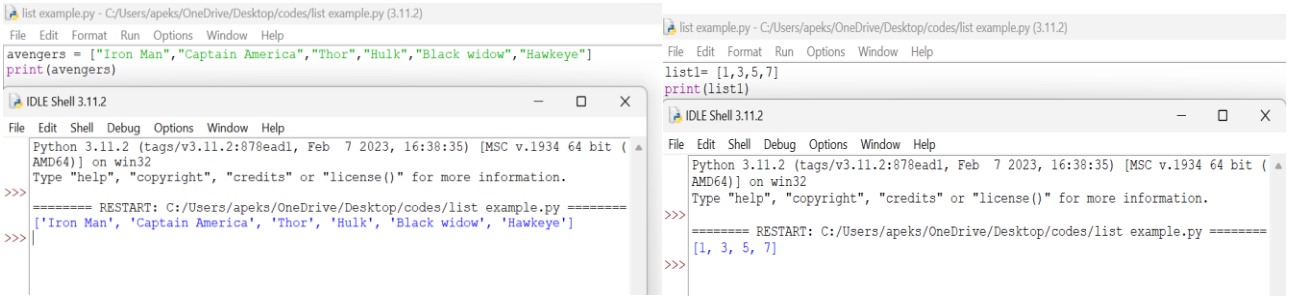
Week 6: Workshop

Exercise 1

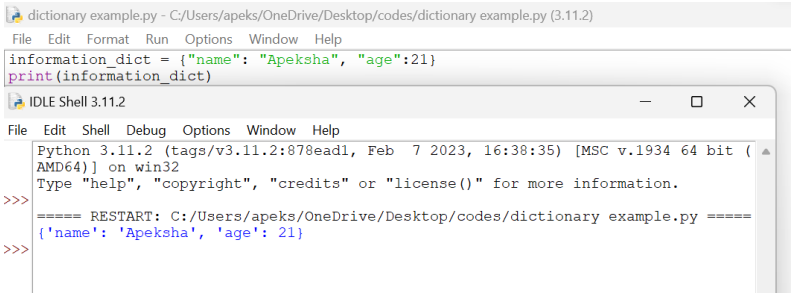
1. What is a data structure and what are some common types?
   - In simple words, data structures are the containers that organize and group data according to their type. It is a way of organizing and storing data so that they can be accessed easily and efficiently at the time of need. The common types of data structures are list, tuple, dictionary and set.

2. How do arrays differ from lists?
   - Lists are a type of data structure which supports several operations in programming whereas array is a method of organizing data in a memory.
   - All the data stored in the arrays are of the same type whereas lists can store different types of data.

3. What is a list in Python and what are some common use cases?
   - List is one of the built-in data structures in python which is used to store an ordered collection of data. It is basically used to store multiple items in a single variable. The list is mutable in nature.
   - List is used to store any type of data whether it is integer or string. For example:



4. What are some common use cases for dictionaries?
   - Dictionary is used to store complex data types. It has key and value. For example:



5. What is a nested data structure, and how can you access its elements?
   - A nested data structure is a data structure within another data structure. The elements of nested data structures can be accessed by using indexing [ ] syntax.

6. What are the key differences between a list and a tuple in Python, and when would you use one over the other?

List	Tuple
- The list is mutable.	- Tuples are immutable.
- It uses square brackets to store data.	- It uses parenthesis to store data.
- Iteration of lists can be time consuming.	- Iteration is much faster.

