

1.interchange first and last elem in a list

```
def interchange_first_last_elements(list1):
    first_element = list1[0]
    last_element = list1[-1]
    list1[0] = last_element
    list1[-1] = first_element
    return list1
def main():
    list1 = [1, 2, 3, 4, 5]
    print("The original list is: ", list1)
    list1 = interchange_first_last_elements(list1)
    print("The list after interchanging elements is: ", list1)
if __name__ == "__main__":
    main()
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:\python programme\interchange first and last elem.py =====
The original list is: [1, 2, 3, 4, 5]
The list after interchanging elements is: [5, 2, 3, 4, 1]
>>>
```

2.max in two num

```
def maximum(a,b):
    if a > b:
        return a
    else:
        return b
```

```

def main():
    a = 10
    b = 20
    print("The maximum number is {}".format(maximum(a,b)))
if __name__ == "__main__":
    main()

```

IDLE Shell 3.8.7

File Edit Shell Debug Options Window Help

Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32

Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: D:/python programme/max between two num.py =====

The maximum number is 20

>>> |

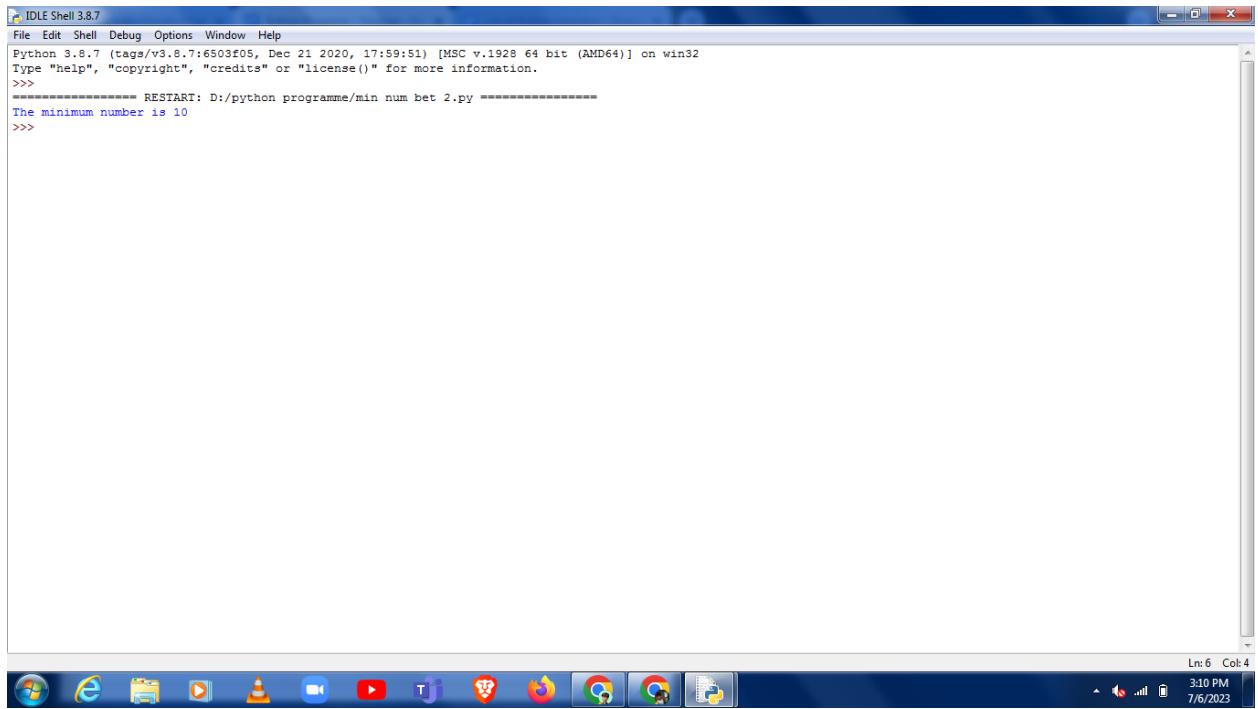
Ln: 6 Col: 4

3.min between 2 num

```

def minimum(a,b):
    if a < b:
        return a
    else:
        return b
def main():
    a = 10
    b = 20
    print("The minimum number is {}".format(minimum(a,b)))
if __name__ == "__main__":
    main()

```



The screenshot shows the Python IDLE Shell window. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/min num bet 2.py =====
The minimum number is 10
>>>
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:10 PM", and the date "7/6/2023".

4.element exist in the list

```
list1 = [1, 2, 3, 4, 5]
```

```
if 5 in list1:
```

```
    print("The value 5 exists in the list")
```

```
else:
```

```
    print("The value 5 does not exist in the list")
```

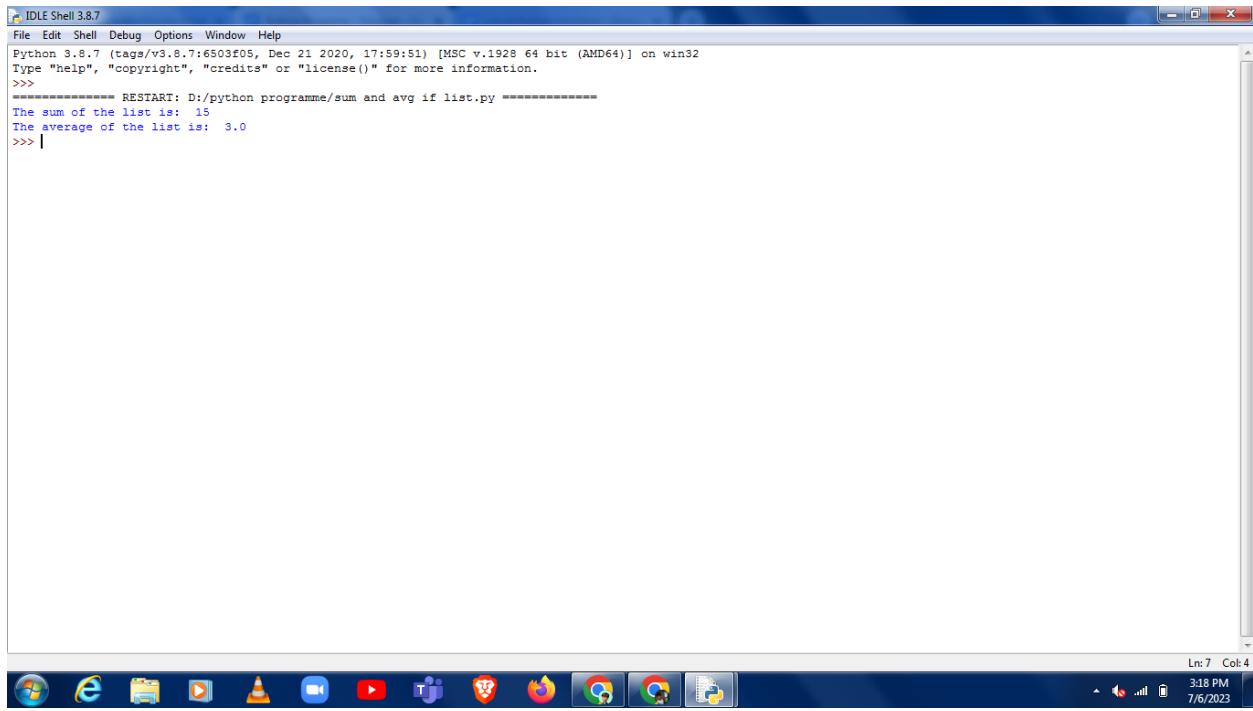
The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays a Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/element exists in the list.py =====
The value 5 exists in the list
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 3:13 PM, and 7/6/2023.

5.find sum and avg in a list

```
def find_sum_and_average(list1):
    sum = 0
    for element in list1:
        sum += element
    average = sum / len(list1)
    return sum, average
def main():
    list1 = [1, 2, 3, 4, 5]
    sum,average = find_sum_and_average(list1)
    print("The sum of the list is: ", sum)
    print("The average of the list is: ", average)
if __name__ == "__main__":
    main()
```



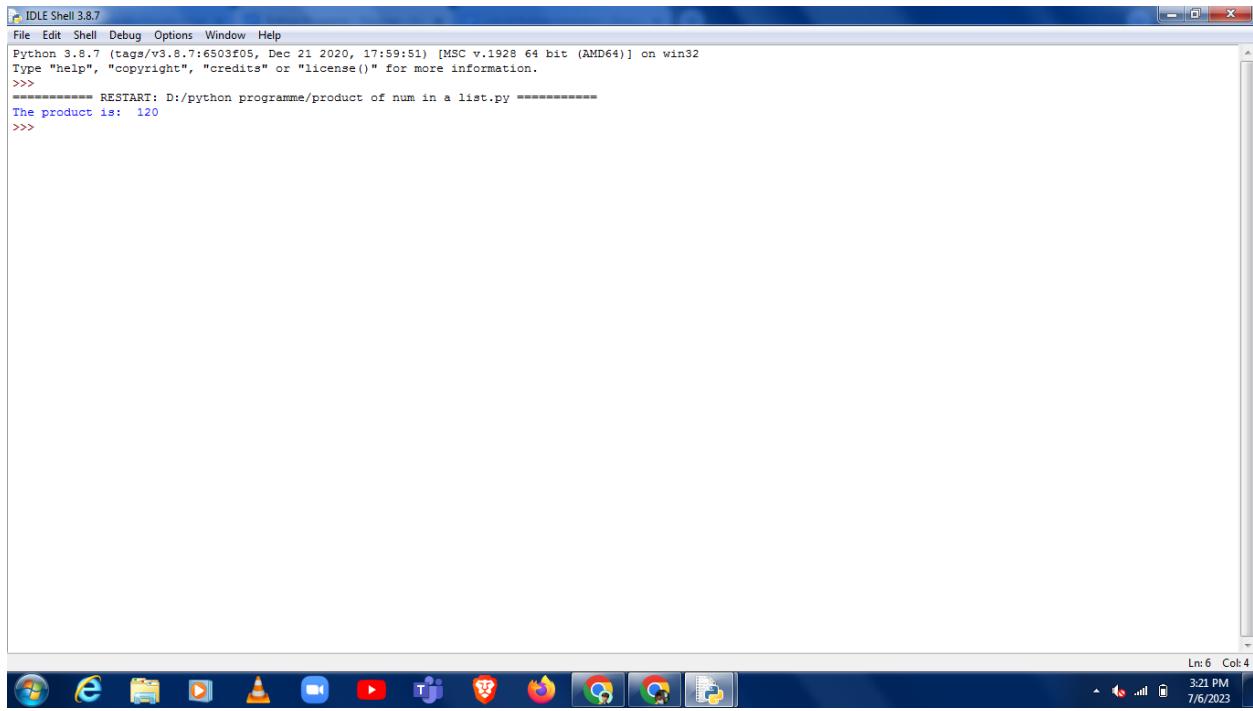
The screenshot shows the Python IDLE Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell window displays the following text:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/sum and avg if list.py =====
The sum of the list is: 15
The average of the list is: 3.0
>>> |
```

The status bar at the bottom right shows Ln: 7 Col: 4, 3:18 PM, and 7/6/2023.

6.product of numbers in a list

```
def multiply_all_numbers(list1):
    product = 1
    for number in list1:
        product *= number
    return product
def main():
    list1 = [1, 2, 3, 4, 5]
    product = multiply_all_numbers(list1)
    print("The product is: ", product)
if __name__ == "__main__":
    main()
```



The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/product of num in a list.py =====
The product is: 120
>>>
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:21 PM", and "7/6/2023".

7. Python program to find smallest number in a list

```
def find_smallest_number(list1):
    smallest_number = list1[0]
    for number in list1:
        if number < smallest_number:
            smallest_number = number
    return smallest_number

def main():
    list1 = [10, 4, 2, 9, 7]
    smallest_number = find_smallest_number(list1)
    print("The smallest number is: ", smallest_number)
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following text:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/smallest num in the list.py =====
The smallest number is: 2
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:23 PM", and "7/6/2023". Below the status bar is a taskbar with various icons.

