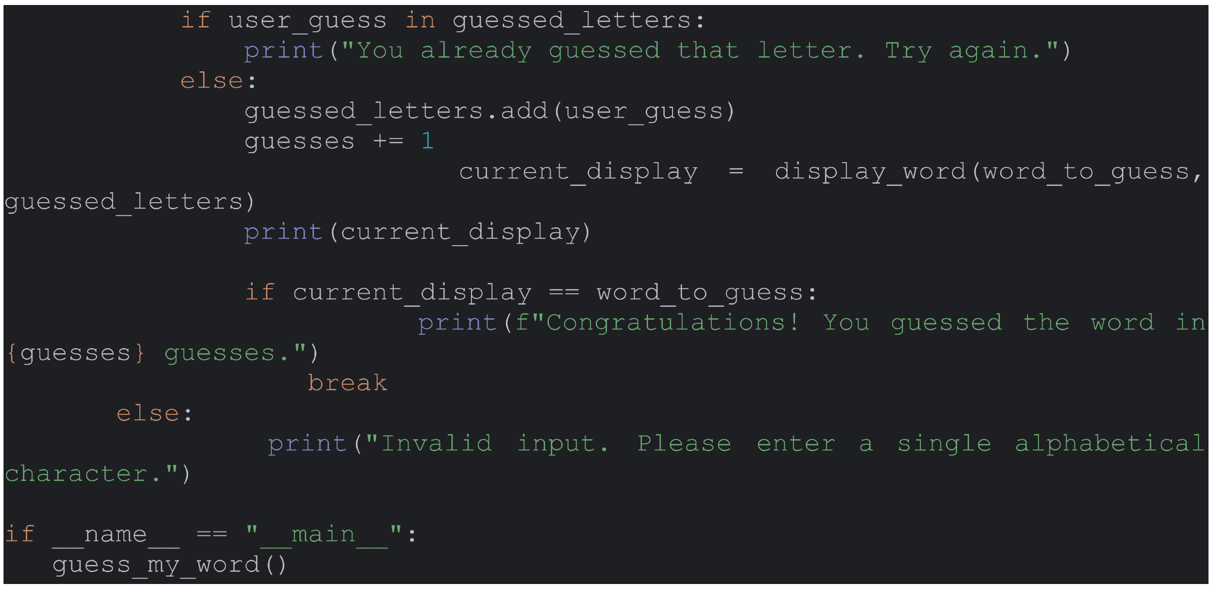
```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\biswa\PycharmProjects\Exp_1\Tic-Tac-Toe.py
Tic-Tac-Toe
~~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~~~
|| _ || _ || _ ||
```

```
~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 1
Tic-Tac-Toe
~~~~~~~~~
|| X || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 1
That spot has already been chosen. Try again.
Tic-Tac-Toe
~~~~~~~~~
|| X || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 2
Tic-Tac-Toe
~~~~~~~~~
|| X || O || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 5
Tic-Tac-Toe
~~~~~~~~~
|| X || O || _ ||
~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~
|| _ || _ || _ ||
~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 9
Tic-Tac-Toe
~~~~~~~~~
|| X || O || _ ||
~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~
|| _ || _ || O ||
~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 7
Tic-Tac-Toe
~~~~~~~~~
```

```
|| X || O || _ ||
~~~~~~~~~~
|| _ || X || _ ||
~~~~~~~~~~
|| X || _ || O ||
~~~~~~~~~~
O: Where would you like to place your piece (1 - 9): 4
Tic-Tac-Toe
~~~~~~~~~~
|| X || O || _ ||
~~~~~~~~~~
|| O || X || _ ||
~~~~~~~~~~
|| X || _ || O ||
~~~~~~~~~~
X: Where would you like to place your piece (1 - 9): 3
Tic-Tac-Toe
~~~~~~~~~~
|| X || O || X ||
~~~~~~~~~~
|| O || X || _ ||
~~~~~~~~~~
|| X || _ || O ||
~~~~~~~~~~
X is the winner!

Process finished with exit code 0
```

POSTLAB:

```python
import random

def choose_word():
    words = ["python", "programming", "challenge", "developer", "coding",
"computer"]
   return random.choice(words)

def display_word(word, guessed_letters):
    return ''.join(letter if letter in guessed_letters else '-' for letter
in word)

def guess_my_word():
   print("Welcome to the Guess My Word App!")

   word_to_guess = choose_word()
   guessed_letters = set()
   guesses = 0

   print(f"\nThe word has {len(word_to_guess)} letters.")
   print(display_word(word_to_guess, guessed_letters))

   while True:
       user_guess = input("\nEnter your guess: ").lower()

       if len(user_guess) == 1 and user_guess.isalpha():
```



```
            if user_guess in guessed_letters:
                print("You already guessed that letter. Try again.")
            else:
                guessed_letters.add(user_guess)
                guesses += 1
                            current_display  =  display_word(word_to_guess,
guessed_letters)
                print(current_display)

                if current_display == word_to_guess:
                           print(f"Congratulations! You guessed the word in
{guesses} guesses.")
                    break
        else:
                 print("Invalid  input.  Please  enter  a  single  alphabetical
character.")

if __name__ == "__main__":
   guess_my_word()
```

OUTPUT:

```
C:\Users\biswa\AppData\Local\Programs\Python\Python312\python.exe
C:\Users\biswa\PycharmProjects\Exp_1\guess_the_no..py
Welcome to the Guess My Word App!

The word has 9 letters.
---------

Enter your guess: l
---ll----

Enter your guess: c
c--ll----

Enter your guess: h
ch-ll----

Enter your guess: a
chall----

Enter your guess: e
challe--e

Enter your guess: n
challen-e

Enter your guess: g
challenge
Congratulations! You guessed the word in 7 guesses.

Process finished with exit code 0
```