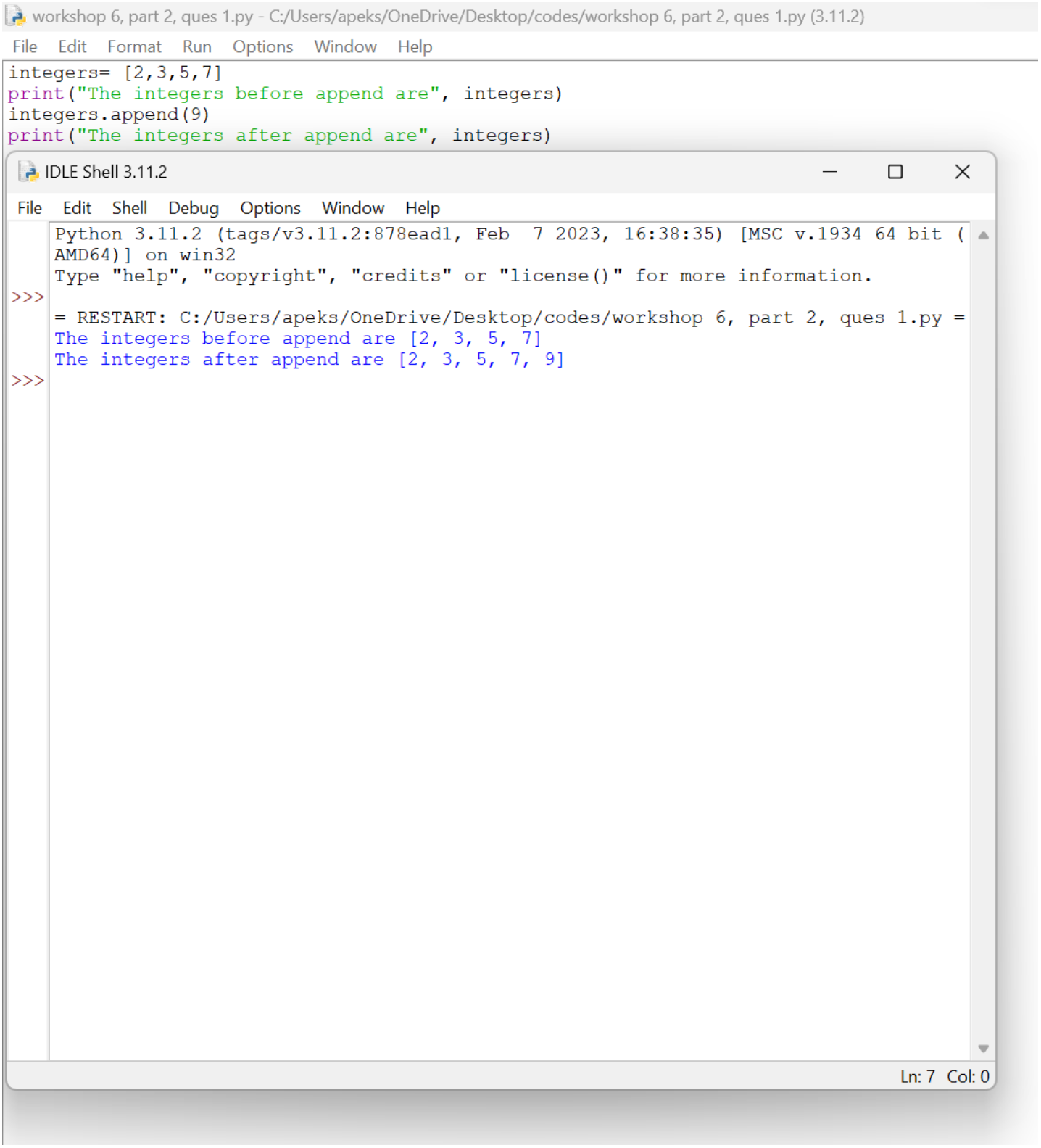
   - Lists are used over tuples when the data needs to be changed or modified later. It's the same with tuple. Tuples are used when the data does not need to be modified later.

7. How can you use Python's built-in functions and methods to manipulate and sort lists, tuples, dictionaries, and sets, and what should you watch out for when working with these data structures?
   - The built in function in python is used to manipulate lists, tuples, dictionaries and sets. The built in functions are used in list, dictionary and set as tuple is immutable. Functions like append(), remove(), clear(), insert() etc. are used to add, remove , insert, and clear the data.
   - For lists, you need to watch out for unwanted changes. For tuples, you need to remember that they are immutable. For dictionary, make sure that the keys are immutable and watch out for the errors when updating and accessing keys. For sets, remember that they are not in order and mutable.