8.largest num in a list

```
def find_largest_number(list1):
    largest_number = list1[0]
    for number in list1:
        if number > largest_number:
            largest_number = number
    return largest_number
def main():
    list1 = [18, 90, 954, 34, 7]
    largest_number = find_largest_number(list1)
    print("The largest number is: ", largest_number)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays a Python script named "smallest num in the list.py" with the following content:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/smallest num in the list.py =====
The largest number is: 954
>>>
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:26 PM", and the date "7/6/2023".

9.print eve num in a list

```
def print_even_numbers(list1):
    for number in list1:
        if number % 2 == 0:
            print(number)
def main():
    list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
    print_even_numbers(list1)
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print eve num in a list.py =====
2
4
6
8
>>> |
```

The status bar at the bottom right shows "Ln: 9 Col: 4", "3:34 PM", and the date "7/6/2023". Below the status bar is a taskbar with various icons.

10.print odd num in a list

```
def print_odd_numbers(list1):
    for number in list1:
        if number % 2 != 0:
            print(number)
def main():
    list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
    print_odd_numbers(list1)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays Python version information and a command-line session. The session starts with a restart command, followed by the output of a list of odd numbers from 1 to 9. The status bar at the bottom right shows Ln: 10 Col: 4, 3:36 PM, and 7/6/2023.

```
>>> ===== RESTART: D:/python programme/print odd num iin list.py =====
1
3
5
7
9
>>> |
```

```
11.print eve num b/w interval
def print_even_numbers(start,end):
    for number in range(start,end + 1):
        if number % 2 == 0:
            print(number)
def main():
    start = 20
    end = 40
    print_even_numbers(start, end)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The status bar at the bottom right shows "Ln: 16 Col: 4", "3:40 PM", and the date "7/6/2023". The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/eve num between an interval.py =====
20
22
24
26
28
30
32
34
36
38
40
>>>
```

```
12.print odd num b/w interval
def print_even_numbers(start,end):
    for number in range(start,end + 1):
        if number % 2 != 0:
            print(number)
def main():
    start = 30
    end = 50
    print_even_numbers(start, end)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The shell area displays the following text:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print odd num between an interval.py =====
31
33
35
37
39
41
43
45
47
49
>>> |
```

The status bar at the bottom right shows "Ln: 15 Col: 4", "3:42 PM", and the date "7/6/2023". The taskbar at the bottom has icons for various Windows applications.

13. Python program to print positive numbers in a list

```
def print_positive_numbers(list1):
    for number in list1:
        if number >= 0:
            print(number)
def main():
    list1 = [-1, 2, 3, -4, 5, -6, 7, -8, 9]
    print_positive_numbers(list1)
if __name__ == "__main__":
    main()
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/positive num in a list.py =====
2
3
5
7
9
>>> |
```

14.negative num in a list

```
def print_negative_numbers(list1):
    for number in list1:
        if number <= 0:
            print(number)
def main():
    list1 = [-1, 2, 3, -4, 5, -6, 7, -8, 9]
    print_negative_numbers(list1)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/negative num in a list.py =====
-1
-4
-6
-8
>>> |
```

The status bar at the bottom right shows Ln: 9 Col: 4, 3:46 PM, and 7/6/2023.

15. Python program to count Even and Odd numbers in a List

```
def count_even_odd_numbers(list1):
    even_count = 0
    odd_count = 0
    for number in list1:
        if number % 2 == 0:
            even_count += 1
        else:
            odd_count += 1
    return even_count, odd_count
def main():
    list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
    even_count, odd_count = count_even_odd_numbers(list1)
    print("The no.of even numbers in the list is: ", even_count)
    print("The no.of odd numbers in the list is: ", odd_count)
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/no of even and odd num in the list.py =====
The no.of even numbers in the list is: 4
The no.of odd numbers in the list is: 5
>>>
```

The status bar at the bottom right shows Ln: 7 Col: 4, 3:51 PM, and 7/6/2023. The taskbar below the window contains icons for various Windows applications like File Explorer, Edge, and File Explorer.

16.print +ve nums in a range

```
def print_positive_numbers_in_range(start, end):
    for number in range(start, end + 1):
        if number >= 0:
            print(number)
def main():
    start = -10
    end = 10
    print_positive_numbers_in_range(start, end)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print positive num in a range.py =====
0
1
2
3
4
5
6
7
8
9
10
>>> |
```

The status bar at the bottom right shows "Ln: 16 Col: 4", "3:53 PM", and "7/6/2023". The taskbar at the bottom has icons for various Windows applications.

17.print -ve nums in a range

```
def print_positive_numbers_in_range(start, end):
    for number in range(start, end + 1):
        if number < 0:
            print(number)
def main():
    start = -12
    end = 5
    print_positive_numbers_in_range(start, end)
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its execution:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print negative nums in a range.py =====
-12
-11
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
>>>
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date as 7/6/2023, the time as 3:55 PM, and battery level.

18.no of +ve and -ve nums in a list

```
def count_positive_negative_numbers(list1):
    positive_count = 0
    negative_count = 0
    for number in list1:
        if number >= 0:
            positive_count += 1
        else:
            negative_count += 1
    return positive_count, negative_count
def main():
    list1 = [1, 2, 3, 4, 5, -6, -7, -8, 9]
    positive_count, negative_count = count_positive_negative_numbers(list1)
    print("The no.of positive numbers in the list is: ", positive_count)
    print("The no.of negative numbers in the list is: ", negative_count)
if __name__ == "__main__":
    main()
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: D:/python programme/no of positive and negative nums in a list.py ==
The no.of positive numbers in the list is: 6
The no.of negative numbers in the list is: 3
>>>
```

Ln: 7 Col: 4
3:59 PM 7/6/2023

19.remove multiple elem from list

```
list1 = [1,2,3,4,5]
```

```
list1.remove(2)
```

```
list1.remove(4)
```

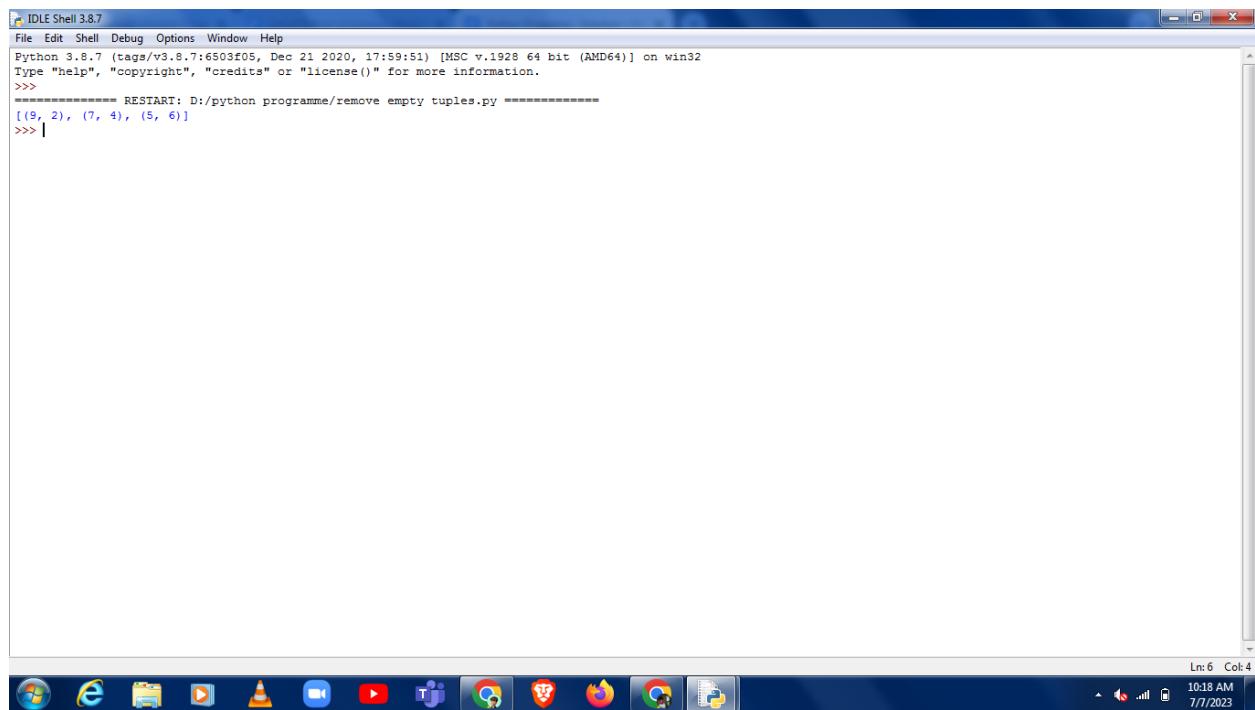
```
print(list1)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/remove multiple elements from the list.py =====
[1, 3, 5]
>>>
```

Ln: 6 Col: 4
10:15 AM 7/7/2023

20.remove empty tuples

```
def remove_empty_tuples(list1):
    new_list = []
    for tuple in list1:
        if tuple:
            new_list.append(tuple)
    return new_list
list1 = [(9, 2), (), (7, 4), (), (5, 6)]
new_list = remove_empty_tuples(list1)
print(new_list)
```



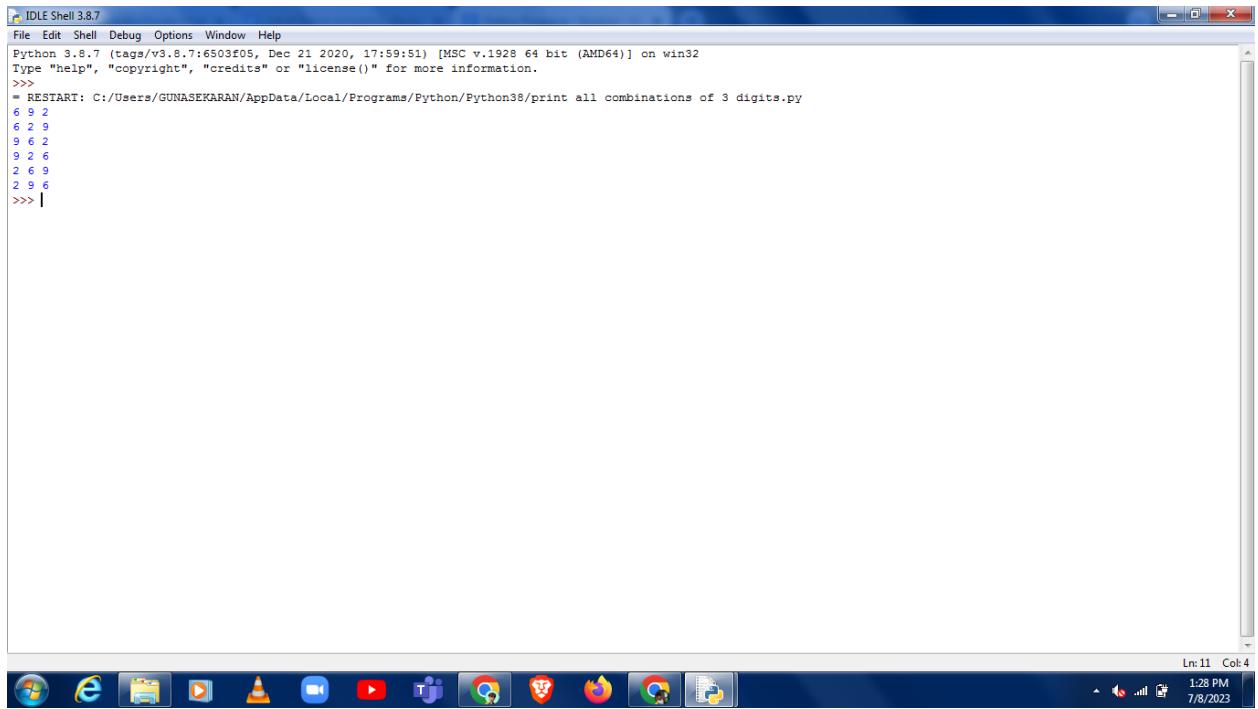
The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell window displays the following text:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/remove empty tuples.py =====
[(9, 2), (7, 4), (5, 6)]
>>>
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 10:18 AM, and 1/7/2023.

21.print all combinations of 3 nums

```
def print_all_combinations(digits):
    for i in range(3):
        for j in range(3):
            for k in range(3):
                if i != j and i != k and j != k:
                    print(digits[i], digits[j], digits[k])
if __name__ == "__main__":
    digits = [6,9,2]
    print_all_combinations(digits)
```



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/GUNASEKARAN/AppData/Local/Programs/Python/Python38/print all combinations of 3 digits.py
6 9 2
6 2 9
9 6 2
9 2 6
2 6 9
2 9 6
>>> |
```

22.remove all occurrences of a elem from the list

```
def remove_all_occurrences(list_of_elements,element):
    new_list = []
    for item in list_of_elements:
        if item != element:
            new_list.append(item)
    return new_list
if __name__ == "__main__":
    list_of_elements = [1,2,3,1,2,3,4,5,6,7,8,9]
    element = 3
    new_list = remove_all_occurrences(list_of_elements, element)
    print(new_list)
```

The screenshot shows the Python IDLE Shell window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/remove all occurences from the list.py =====
[1, 2, 1, 2, 4, 5, 6, 7, 8, 9]
>>>
```

The status bar at the bottom right shows Ln: 6 Col: 4, 1:32 PM, and 7/8/2023.

23.count string with substring

```
def count_strings_with_substring(string_list, substring):
    count = 0
    for string in string_list:
        if substring in string:
            count += 1
    return count
if __name__ == "__main__":
    string_list = ["hi", "hari", "hi hari"]
    substring = "hari"
    count = count_strings_with_substring(string_list, substring)
    print(count)
```

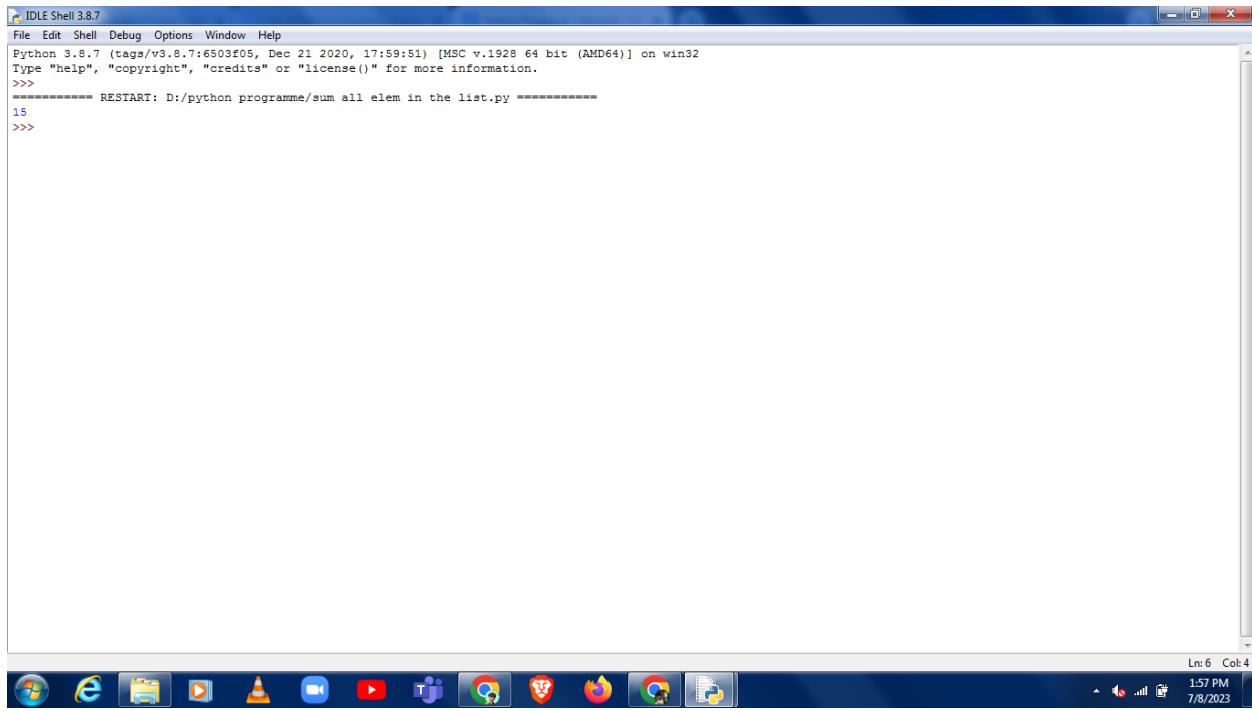
The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays Python code and its output. The code is as follows:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/count string with substring.py =====
2
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 1:49 PM, and 7/8/2023.

24.sum all the elem in the list

```
def sum_list_items(list_of_items):
    sum = 0
    for item in list_of_items:
        sum += item
    return sum
if __name__ == "__main__":
    list_of_items = [1,2,3,4,5]
    sum = sum_list_items(list_of_items)
    print(sum)
```

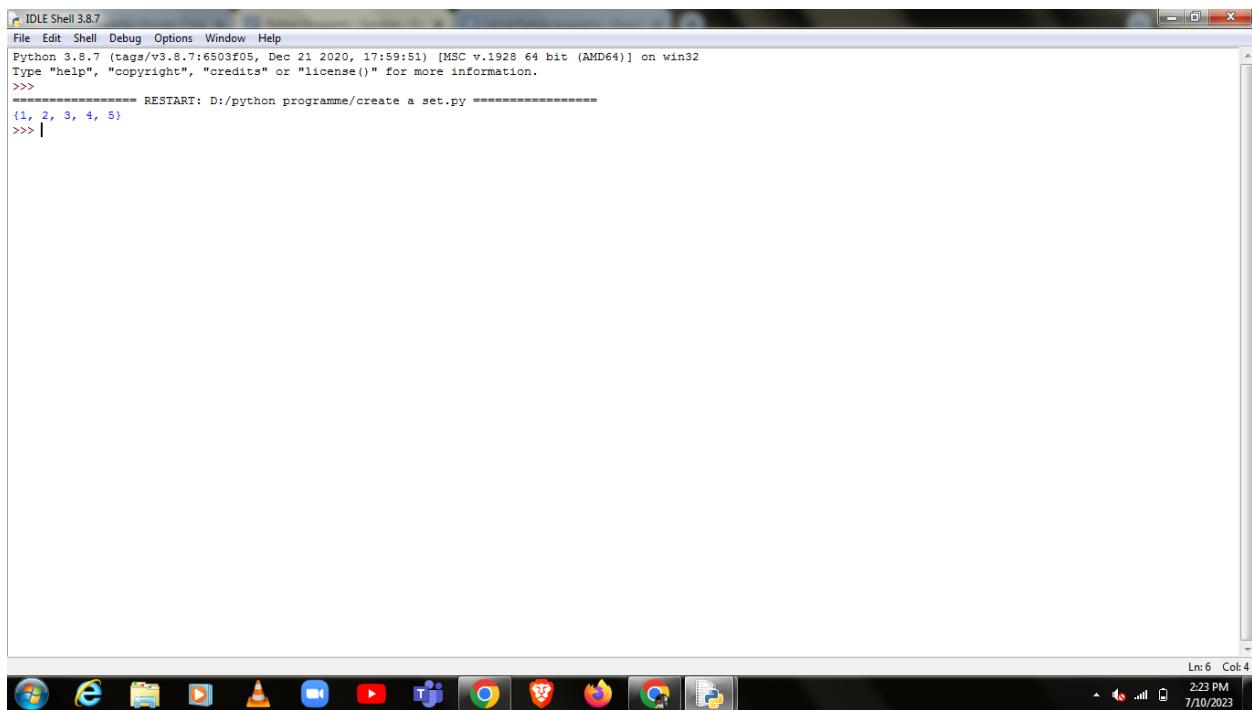


```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/sum all elem in the list.py =====
15
>>>
```

Ln: 6 Col: 4
1:57 PM 7/8/2023

25.create a set

```
def create_set():
    set1 = {1, 2, 3, 4, 5}
    print(set1)
if __name__ == "__main__":
    create_set()
```

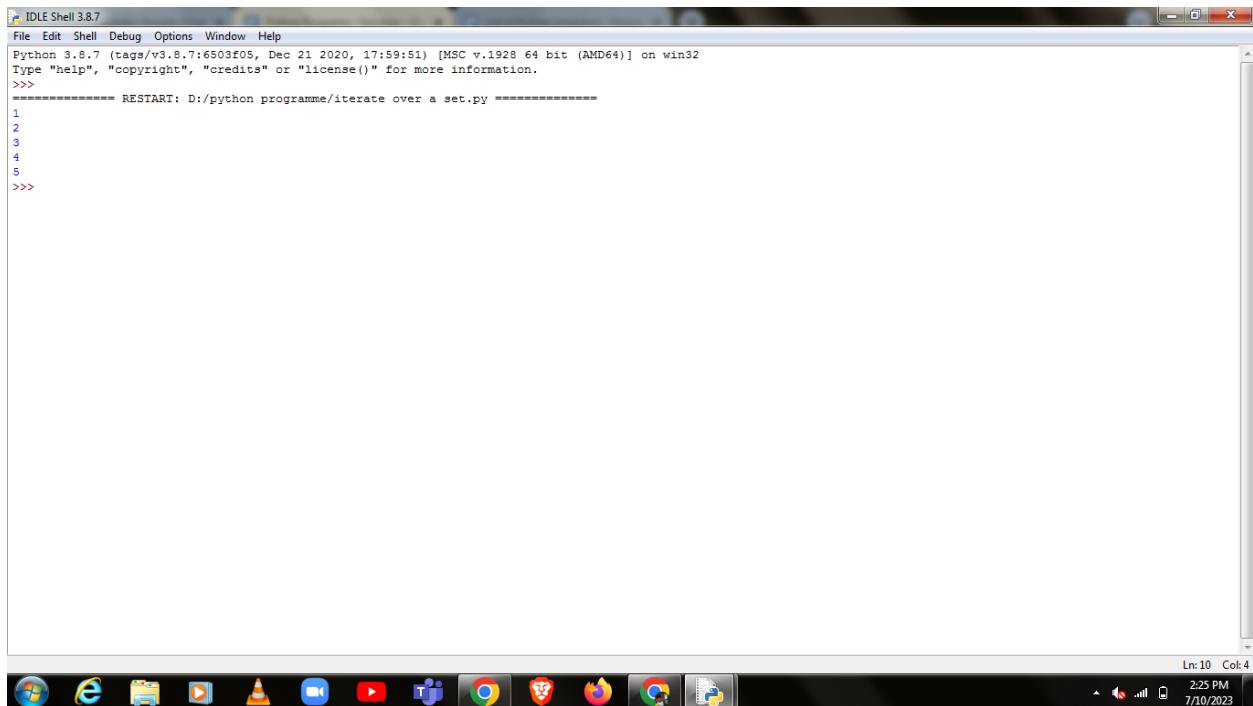


```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/create a set.py =====
(1, 2, 3, 4, 5)
>>> |
```

Ln: 6 Col: 4
2:23 PM 7/10/2023

26.iterate over a set

```
def iterate_over_set():
    set1 = {1, 2, 3, 4, 5}
    for item in set1:
        print(item)
if __name__ == "__main__":
    iterate_over_set()
```



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/iterate over a set.py =====
1
2
3
4
5
>>>
```

Ln: 10 Col: 4
2:25 PM 7/10/2023

27.add members in a set

```
def add_members_to_set():
    set1 = {1, 2, 3, 4, 5}
    set1.add(6)
    set1.add(7)
    print(set1)
if __name__ == "__main__":
    add_members_to_set()
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/add elem to set.py =====
{1, 2, 3, 4, 5, 6, 7}
>>> |
```

28.del elem from a set

```
def remove_items_from_set():
    set1 = {1, 2, 3, 4, 5}
    set1.remove(2)
    set1.remove(4)
    set1.remove(5)
    print(set1)
if __name__ == "__main__":
    remove_items_from_set()
```

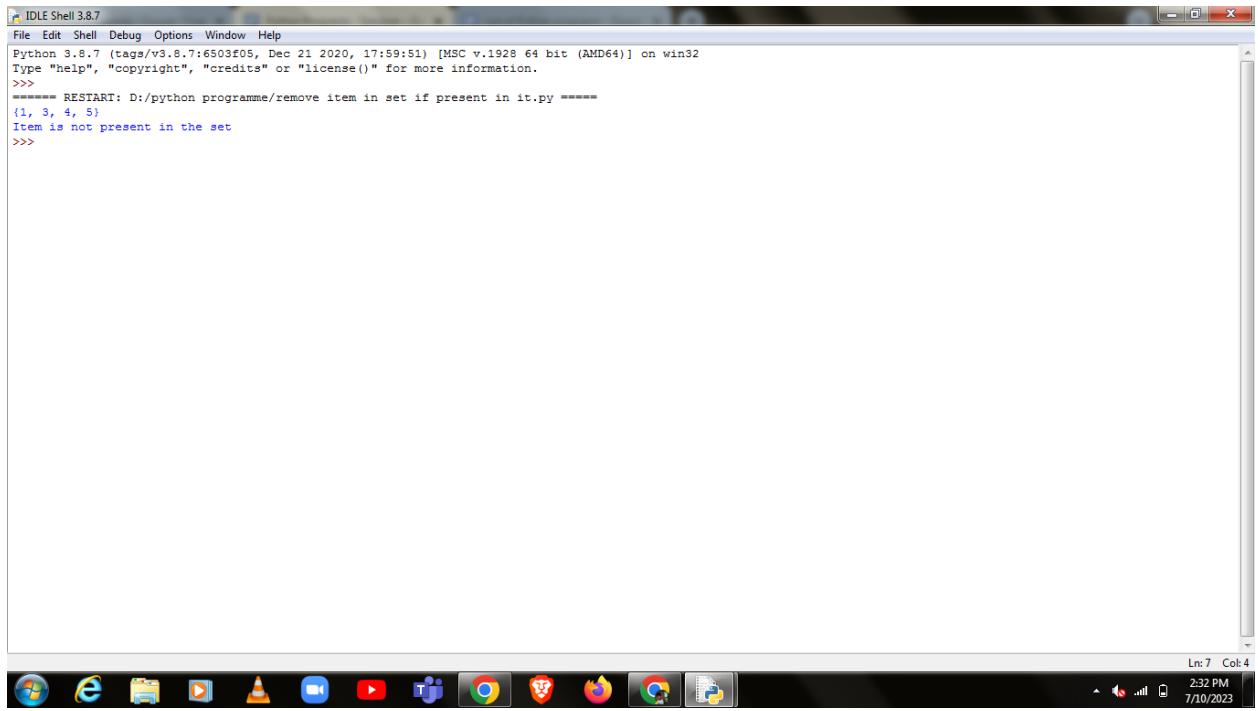
The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/remove elem from the set.py =====
{1, 3}
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:29 PM", and the date "7/10/2023". The taskbar at the bottom of the screen contains icons for various applications, including a browser, file explorer, and video player.