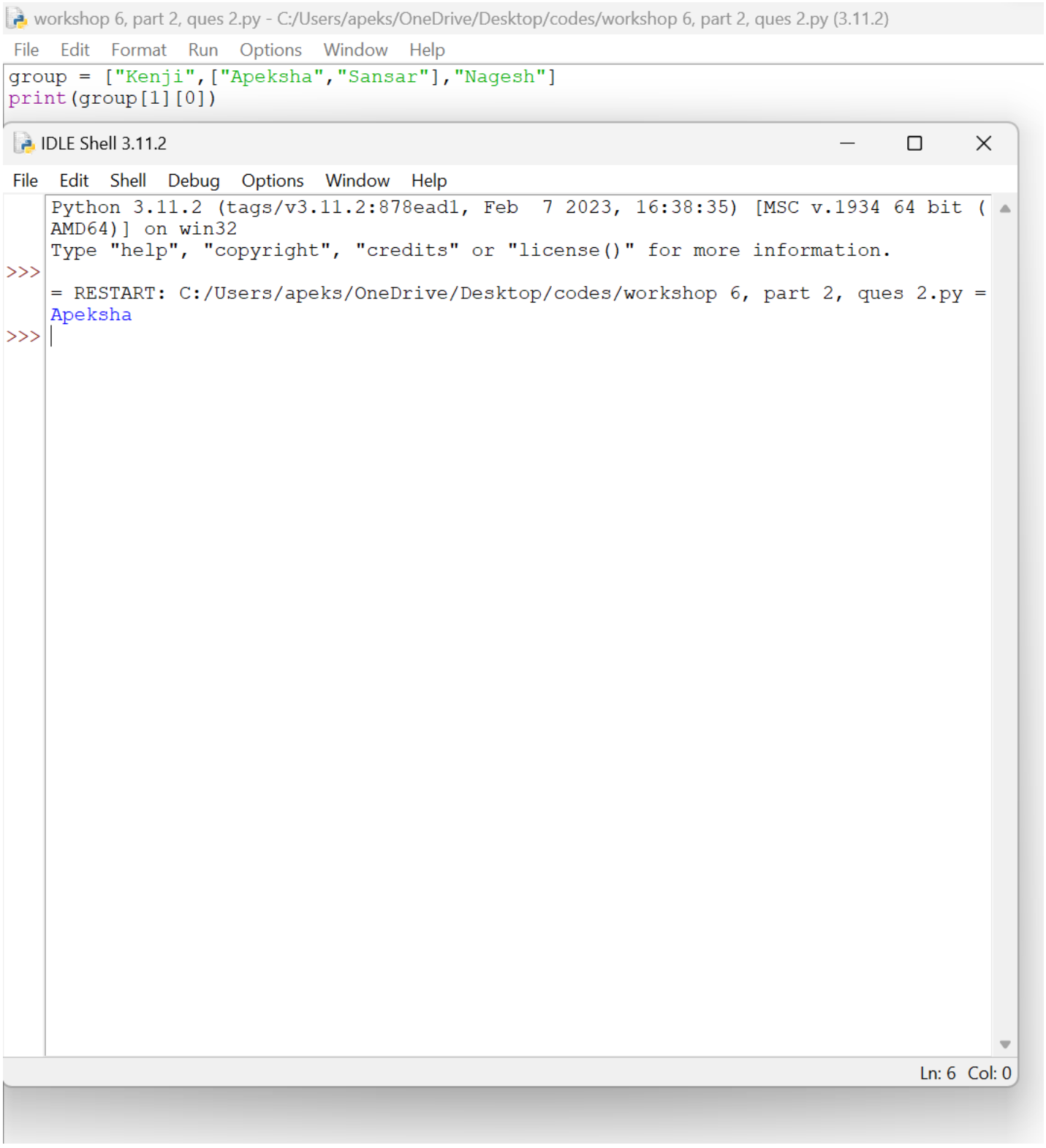
Exercise 2

---

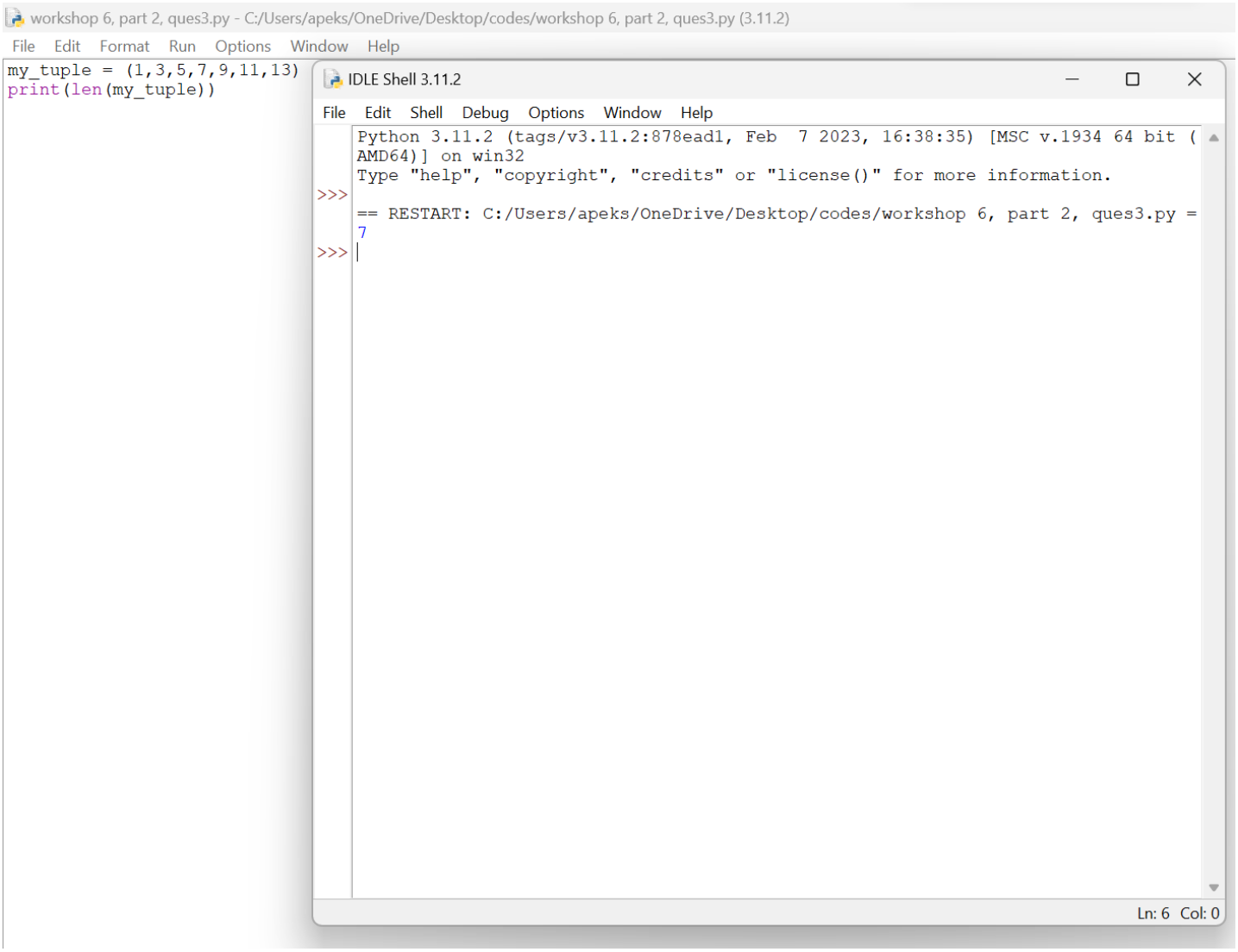
1. Write a Python program to create a list of integers and then append a new integer to the end of the list.