29.del elem from the set if it is present in it

```
def remove_item_if_present(item):
    set1 = {1, 2, 3, 4, 5}
    if item in set1:
        set1.remove(item)
        print(set1)
    else:
        print("Item is not present in the set")
if __name__ == "__main__":
    remove_item_if_present(2)
    remove_item_if_present(6)
```



The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/remove item in set if present in it.py =====
{1, 3, 4, 5}
Item is not present in the set
>>>
```

The status bar at the bottom right indicates Ln: 7 Col: 4, 2:32 PM, and 7/10/2023.

30.create intersection of sets

```
def create_intersection_of_sets():
    set1 = {1, 2, 3, 4, 5}
    set2 = {3, 4, 5, 6, 7}
    intersection = set1 & set2
    print(intersection)
if __name__ == "__main__":
    create_intersection_of_sets()
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/create intersection of sets.py =====
{3, 4, 5}
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:34 PM", and "7/10/2023". Below the status bar is the Windows taskbar with various application icons.

31.union of sets

```
def create_union_of_sets():
    set1 = {1, 2, 3, 4, 5}
    set2 = {3, 4, 5, 6, 7}
    union = set1 | set2
    print(union)
if __name__ == "__main__":
    create_union_of_sets()
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/union of sets.py =====
{1, 2, 3, 4, 5, 6, 7}
>>> |
```

The status bar at the bottom right indicates "Ln: 6 Col: 4", "2:35 PM", and the date "7/10/2023". Below the status bar is a taskbar with various icons, including the Python IDLE icon.

```
32.create set difference
def create_set_difference():
    set1 = {1, 2, 3, 4, 5}
    set2 = {3, 4, 5, 6, 7}
    difference = set1 - set2
    print(difference)
if __name__ == "__main__":
    create_set_difference()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays Python 3.8.7 code:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/create set difference.py =====
{1, 2}
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 2:38 PM, and 7/10/2023.

33.symmetric diff

```
def create_symmetric_difference():
    set1 = {1, 2, 3, 4, 5}
    set2 = {3, 4, 5, 6, 7}
    symmetric_difference = set1 ^ set2
    print(symmetric_difference)
if __name__ == "__main__":
    create_symmetric_difference()
```

IDLE Shell 3.8.7

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/symmetric difference.py =====
{1, 2, 6, 7}
>>> |
```

Ln: 6 Col: 4

2:40 PM 7/10/2023

34.shallow copy of sets

```
a = {"run", "fast"}
b = {"fast", "run"}
a = b.copy()
print(a)
```

IDLE Shell 3.8.7

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/shallow copy of sets.py =====
{'run', 'fast'}
>>>
```

Ln: 6 Col: 4

2:50 PM 7/10/2023

35.min and max in a set

```
def find_max_min_values(set1):
    max_value = max(set1)
    min_value = min(set1)
    return max_value,min_value
set1 = {10,5,8,15,3}
max_value, min_value = find_max_min_values(set1)
print("Maximum value:",max_value)
print("Minimum value:",min_value)
```

The screenshot shows the IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The shell window displays the Python version (3.8.7), build date (Dec 21 2020), and bitness (64 bit AMD64). It shows the command line starting with '>>>' and the output of the script, which prints 'Maximum value: 15' and 'Minimum value: 3'. The status bar at the bottom right indicates the line number (Ln: 7), column number (Col: 4), and current time (1:53 PM) and date (7/11/2023).

36.remove all elem in a set

```
def remove_all_elements(set1):
    set1.clear()
set1 = {1, 2, 3, 4, 5}
print("Original Set:", set1)
remove_all_elements(set1)
print("Set after removing all elements:", set1)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/remove all elem in a set.py =====
Original Set: {1, 2, 3, 4, 5}
Set after removing all elements: set()
>>> |
```

The status bar at the bottom right indicates Ln: 7 Col: 4, 1:55 PM, and 7/11/2023.

37.string to set

```
def convert_string_to_set(string):
    string_set = set(string)
    return string_set
string1 = "hello"
set1 = convert_string_to_set(string1)
print("String:", string1)
print("Set:", set1)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/string to set.py =====
String: hello
Set: {'l', 'h', 'e', 'o'}
>>>
```

The status bar at the bottom right shows "Ln: 7 Col: 4", "1:59 PM", and the date "7/11/2023". Below the status bar is a taskbar with various icons for Windows applications like File Explorer, Edge, and FileZilla.

38.set to string

```
def set_to_string(set1):
    string_list = list(map(str, set1))
    result_string = ', '.join(string_list)
    return result_string
set1 = {'apple','banana','orange'}
result = set_to_string(set1)
print(result)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/set to string.py =====
banana, orange, apple
>>>
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 2:03 PM, and 7/11/2023.

39.set to list

```
def set_to_list(set1):
    result_list = list(set1)
    return result_list
set1 = {'apple', 'banana', 'orange'}
result = set_to_list(set1)
print(result)
```

The screenshot shows the IDLE Shell 3.8.7 window. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/set to list.py =====
['orange', 'apple', 'banana']
>>>
```

The status bar at the bottom right shows Ln: 6 Col: 4, 2:05 PM, and 7/11/2023.

40.set to tuple

```
def set_to_tuple(set1):
    result_tuple = tuple(set1)
    return result_tuple
set1 = {'apple', 'banana', 'orange'}
result = set_to_tuple(set1)
print(result)
```

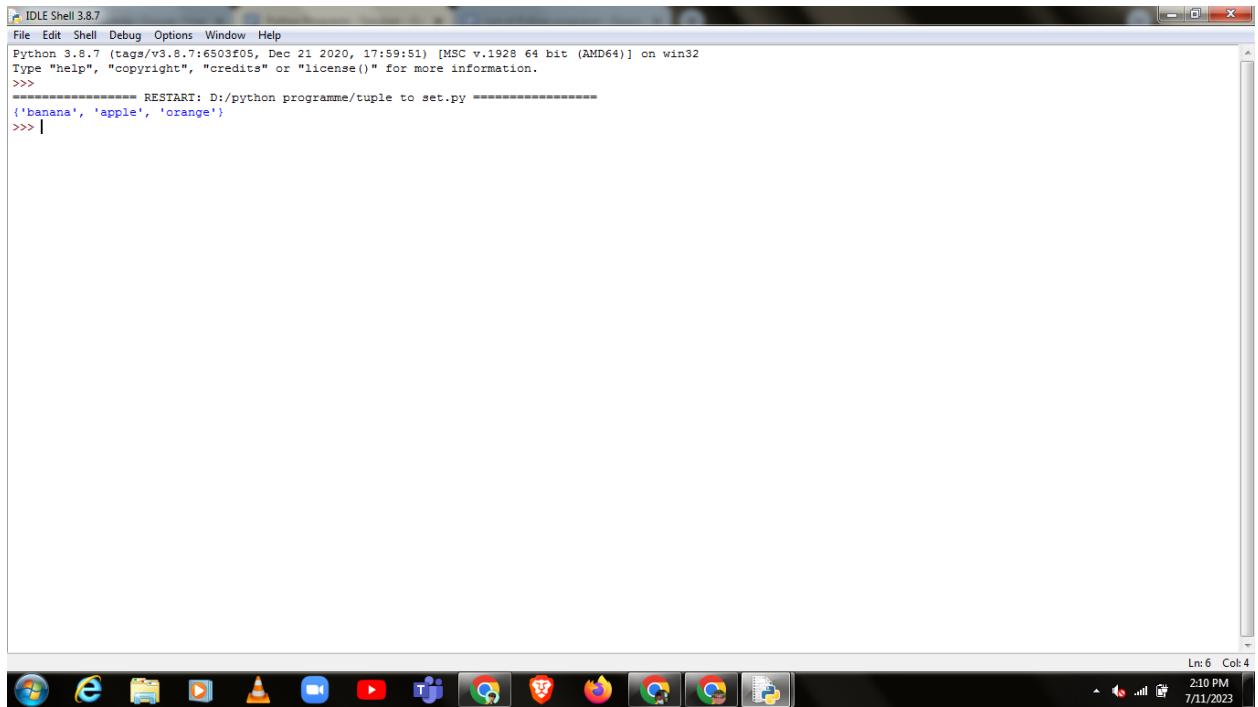
The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.8:653e3f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/set to tuple.py =====
('orange', 'apple', 'banana')
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:07 PM", and the date "7/11/2023". The taskbar at the bottom has icons for various Windows applications like File Explorer, Edge, and FileZilla.

41.tuple to set

```
def tuple_to_set(tuple1):
    result_set = set(tuple1)
    return result_set
tuple1 = ('apple', 'banana', 'orange')
result = tuple_to_set(tuple1)
print(result)
```



The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays Python 3.8.7 code:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/tuple to set.py =====
{'banana', 'apple', 'orange'}
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 2:10 PM, and 7/11/2023.

42.get unique items from two sets and return in new set

```
def get_unique_items(set1,set2):
    combined_set = set1.union(set2)
    return combined_set
set1 = {1, 2, 3, 4, 5}
set2 = {4, 5, 6, 7, 8}
result = get_unique_items(set1,set2)
print(result)
```



The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/get unique items from two sets.py =====
{1, 2, 3, 4, 5, 6, 7, 8}
>>> |
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 2:14 PM, and 7/11/2023.

43.sort a dictionary ascending and descending by value

```
def sort_dict_by_value(input_dict, ascending=True):
    sorted_dict = dict(sorted(input_dict.items(), key=lambda item: item[1], reverse=not
ascending))
    return sorted_dict
my_dict = {'apple': 10, 'banana': 5, 'orange': 8, 'grape': 3}
ascending_result = sort_dict_by_value(my_dict)
descending_result = sort_dict_by_value(my_dict, ascending=False)
print("Ascending order:")
for key, value in ascending_result.items():
    print(key, ":", value)
print("\nDescending order:")
for key, value in descending_result.items():
    print(key, ":", value)
```

The screenshot shows the Python IDLE Shell window. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The script content is as follows:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/sort ascending and descending a dictionary by value.py
Ascending order:
grape : 3
banana : 5
orange : 8
apple : 10

Descending order:
apple : 10
orange : 8
banana : 5
grape : 3
>>>
```

The status bar at the bottom right indicates "Ln: 16 Col: 4", "2:22 PM", and the date "7/11/2023".

44.add key to dictionary

```
def add_key_to_dict(input_dict,key, value):
    input_dict[key] = value
my_dict = {'apple': 10, 'banana': 5, 'orange': 8}
new_key = 'grape'
new_value = 3
add_key_to_dict(my_dict,new_key,new_value)
print(my_dict)
```



The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/add key to dictionary.py =====
{'apple': 10, 'banana': 5, 'orange': 8}
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:24 PM", and the date "7/11/2023". The taskbar at the bottom has icons for various Windows applications.

45.check a key exists in a dictionary or not

```
def check_key_existence(input_dict,key):
    if key in input_dict:
        return True
    else:
        return False
my_dict = {'apple': 10, 'banana': 5, 'orange': 8}
search_key = 'banana'
key_exists = check_key_existence(my_dict, search_key)
if key_exists:
    print(f"The key exists.")
else:
    print(f"The key does not exist.")
```



The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/check key exists or not in a dictionary.py ====
The key exists.
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:26 PM", and the date "7/11/2023". Below the status bar is a taskbar with various icons.

46.merge dictionary

```
def merge_dicts(dict1, dict2):
    merged_dict = {**dict1, **dict2}
    return merged_dict
dict1 = {'apple': 10, 'banana': 5}
dict2 = {'orange': 8, 'grape': 3}
merged_dict = merge_dicts(dict1, dict2)
print(merged_dict)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.5:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/merge dictionarys.py =====
{'apple': 10, 'banana': 5, 'orange': 8, 'grape': 3}
>>>
```

The status bar at the bottom right indicates "Ln: 6 Col: 4", "2:29 PM", and the date "7/11/2023".

47.sum of items is dictionary

```
def sum_dictionary_items(input_dict):
    total_sum = sum(input_dict.values())
    return total_sum
my_dict = {'apple': 10,'banana': 5,'orange': 8,'grape': 3}
sum_of_items = sum_dictionary_items(my_dict)
print("Sum =:", sum_of_items)
```



IDLE Shell 3.8.7

File Edit Shell Debug Options Window Help

Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.