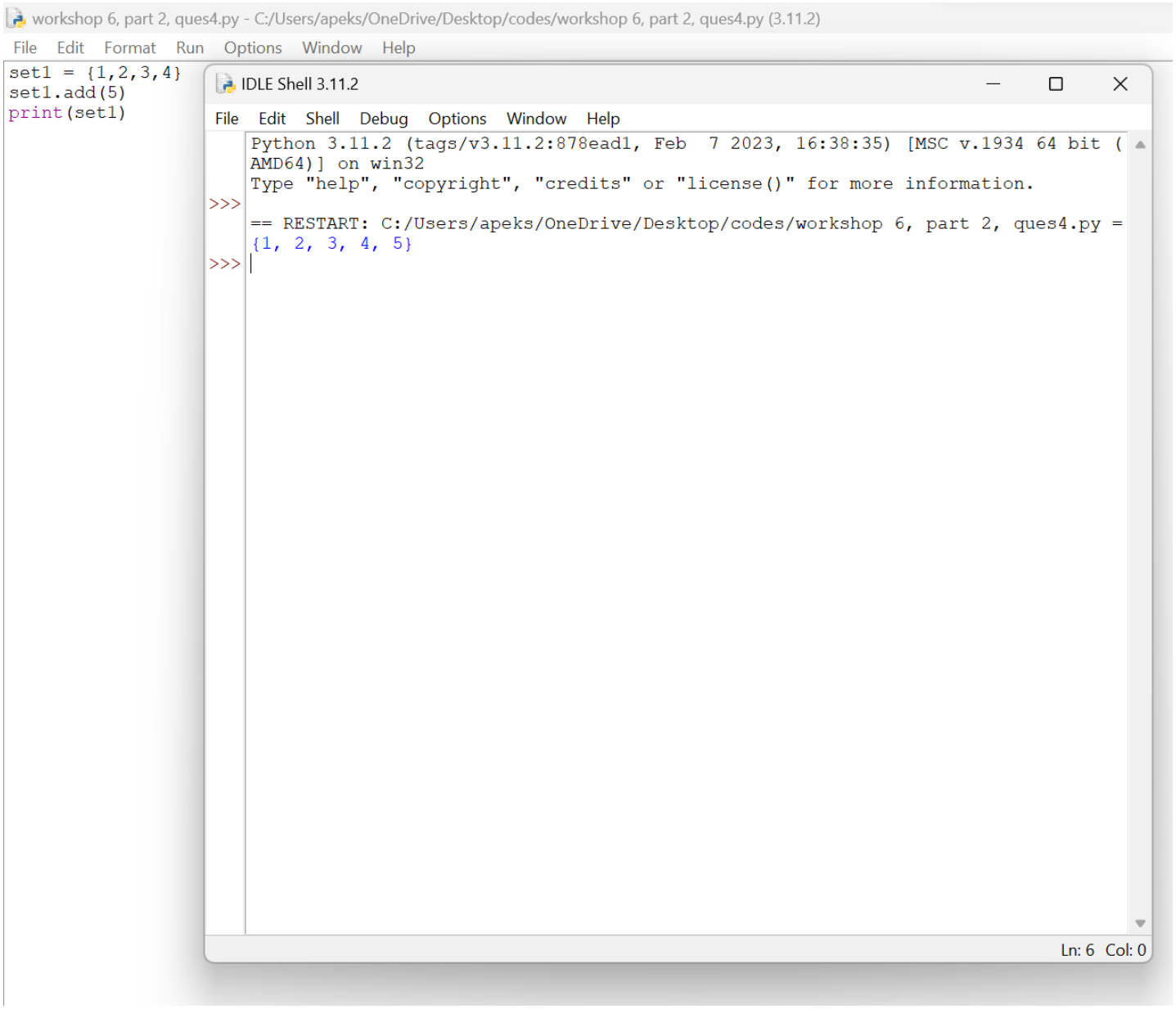
2. Write a Python program to create a nested list of strings and then print the first element of the second list.



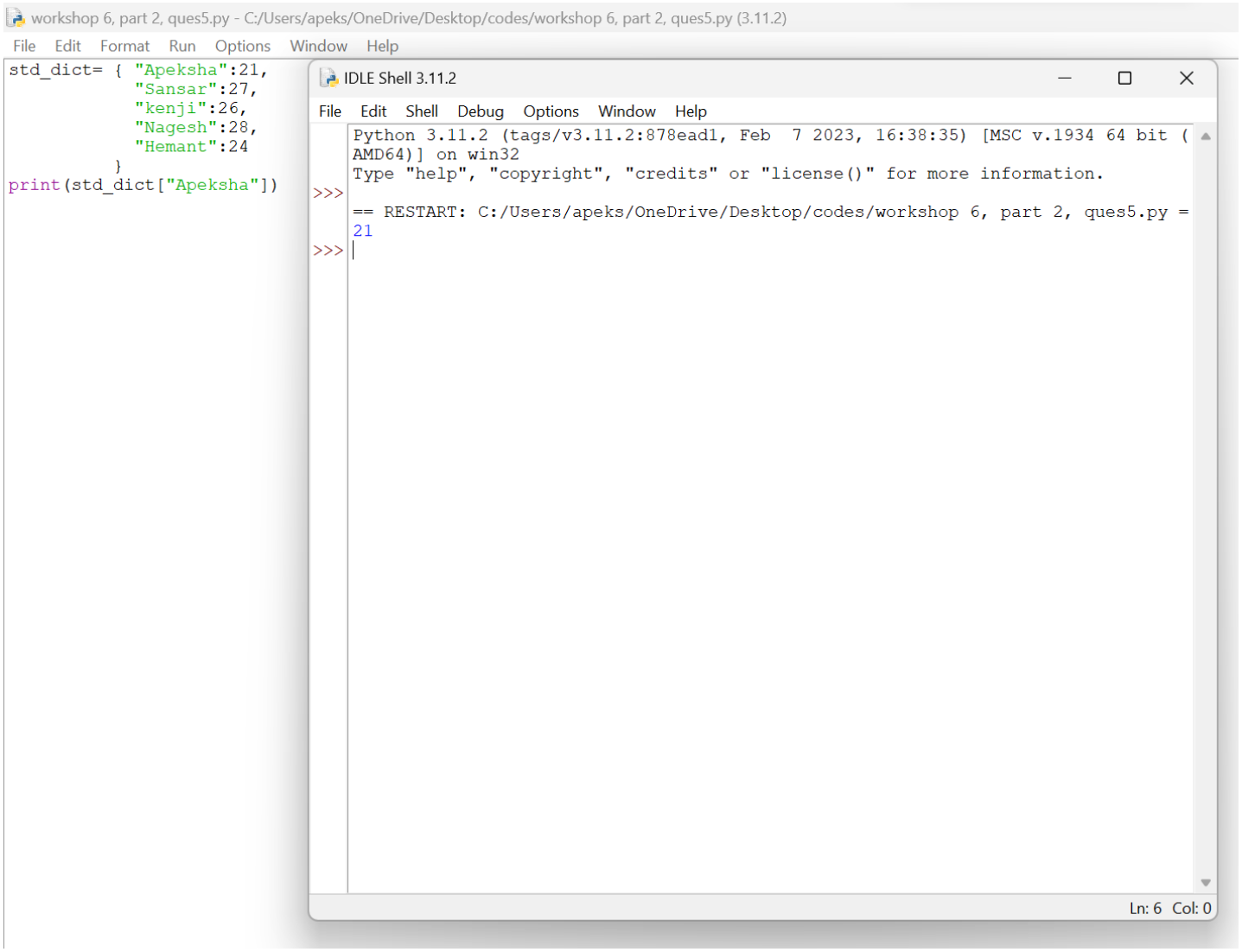
3. Write a Python program to create a tuple of integers and then print the length of the tuple.