>>> ===== RESTART: D:/python programme/sum of items in dictionary.py ======

Sum = : 26

>>>

Ln: 6 Col: 4

2:31 PM 7/11/2023

48.count list items

```
def count_list_items(dict_value):
    if isinstance(dict_value, list):
        return len(dict_value)
    else:
        return 0
my_dict = {'fruit':['apple','banana','orange'],'vegetable':['carrot','beans']}
for key, value in my_dict.items():
    count = count_list_items(value)
    print(f"Key: {key}, List count: {count}")
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/count list items in dictionary.py =====
Key: fruit, List count: 3
Key: vegetable, List count: 2
>>> |
```

The status bar at the bottom right indicates Ln: 7 Col: 4, 2:39 PM, and 7/11/2023.

49.check key existence in dictionary

```
def check_keys_existence(input_dict, keys):
    for key in keys:
        if key not in input_dict:
            return False
    return True

my_dict = {'apple': 10, 'banana': 5, 'orange': 8}
keys_to_check = ['apple', 'banana', 'grape']
keys_exist = check_keys_existence(my_dict, keys_to_check)
if keys_exist:
    print(" keys exist in the dictionary.")
else:
    print(" one key does not exist in the dictionary.")
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The text area displays the following Python code and its execution:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/check keys existence.py =====
one key does not exist in the dictionary.
>>> |
```

The status bar at the bottom right indicates "Ln: 6 Col: 4", "2:43 PM", and the date "7/11/2023". Below the status bar is the Windows taskbar with various application icons.

50.print dictionary line by line

```
def print_dictionary(dict_obj):
    for key, value in dict_obj.items():
        print(key,":",value)
my_dict = {'apple': 10, 'banana': 5, 'orange': 8}
print("Dictionary:")
print_dictionary(my_dict)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. A status bar at the bottom right shows "Ln: 9 Col: 4", "2:45 PM", and the date "7/11/2023". The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print dictionary line by line.py =====
Dictionary:
apple : 10
banana : 5
orange : 8
>>> |
```

51.two lists into dictionary

```
def lists_to_dictionary(keys, values):
    dictionary = {}
    for i in range(len(keys)):
        dictionary[keys[i]] = values[i]
    return dictionary
keys = ['name','age','work']
values = ['ramu',25,'Teacher']
result = lists_to_dictionary(keys, values)
print(result)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/2 lists into dictionary.py =====
{'name': 'ramu', 'age': 25, 'work': 'Teacher'}
>>> |
```

The status bar at the bottom right indicates "Ln: 6 Col: 4", "12:01 PM", and the date "7/12/2023". The taskbar at the bottom shows various application icons.

52.del list of keys from dictionary

```
def delete_keys(dictionary,keys):
    for key in keys:
        dictionary.pop(key)
my_dict = {'name': 'ramu', 'age': 25, 'city': 'chennai'}
keys_to_delete = ['age','city']
delete_keys(my_dict, keys_to_delete)
print(my_dict)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/del list of keys from dictionary.py =====
Traceback (most recent call last):
  File "D:/python programme/del list of keys from dictionary.py", line 6, in <module>
    delete_keys(my_dict, keys_to_delete)
  File "D:/python programme/del list of keys from dictionary.py", line 3, in delete_keys
    dictionary.pop(key)
NameError: name 'key' is not defined
>>> ===== RESTART: D:/python programme/del list of keys from dictionary.py =====
{'name': 'ramu'}
>>> |
```

The status bar at the bottom right shows "Ln:14 Col:4", "12:05 PM", and the date "7/12/2023". The taskbar at the bottom of the screen shows various application icons.

53.print only keys of a dict

```
def print_dictionary_keys(dictionary):
    for key in dictionary.keys():
        print(key)
my_dict = {'name': 'ram', 'age': 19, 'city': 'chennai'}
print_dictionary_keys(my_dict)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print only keys of a dict.py =====
name
age
city
>>>
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the output of a Python script named "print only keys of a dict.py". The script contains three key definitions: "name", "age", and "city". The command prompt is shown at the bottom with ">>>".

54.print values of a dict

```
def print_dictionary_values(dictionary):
    for value in dictionary.values():
        print(value)
my_dict = {'name': 'raj', 'age': 30, 'city': 'bangalore'}
print_dictionary_values(my_dict)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print values of a dict.py =====
raj
30
bangalore
>>> |
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the output of a Python script named "print values of a dict.py". The script defines a dictionary "my_dict" with keys "name", "age", and "city" and prints their values. The command prompt is shown at the bottom with ">>>".

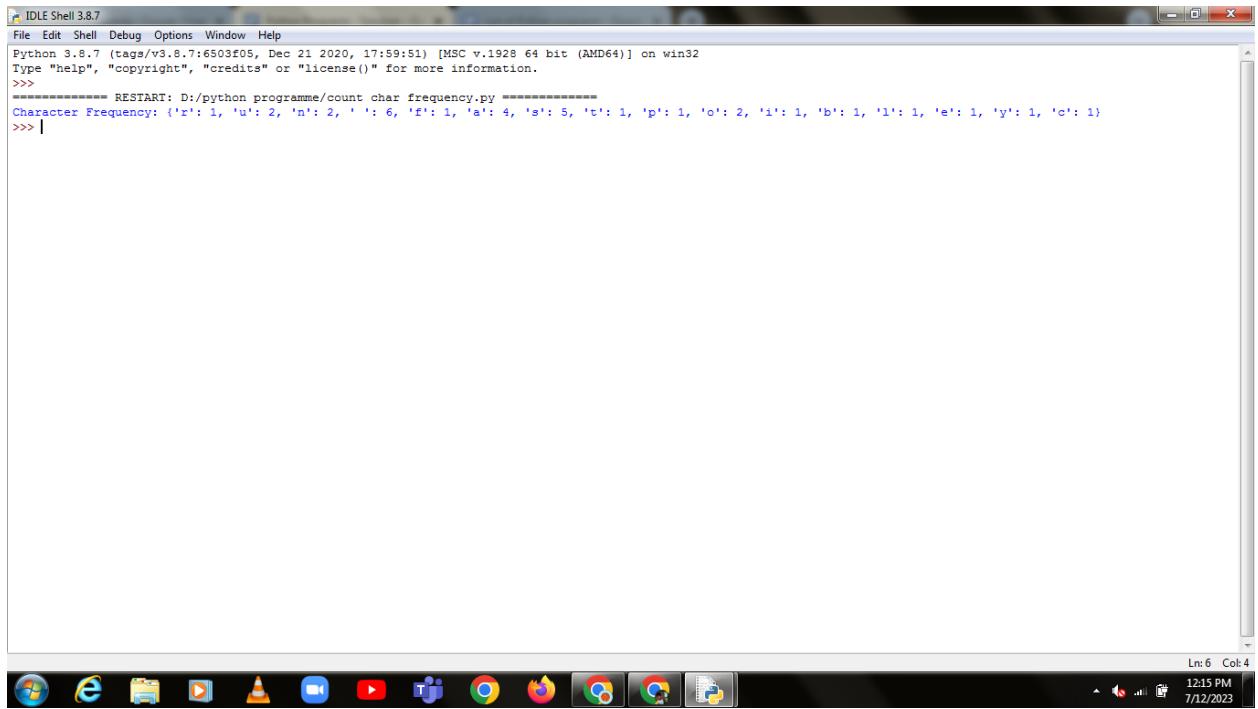
```
55.calc length of a string
def calc_string_length(string):
    length = len(string)
    return length
string1 = "Madras"
length = calc_string_length(string1)
print("Length of the string:", length)
```

The screenshot shows the IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays the following output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/calc len of a string.py =====
Length of the string: 6
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 12:12 PM, and 7/12/2023.

```
56.count char frequency
def count_char_frequency(string):
    frequency = {}
    for char in string:
        frequency[char] = frequency.get(char, 0) + 1
    return frequency
my_string = "run fast as possible as you can"
character_frequency = count_char_frequency(my_string)
print("Character Frequency:", character_frequency)
```



IDLE Shell 3.8.7

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/count char frequency.py =====
Character Frequency: {'r': 1, 'u': 2, 'n': 2, ' ': 6, 'f': 1, 'a': 4, 's': 5, 't': 1, 'p': 1, 'o': 2, 'i': 1, 'b': 1, 'l': 1, 'e': 1, 'y': 1, 'c': 1}
>>> |
```

Ln: 6 Col: 4
12:15 PM 7/12/2023

57.print floating num upto two decimal points

```
def print_float_numbers(numbers):
    for number in numbers:
        print("{:.2f}".format(number))
floating_numbers = [3.5867, 2.71828, 0.123456]
print_float_numbers(floating_numbers)
```

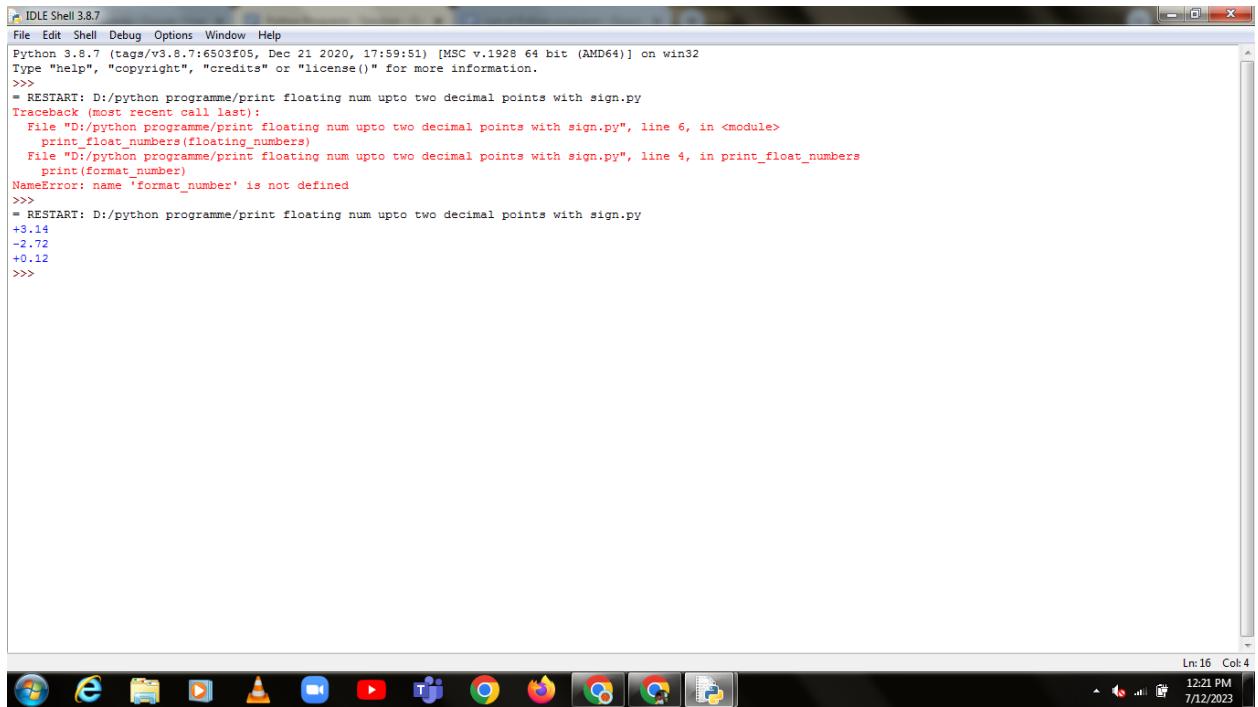
The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print float nums upto 2 deci points.py =====
3.59
2.72
0.12
>>>
```

The status bar at the bottom right shows "Ln: 8 Col: 4", "12:18 PM", and the date "7/12/2023".

58.print floating num upto two decimal points

```
def print_float_numbers(numbers):
    for number in numbers:
        format_number = "{:+.2f}".format(number)
        print(format_number)
floating_numbers = [3.14159, -2.71828, 0.123456]
print_float_numbers(floating_numbers)
```



The screenshot shows the Python IDLE Shell window with the following text:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/print floating num upto two decimal points with sign.py
Traceback (most recent call last):
  File "D:/python programme/print floating num upto two decimal points with sign.py", line 6, in <module>
    print_float_numbers(floating_numbers)
  File "D:/python programme/print floating num upto two decimal points with sign.py", line 4, in print_float_numbers
    print(format_number)
NameError: name 'format_number' is not defined
>>>
= RESTART: D:/python programme/print floating num upto two decimal points with sign.py
+3.14
-2.72
+0.12
>>>
```

The taskbar at the bottom of the screen shows various application icons, including a Python icon.

59.float nums without deci points

```
def print_float_numbers(numbers):
    for number in numbers:
        format_number = "{:.0f}".format(number)
        print(format_number)
floating_numbers = [3.675,8.656565,6.3345]
print_float_numbers(floating_numbers)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays Python version information and a session transcript:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/print float nums without deci points.py =====
4
9
6
>>>
```

The status bar at the bottom right shows Ln: 8 Col: 4, 12:23 PM, and 7/12/2023.

60.reverse string

```
def reverse_string(string):
    reversed_string = string[::-1]
    return reversed_string
my_string = "eatfast"
reversed_string = reverse_string(my_string)
print("Reversed string:", reversed_string)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/reverse string.py =====
Reversed string: tsaftae
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 12:25 PM, and 7/12/2023. The taskbar below the window contains icons for various Windows applications.

61.string to list of words

```
def string_to_word_list(string):
    word_list = string.split()
    return word_list
input_string = "which college are you studying?"
result = string_to_word_list(input_string)
print(result)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/string to word list.py =====
['which', 'college', 'are', 'you', 'studying?']
>>>
```

The status bar at the bottom right shows Ln: 6 Col: 4, 12:36 PM, and 7/13/2023.

62.swap comma dot in string

```
def swap_comma_dot(string):
    swapped_string = string.translate(str.maketrans(',','.',','))
    return swapped_string
input_string = "These are,some numbers 1,234.56"
result = swap_comma_dot(input_string)
print(result)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/swap dot and comma.py =====
These are some numbers 1.234,56
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 1:25 PM, and 7/13/2023.

63.count and display vowels

```
def count_and_display_vowels(text):
    vowels = ['a','e','i','o','u']
    vowel_count = 0
    vowel_list = []
    for char in text:
        if char.lower() in vowels:
            vowel_count += 1
            vowel_list.append(char)
    print("Total vowels:", vowel_count)
    print("Vowels:", ', '.join(vowel_list))
input_text = "Where do you live?"
count_and_display_vowels(input_text)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/count and display vowels.py =====
Total vowels: 7
Vowels: e, e, o, o, u, i, e
>>> |
```

The status bar at the bottom right indicates Ln: 7 Col: 4, 1:28 PM, and 7/13/2023.

64. find smallest and largest word in a string

```
def find_smallest_and_largest_word(string):
    words = string.split()
    smallest_word=min(words, key=len)
    largest_word=max(words, key=len)
    return smallest_word, largest_word
input_string = "where do you come from?"
smallest,largest = find_smallest_and_largest_word(input_string)
print("Smallest word:",smallest)
print("Largest word:",largest)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code area displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/find small and large word in a string.py =====
Smallest word: do
Largest word: where
>>> |
```

The status bar at the bottom right shows "Ln: 7 Col: 4", "1:32 PM", and the date "7/13/2023". The taskbar at the bottom has icons for various applications like File Explorer, Edge, and Python.

65.print four values decimal,octal,binary,hexa of integer

```
def print_integer_representation(num):
    decimal = str(num)
    octal = oct(num)
    hexadecimal = hex(num).upper()
    binary = bin(num)
    print("Decimal:", decimal,end=" ")
    print("Octal:", octal,end=" ")
    print("Hexadecimal:",hexadecimal,end=" ")
    print("Binary:", binary)
input_num = 42
print_integer_representation(input_num)
```

```

66.Days b/w 2 dates
from datetime import datetime
def get_days_between_dates(start_date, end_date):
    date_format = "%Y-%m-%d"
    start_datetime = datetime.strptime(start_date, date_format)
    end_datetime = datetime.strptime(end_date, date_format)
    diff = end_datetime - start_datetime
    num_days = diff.days
    return num_days
start_date = "2023-07-01"
end_date = "2023-07-13"
result = get_days_between_dates(start_date, end_date)
print("Number of days:", result)

```

The screenshot shows the IDLE Shell interface running on Windows. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the Python code and its output. The code calculates the number of days between "2023-07-01" and "2023-07-13", resulting in 12 days. The bottom status bar shows "Ln: 6 Col: 4", "1:42 PM", and the date "7/13/2023". The taskbar at the bottom of the screen shows various application icons.

```

File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/days between two dates.py =====
Number of days: 12
>>> |
```

```

67.print string ten times with two seconds time delay
import time
def print_string_with_delay(string, num_times):
    for i in range(num_times):
        print(string)
        time.sleep(2)
input_string = "How are you?"
num_times = 10
print_string_with_delay(input_string, num_times)

```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/print a string ten times with 2 seconds timedelay.py
How are you?
>>> |
```

Ln: 15 Col: 4
1:55 PM 7/13/2023

69.current date and time

```
from datetime import datetime
def get_current_datetime():
    current_datetime = datetime.now()
    return current_datetime
result = get_current_datetime()
print("Current Date and Time:", result)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/current date and time.py =====
Current Date and Time: 2023-07-13 14:34:07.645813
>>>
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:34 PM", and the date "7/13/2023".

70.date and time as string

```
from datetime import datetime
def generate_date_time_string():
    current_datetime = datetime.now()
    date_time_string = current_datetime.strftime("%Y-%m-%d %H:%M:%S")
    return date_time_string
result = generate_date_time_string()
print("Date and Time String:", result)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/date and time as string.py =====
Date and Time String: 2023-07-13 14:35:35
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "2:35 PM", and "7/13/2023". Below the status bar is a taskbar with various icons, including the Python IDLE icon.

```
71.print year calendar
import calendar
def print_year_calendar(year):
    for month in range(1,13):
        print(calendar.month_name[month], year)
        print(calendar.monthcalendar(year, month))
        print()
input_year = 2023
print_year_calendar(input_year)
```

The screenshot shows the Python IDLE Shell window with the title "IDLE Shell 3.8.7". The window displays the output of a script named "print_year_calendar.py". The output lists the days of each month from January 2023 to December 2023. The months are labeled in blue, and the days are listed in brackets. The script uses the `calendar` module to generate the calendar. The bottom status bar shows the date as 7/13/2023, time as 2:51 PM, and file path as D:/python programme/print year calendar.py.

```
>>> ===== RESTART: D:/python programme/print year calendar.py ======January 2023 [[0, 0, 0, 0, 0, 1], [2, 3, 4, 5, 6, 7, 8], [9, 10, 11, 12, 13, 14, 15], [16, 17, 18, 19, 20, 21, 22], [23, 24, 25, 26, 27, 28, 29], [30, 31, 0, 0, 0, 0, 0]]February 2023 [[0, 0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17, 18, 19], [20, 21, 22, 23, 24, 25, 26], [27, 28, 0, 0, 0, 0, 0]]March 2023 [[0, 0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17, 18, 19], [20, 21, 22, 23, 24, 25, 26], [27, 28, 29, 30, 31, 0, 0]]April 2023 [[0, 0, 0, 0, 1, 2], [3, 4, 5, 6, 7, 8, 9], [10, 11, 12, 13, 14, 15, 16], [17, 18, 19, 20, 21, 22, 23], [24, 25, 26, 27, 28, 29, 30]]May 2023 [[1, 2, 3, 4, 5, 6, 7], [8, 9, 10, 11, 12, 13, 14], [15, 16, 17, 18, 19, 20, 21], [22, 23, 24, 25, 26, 27, 28], [29, 30, 31, 0, 0, 0, 0]]June 2023 [[0, 0, 0, 1, 2, 3, 4], [5, 6, 7, 8, 9, 10, 11], [12, 13, 14, 15, 16, 17, 18], [19, 20, 21, 22, 23, 24, 25], [26, 27, 28, 29, 30, 0, 0]]July 2023 [[0, 0, 0, 0, 0, 1, 2], [3, 4, 5, 6, 7, 8, 9], [10, 11, 12, 13, 14, 15, 16], [17, 18, 19, 20, 21, 22, 23], [24, 25, 26, 27, 28, 29, 30], [31, 0, 0, 0, 0, 0, 0]]August 2023 [[0, 1, 2, 3, 4, 5, 6], [7, 8, 9, 10, 11, 12, 13], [14, 15, 16, 17, 18, 19, 20], [21, 22, 23, 24, 25, 26, 27], [28, 29, 30, 31, 0, 0, 0]]September 2023 [[0, 0, 0, 0, 1, 2, 3], [4, 5, 6, 7, 8, 9, 10], [11, 12, 13, 14, 15, 16, 17], [18, 19, 20, 21, 22, 23, 24], [25, 26, 27, 28, 29, 30, 0]]October 2023 [[0, 0, 0, 0, 0, 0, 1], [2, 3, 4, 5, 6, 7, 8], [9, 10, 11, 12, 13, 14, 15], [16, 17, 18, 19, 20, 21, 22], [23, 24, 25, 26, 27, 28, 29], [30, 31, 0, 0, 0, 0, 0]]November 2023 [[0, 0, 1, 2, 3, 4, 5], [6, 7, 8, 9, 10, 11, 12], [13, 14, 15, 16, 17, 18, 19], [20, 21, 22, 23, 24, 25, 26], [27, 28, 29, 30, 0, 0, 0]]December 2023 [[0, 0, 0, 0, 1, 2, 3], [4, 5, 6, 7, 8, 9, 10], [11, 12, 13, 14, 15, 16, 17], [18, 19, 20, 21, 22, 23, 24], [25, 26, 27, 28, 29, 30, 31]]>>>
```

72.print month calendar

```
import calendar
```

```
def print_month_calendar(year, month):
```

```
    print(calendar.month_name[month], year)
```

```
    print(calendar.monthcalendar(year, month))
```

```
    print()
```

```
input_year = 2023
```

```
input_month = 4
```

```
print_month_calendar(input_year, input_month)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print montly calendar.py =====
April 2023
[[0, 0, 0, 0, 1, 2], [3, 4, 5, 6, 7, 8, 9], [10, 11, 12, 13, 14, 15, 16], [17, 18, 19, 20, 21, 22, 23], [24, 25, 26, 27, 28, 29, 30]]
>>> |
```

73.dates b/w two dates

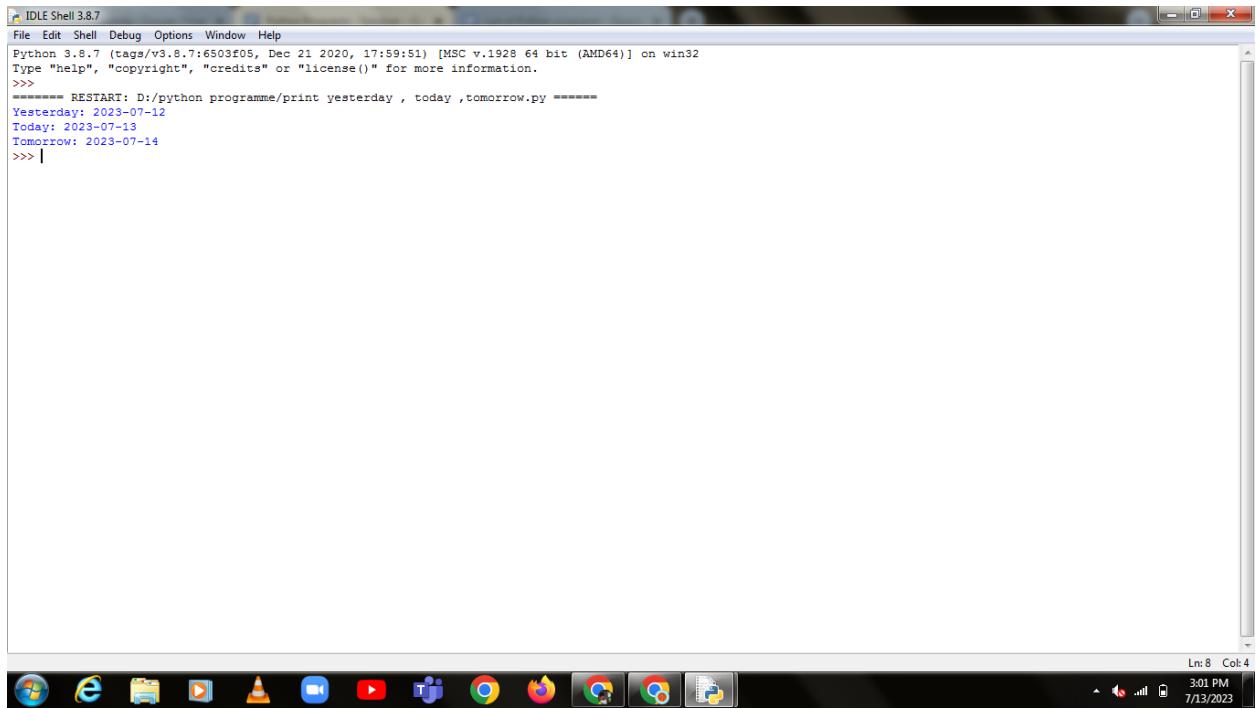
```
from datetime import datetime, timedelta
def get_dates_between(start_date, end_date):
    date_format = "%Y-%m-%d"
    dates = []
    start_datetime = datetime.strptime(start_date, date_format)
    end_datetime = datetime.strptime(end_date, date_format)
    current_date = start_datetime
    while current_date <= end_datetime:
        dates.append(current_date.strftime(date_format))
        current_date += timedelta(days=1)
    return dates
start_date = "2023-07-10"
end_date = "2023-07-15"
result = get_dates_between(start_date, end_date)
print("Dates between", start_date, "and", end_date, ":", result)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/dates between two dates.py =====
Dates between 2023-07-10 and 2023-07-15 : ['2023-07-10', '2023-07-11', '2023-07-12', '2023-07-13', '2023-07-14', '2023-07-15']
>>>
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 2:58 PM, and 7/13/2023.

```
75.print yesterday today tomorrow
from datetime import datetime, timedelta
def print_dates():
    today = datetime.today()
    yesterday = today - timedelta(days=1)
    tomorrow = today + timedelta(days=1)
    print("Yesterday:", yesterday.date())
    print("Today:", today.date())
    print("Tomorrow:", tomorrow.date())
print_dates()
```



The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print yesterday , today ,tomorrow.py =====
Yesterday: 2023-07-12
Today: 2023-07-13
Tomorrow: 2023-07-14
>>> |
```

The status bar at the bottom right shows Ln: 8 Col: 4, 3:01 PM, and 7/13/2023.

76.print next 5 days from today

```
from datetime import datetime, timedelta
```

```
def print_next_five_days():
```

```
    current_date = datetime.today()
```

```
    for i in range(5):
```

```
        next_date = current_date + timedelta(days=i)
```

```
        print(next_date.date())
```

```
print_next_five_days()
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print next 5 days.py =====
2023-07-13
2023-07-14
2023-07-15
2023-07-16
2023-07-17
>>>
```

The status bar at the bottom right shows "Ln: 10 Col: 4", "3:02 PM", and the date "7/13/2023".

```
77.add 5 seconds to current time
from datetime import datetime, timedelta
def add_seconds_to_current_time(seconds):
    current_time = datetime.now().time()
    current_datetime = datetime.combine(datetime.today(), current_time)
    new_datetime = current_datetime + timedelta(seconds=seconds)
    new_time = new_datetime.time()
    return new_time
seconds_to_add = 5
result = add_seconds_to_current_time(seconds_to_add)
print("New Time:", result)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/add 5 sec to current time.py =====
New Time: 15:04:32.912571
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:04 PM", and the date "7/13/2023". The taskbar at the bottom has icons for various applications, and the system tray shows battery and signal status.

```
78.get current seconds in milliseconds
import time
def get_current_time_milliseconds():
    current_time = time.time()
    current_time_milliseconds = int(current_time * 1000)
    return current_time_milliseconds
result = get_current_time_milliseconds()
print("Current Time in Milliseconds:", result)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/get current time in milliseconds.py =====
>>> Current Time in Milliseconds: 1689241239560
>>> |
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 3:11 PM, and 7/13/2023.

```
79.del 3 days from current days
from datetime import datetime, timedelta
def subtract_three_days():
    current_date = datetime.today()
    new_date = current_date - timedelta(days=3)
    return new_date
result = subtract_three_days()
print("New Date:", result.date())
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/del 3 days from current date.py =====
>>> New Date: 2023-07-10
>>> |
```

The status bar at the bottom right indicates "Ln: 6 Col: 4", "3:15 PM", and the date "7/13/2023". Below the status bar is the Windows taskbar with various icons.

80.check leap year or not

```
def is_leap_year(year):
    if year % 4 == 0:
        if year % 100 == 0:
            if year % 400 == 0:
                return True
            else:
                return False
        else:
            return True
    else:
        return False
input_year = 2024
result = is_leap_year(input_year)
if result:
    print(input_year, "is a leap year.")
else:
    print(input_year, "is not a leap year.")
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/check leap year or not.py =====
2024 is a leap year.
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:19 PM", and the date "7/13/2023". Below the status bar is a taskbar with various icons.

81.convert degree to radians

```
import math
def convert_degrees_to_radians(degrees):
    radians = math.radians(degrees)
    return radians
input_degrees = 90
result = convert_degrees_to_radians(input_degrees)
print("Radians:", result)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/convert deg to radians.py =====
Radians: 1.5707963267948966
>>> |
```

Ln: 6 Col: 4
3:24 PM
7/13/2023

82.radian to degree

```
import math
def convert_radians_to_degrees(radians):
    degrees = math.degrees(radians)
    return degrees
input_radians = 1.70679
result = convert_radians_to_degrees(input_radians)
print("Degrees:", result)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The shell area displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/radians to degree.py =====
Degrees: 97.79186351513378
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:25 PM", and the date "7/13/2023". The taskbar at the bottom has icons for various Windows applications.

83.area of trapezoid

```
def calculate_trapezoid_area(a, b, h):
    area = ((a+b)*h)/2
    return area
side_a = 5
side_b = 7
height = 4
result = calculate_trapezoid_area(side_a, side_b, height)
print("Area:", result)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays Python version information and a command-line session. The session starts with a restart of the trapezoid.py program, followed by the output 'Area: 24.0'. The status bar at the bottom right shows 'Ln: 6 Col: 4' and the system date and time as '3:27 PM 7/13/2023'. The taskbar at the bottom features icons for various Windows applications.

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/area of trapezoid.py =====
Area: 24.0
>>>
```

84.area of parallelogram

```
def calculate_parallelgram_area(base,height):
    area = base * height
    return area
base_length = 7
height = 4
result = calculate_parallelgram_area(base_length, height)
print("Area", result)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/area of parallelogram.py =====
Area 28
>>> |
```

The status bar at the bottom right shows "Ln: 6 Col: 4", "3:28 PM", and "7/13/2023". Below the status bar is the Windows taskbar with various application icons.

85.surface area and vol of cylinder

```
import math
def calc_cylinder_surface_area(radius, height):
    surface_area = 2*math.pi*radius**2 + 2*math.pi*radius*height
    return surface_area
def calc_cylinder_volume(radius, height):
    volume = math.pi*radius**2*height
    return volume
cylinder_radius = 5
cylinder_height = 10
surface_area = calc_cylinder_surface_area(cylinder_radius, cylinder_height)
volume = calc_cylinder_volume(cylinder_radius, cylinder_height)
print("Surface Area:", surface_area)
print("Volume:", volume)
```



The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The Python version is 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32. The command prompt shows the path D:/python programme/surface area and volume of cylinder.py and the output Surface Area: 471.23889803846896 and Volume: 785.3981633974483. The status bar at the bottom right indicates Ln: 7 Col: 4, 3:31 PM, and 7/13/2023.

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: D:/python programme/surface area and volume of cylinder.py =====
Surface Area: 471.23889803846896
Volume: 785.3981633974483
>>>
```

86.surface area and vol of a sphere

```
import math
def calc_sphere_volume(radius):
    volume = (4/3)*math.pi*radius**3
    return volume
def calc_sphere_area(radius):
    area = 4*math.pi*radius**2
    return area
def main():
    radius = float(input("Enter the radius of the sphere: "))
    volume = calc_sphere_volume(radius)
    area = calc_sphere_area(radius)
    print(f"The volume of the sphere is: {volume:.2f}")
    print(f"The surface area of the sphere is: {area:.2f}")
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The Python version is 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32. The script code calculates the volume and surface area of a sphere based on user input for the radius. The output shows the volume as 268.08 and the surface area as 201.06.

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/surface area and vol of sphere.py =====
Enter the radius of the sphere: 4
The volume of the sphere is: 268.08
The surface area of the sphere is: 201.06
>>> |
```

87.calculate arc length of an angle

```
import math
def calculate_arc_length(radius, angle):
    arc_length = (angle/360)*(2*math.pi*radius)
    return arc_length
def main():
    radius = float(input("Enter the radius: "))
    angle = float(input("Enter the angle: "))
    arc_length = calculate_arc_length(radius, angle)
    print(f"The arc length is: {arc_length:.2f}")
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell window. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/arc length.py =====
Enter the radius: 4
Enter the angle: 45
The arc length is: 3.14
>>>
```

The status bar at the bottom right shows "Ln: 8 Col: 4", "3:47 PM", and the date "7/17/2023".

88.calculate discriminant value

```
def calculate_discriminant(a, b, c):
    discriminant = b**2-4*a*c
    return discriminant
def main():
    a = float(input("Enter the coefficient of x^2: "))
    b = float(input("Enter the coefficient of x: "))
    c = float(input("Enter the constant term: "))
    discriminant = calculate_discriminant(a, b, c)
    print(f"The discriminant value is: {discriminant:.2f}")
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The Python version is 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32. The shell window displays the following code and output:

```
>>> ===== RESTART: D:/python programme/calc discriminant.py =====
Enter the coefficient of x^2: 4
Enter the coefficient of x: 5
Enter the constant term: 6
The discriminant value is: -71.00
>>> |
```

The status bar at the bottom right shows Ln: 9 Col: 4, 3:50 PM, and 7/17/2023.

90.sum of all divisors of a num

```
def sum_of_divisors(number):
    divisor_sum = 0
    for i in range(1, number + 1):
        if number % i == 0:
            divisor_sum += i
    return divisor_sum

def main():
    number = int(input("Enter a num: "))
    divisor_sum = sum_of_divisors(number)
    print(f"The sum of all divisors is: {divisor_sum}")
if __name__ == "__main__":
    main()
```

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/sum of divisors of a num.py =====
Enter a num: 56
The sum of all divisors is: 120
>>>
```

91.area of regular polygon

```
import math
def calculate_polygon_area(n, s):
    area = (0.25*n*s**2)/math.tan(math.pi/n)
    return area
def main():
    n = int(input("Enter the number of sides: "))
    s = float(input("Enter the length of each side: "))
    area = calculate_polygon_area(n, s)
    print(f"The area of the regular polygon is: {area:.2f}")
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its execution output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/area of regular polygon.py =====
Enter the number of sides: 5
Enter the length of each side: 6
The area of the regular polygon is: 61.94
>>>
```

The status bar at the bottom right shows "Ln: 8 Col: 4", "3:55 PM", and the date "7/17/2023".

92.get square root and expo of a deci num

```
import math
def calc_square_root(number):
    return math.sqrt(number)
def calc_exponential(number):
    return math.exp(number)
decimal_number = float(input("Enter a decimal number: "))
square_root = calc_square_root(decimal_number)
exponential = calc_exponential(decimal_number)
print("Square root:", square_root)
print("Exponential:", exponential)
```



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/calc square root and expo of a deci num.py ====
Enter a decimal number: 36.66
Square root: 6.0547502012882415
Exponential: 8341337749180662.0
>>> |
```

Ln: 8 Col: 4
10:55 AM 7/18/2023

93.generate random float num between a numerical range

```
import random
def generate_random_float(start, end):
    return random.uniform(start, end)
start_range = float(input("Enter the start of the range: "))
end_range = float(input("Enter the end of the range: "))
random_float = generate_random_float(start_range, end_range)
print("Random float number:", random_float)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ---- RESTART: D:/python programme/generate random float nums between points.py --
Enter the start of the range: 1.2
Enter the end of the range: 7.8
Random float number: 6.415585325208192
>>>
```

The status bar at the bottom right shows Ln: 8 Col: 4, 11:19 AM, and 7/18/2023.

95.generate a int between a numerical range

```
import random
def generate_random_integer(start, end):
    return random.randint(start, end - 1)
start_range = int(input("Enter the start of the range: "))
end_range = int(input("Enter the end of the range: "))
random_integer = generate_random_integer(start_range, end_range)
print("Random integer:", random_integer)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
==== RESTART: D:/python programme/generate random int nums between a range.py ====
Enter the start of the range: 2
Enter the end of the range: 9
Random integer: 3
>>>
```

The status bar at the bottom right indicates Ln: 8 Col: 4, 11:22 AM, and 7/18/2023.

```
96.generate random eve int in range
import random
def generate_random_even(start, end):
    if start % 2!= 0:
        start += 1
    return random.randrange(start,end,2)
start_range = int(input("Enter the start of the range: "))
end_range = int(input("Enter the end of the range: "))
random_even = generate_random_even(start_range, end_range)
print("Random even integer:", random_even)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The Python version is 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32. The command line shows a script named 'generate random even int in a range.py' being run. The user enters the start of the range as 2 and the end of the range as 98. The script outputs a random even integer, 32. The status bar at the bottom right indicates Ln: 8 Col: 4, 11:24 AM, and 7/18/2023.

```
>>> ====== RESTART: D:/python programme/generate random even int in a range.py ======
Enter the start of the range: 2
Enter the end of the range: 98
Random even integer: 32
>>>
```

97. flip a coin 1000 times and count heads and tails

```
import random
def flip_coin():
    return random.choice(['Heads','Tails'])
heads_count = 0
tails_count = 0
for _ in range(1000):
    result = flip_coin()
    if result == 'Heads':
        heads_count += 1
    else:
        tails_count += 1
print("Number of Heads:", heads_count)
print("Number of Tails:", tails_count)
```

The screenshot shows the Python IDLE Shell interface. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/count heads and tails after 1000 flips of coin.py
Number of Heads: 481
Number of Tails: 519
>>> |
```

The taskbar at the bottom shows various application icons, and the system tray indicates the date and time as 11:51 AM on 7/18/2023.

98.randomly select item from list

```
import random
def random_item_from_list(items):
    return random.choice(items)
my_list = ['apple','banana','orange','grape','watermelon']
random_item = random_item_from_list(my_list)
print("Random item:", random_item)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The status bar at the bottom right shows "Ln: 6 Col: 4", "11:53 AM", and "7/18/2023". The main area displays the following Python code and its output:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/randomly select item from list.py =====
>>> Random item: apple
```

99.celsius to fahrenheit

```
def celsius_to_fahrenheit(celsius):
    fahrenheit = (celsius * 9/5) + 32
    return fahrenheit
celsius_temp = float(input("Enter the temp in Celsius: "))
fahrenheit_temp = celsius_to_fahrenheit(celsius_temp)
print("Temperature in Fahrenheit:", fahrenheit_temp)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/celsius to farenheit.py =====
Enter the temp in Celsius: 50
Temperature in Fahrenheit: 122.0
>>>
```

```
100.area of triangle
import math
def calculate_area_triangle(a, b, c):
    s = (a + b + c) / 2
    area = math.sqrt(s * (s - a) * (s - b) * (s - c))
    return area
side_a = float(input("Enter the length of side a: "))
side_b = float(input("Enter the length of side b: "))
side_c = float(input("Enter the length of side c: "))
triangle_area = calculate_area_triangle(side_a, side_b, side_c)
print("Area:", triangle_area)
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/area of triangle.py =====
Enter the length of side a: 4
Enter the length of side b: 5
Enter the length of side c: 6
Area: 9.921567416492215
>>>
```

The status bar at the bottom right shows Ln: 9 Col: 4, 12:16 PM, and 7/18/2023.

101. average of integers in a list

```
def calculate_average(numbers):
    total = sum(numbers)
    average = total / len(numbers)
    return average
numbers_list = []
n = int(input("Enter the number of integers: "))
for i in range(n):
    number = int(input(f'Enter integer {i+1}: '))
    numbers_list.append(number)
average_value = calculate_average(numbers_list)
print("Average:", average_value)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays a Python script named avg of integers in a list.py. The script prompts the user for the number of integers (4), then for four integer values (2, 3, 5, 6) and calculates their average (4.0). The bottom status bar shows Ln: 11 Col: 4, 12:19 PM, and 7/18/2023. The taskbar at the bottom has icons for various Windows applications.

```
>>> ===== RESTART: D:/python programme/avg of integers in a list.py =====
Enter the number of integers: 4
Enter integer 1: 2
Enter integer 2: 3
Enter integer 3: 5
Enter integer 4: 6
Average: 4.0
>>>
```

102.product of real nums in a set

```
def calculate_product(numbers):
    product = 1
    for number in numbers:
        product *= number
    return product
numbers_list = []
n = int(input("Enter the no of real numbers: "))
for i in range(n):
    number = float(input(f"Enter real number {i+1}: "))
    numbers_list.append(number)
product_value = calculate_product(numbers_list)
print("Product:", product_value)
```



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/product of real num in a list.py =====
Enter the no of real numbers: 4
Enter real number 1: 2
Enter real number 2: 3
Enter real number 3: 5
Enter real number 4: 6
Product: 180.0
>>> |
```

Ln: 11 Col: 4
12:20 PM 7/18/2023

103.area and circumference of a circle

```
import math
def calculate_circumference(radius):
    circumference = 2 * math.pi * radius
    return circumference
def calculate_area(radius):
    area = math.pi * radius**2
    return area
radius = float(input("Enter the radius of the circle: "))
circumference = calculate_circumference(radius)
area = calculate_area(radius)
print("Circumference:", circumference)
print("Area:", area)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The Python version is 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32. The command line shows a script named 'area and circumference of a circle.py' being run. The user enters a radius of 4, and the script calculates the circumference (25.132741228718345) and area (50.26548245743669). The status bar at the bottom right indicates Ln: 8 Col: 4, 12:22 PM, and 7/18/2023.

```
>>> ===== RESTART: D:/python programme/area and circumference of a circle.py =====
Enter the radius of the circle: 4
Circumference: 25.132741228718345
Area: 50.26548245743669
>>> |
```

104.check given integer is multiple of 5 or not

```
def check_multiple_of_five(number):
    if number % 5 == 0:
        return True
    else:
        return False
integer_number = int(input("Enter an integer: "))
is_multiple_of_five = check_multiple_of_five(integer_number)
if is_multiple_of_five:
    print("The number is a multiple of 5.")
else:
    print("The number is not a multiple of 5.")
```

The screenshot shows the Python IDLE Shell interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/check number multiple of 5.py =====
Enter an integer: 35
The number is a multiple of 5.
>>> |
```

The status bar at the bottom right shows Ln: 7 Col: 4, 12:24 PM, and 7/18/2023.

105.check the given integer is multiple of 5 and 7

```
def check_multiple_of_five_and_seven(number):
    if number % 5 == 0 and number % 7 == 0:
        return True
    else:
        return False
integer_number = int(input("Enter an integer number: "))
is_multiple_of_five_and_seven = check_multiple_of_five_and_seven(integer_number)
if is_multiple_of_five_and_seven:
    print("The number is a multiple of both 5 and 7.")
else:
    print("The number is not a multiple of both 5 and 7.)
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/check multiple of 5 and 7.py =====
Enter an integer number: 35
The number is a multiple of both 5 and 7.
>>>
```

The status bar at the bottom right shows Ln: 7 Col: 4, 12:26 PM, and 7/18/2023.

```
106.avg of 10 nums using while loop
total_sum = 0
count = 0
while count < 10:
    number = float(input("Enter a number: "))
    total_sum += number
    count += 1
average = total_sum / 10
print("Avg of the 10 numbers:", average)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/avg of 10 num using while loop.py =====
Enter a number: 70
Enter a number: 45
Enter a number: 43
Enter a number: 45
Enter a number: 56
Enter a number: 54
Enter a number: 32
Enter a number: 45
Enter a number: 67
Enter a number: 65
Avg of the 10 numbers: 52.2
>>> |
```

107.multiples of 3 within range 10 to 50

start = 10

end = 50

```
print("Multiples of 3:")
for number in range(start, end+1):
    if number % 3 == 0:
        print(number)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/multiples of 3 within range 10 to 50.py =====
Multiples of 3:
12
15
18
21
24
27
30
33
36
39
42
45
48
>>> |
```

108.all integers within the range 100 -200 whose sum of digits is an even number

start = 100

end = 200

```
print("Integers within the range 100 to 200 with an even sum of digits:")
```

```
for number in range(start, end+1):
```

```
    digit_sum = sum(int(digit) for digit in str(number))
```

```
    if digit_sum % 2 == 0:
```

```
        print(number)
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/all integers within the range 100 -200 whose sum of digits is an even number.py
Integers within the range 100 to 200 with an even sum of digits:
101
103
105
107
109
110
112
114
116
118
121
123
125
127
129
130
132
134
136
138
141
143
145
147
149
150
152
154
156
158
161
163
165
167
169
170
...
Ln: 19 Col: 3
12:32 PM 7/18/2023
```

109.print num from n to 0

```
def print_numbers_till_zero(n):
```

```
    if n == 0:
```

```
        print(n)
```

```
    else:
```

```
        print(n)
```

```
        print_numbers_till_zero(n - 1)
```

```
def main():
```

```
    n = int(input("Enter a number: "))
```

```
    print_numbers_till_zero(n)
```

```
if __name__ == "__main__":
```

```
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code in the shell window is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/print num from n to 0.py =====
Enter a number: 9
9
8
7
6
5
4
3
2
1
0
>>> |
```

The status bar at the bottom right indicates Ln: 16 Col: 4, 12:23 PM, and 7/21/2023.

110.linear search

```
def linear_search(list, item):
    for i in range(len(list)):
        if list[i] == item:
            return i
    else:
        return -1
def main():
    list = [1, 2, 3, 4, 5]
    item = 3
    index = linear_search(list, item)
    if index == -1:
        print(f"The item {item} was not found in the list.")
    else:
        print(f"The item {item} was found at index {index}.")
if __name__ == "__main__":
    main()
```

```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/linear search.py =====
The item 3 was found at index 2.
>>>
```

111.binary search

```
def binary_search(list, item):
    low = 0
    high = len(list) - 1
    while low <= high:
        mid = (low + high) // 2
        if list[mid] == item:
            return mid
        elif list[mid] < item:
            low = mid + 1
        else:
            high = mid - 1
    else:
        return -1
def main():
    list = [1, 2, 3, 4, 5]
    item = 3
    index = binary_search(list, item)
    if index == -1:
        print(f"The item {item} was not found in the list.")
    else:
        print(f"The item {item} was found at index {index}.")
```

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



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
=====
RESTART: D:/python programme/binary search.py =====
The item 3 was found at index 2.
>>> |
```

112.largest num in list without using built in func

```
def find_largest_number(list):
    largest_number = list[0]
    for number in list[1:]:
        if number > largest_number:
            largest_number = number
    return largest_number
def main():
    list = [1, 2, 3, 4, 5]
    largest_number = find_largest_number(list)
    print(f"The largest number in the list is {largest_number}")
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python session:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:/python programme/largest num in list without using built in func.py
The largest number in the list is 5
>>> |
```

The status bar at the bottom right shows Ln: 6 Col: 4, 12:29 PM, and 7/21/2023.

115.insert num in any pos in list

```
def insert_number(list, number, position):
    new_list = list[:position]
    new_list.append(number)
    new_list += list[position:]
    return new_list

def main():
    list = [1, 2, 3, 4, 5]
    number = 6
    position = 2
    new_list = insert_number(list, number, position)
    print(f"The new list is {new_list}")
if __name__ == "__main__":
    main()
```

The screenshot shows the IDLE Shell 3.8.7 interface. The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The window title is "IDLE Shell 3.8.7". The code being run is:

```
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/insert num in specific pos in array.py =====
The new list is [1, 2, 6, 3, 4, 5]
>>> |
```

The status bar at the bottom right indicates Ln: 6 Col: 4, 12:32 PM, and 7/21/2023.

116.matrix addition

```
def matrix_addition(matrix1, matrix2):
    new_matrix = []
    for i in range(len(matrix1)):
        row = []
        for j in range(len(matrix1[0])):
            row.append(matrix1[i][j] + matrix2[i][j])
        new_matrix.append(row)
    return new_matrix

def main():
    matrix1 = [[1, 2, 3], [4, 5, 6]]
    matrix2 = [[7, 8, 9], [10, 11, 12]]
    new_matrix = matrix_addition(matrix1, matrix2)
    print(f"The sum of the two matrices is: \n{new_matrix}")
if __name__ == "__main__":
    main()
```

The screenshot shows the Python IDLE Shell window. The title bar reads "IDLE Shell 3.8.7". The menu bar includes File, Edit, Shell, Debug, Options, Window, and Help. The main window displays the following Python code and its output:

```
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/matrix addition.py =====
The sum of the two matrices is:
[[8, 10, 12], [14, 16, 18]]
>>>
```

The status bar at the bottom right shows "Ln: 7 Col: 4", "12:34 PM", and "7/21/2023".

117.simple calculator

```
def calculator():
    print("Select operation.")
    print("1.Addition")
    print("2.Subtraction")
    print("3.Multiplication")
    print("4.Division")
    operation = input("Enter operation: ")
    if operation == "1":
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
        print(f"{num1} + {num2} = {num1 + num2}")
    elif operation == "2":
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
        print(f"{num1} - {num2} = {num1 - num2}")
    elif operation == "3":
        num1 = float(input("Enter first number: "))
        num2 = float(input("Enter second number: "))
        print(f"{num1} * {num2} = {num1 * num2}")
    elif operation == "4":
        num1 = float(input("Enter first number: "))
```

```
num2 = float(input("Enter second number: "))
print(f"{num1} / {num2} = {num1 / num2}")
else:
    print("Invalid operation.")
if __name__ == "__main__":
    calculator()
```



```
IDLE Shell 3.8.7
File Edit Shell Debug Options Window Help
Python 3.8.7 (tags/v3.8.7:6503f05, Dec 21 2020, 17:59:51) [MSC v.1928 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> ===== RESTART: D:/python programme/simple calculator.py =====
Select operation.
1.Addition
2.Subtraction
3.Multiplication
4.Division
Enter operation: 1
Enter first number: 2
Enter second number: 3
2.0 + 3.0 = 5.0
>>> |
```

119.