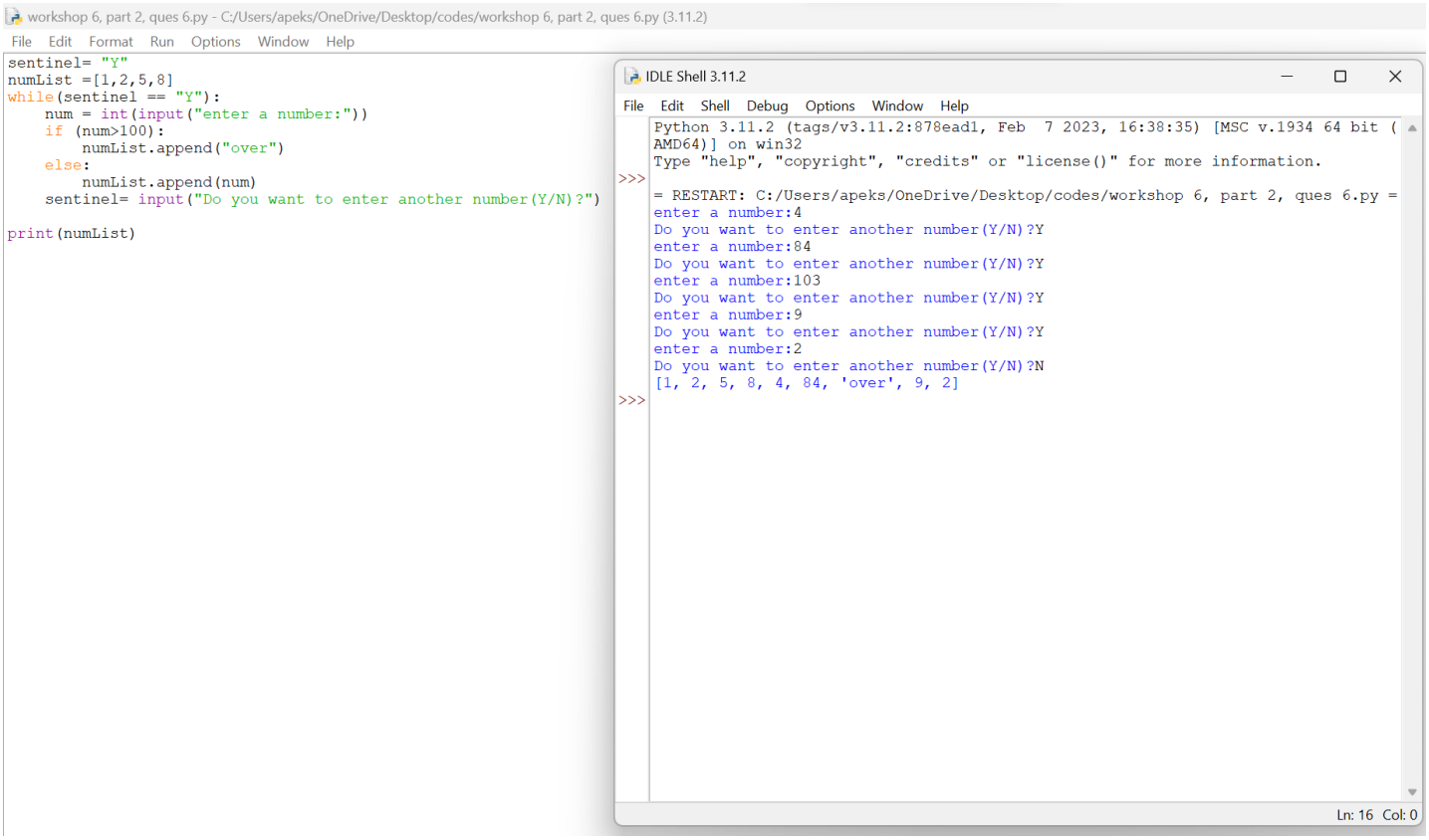
4. Write a Python program to create a set of integers and then add a new integer to the set.



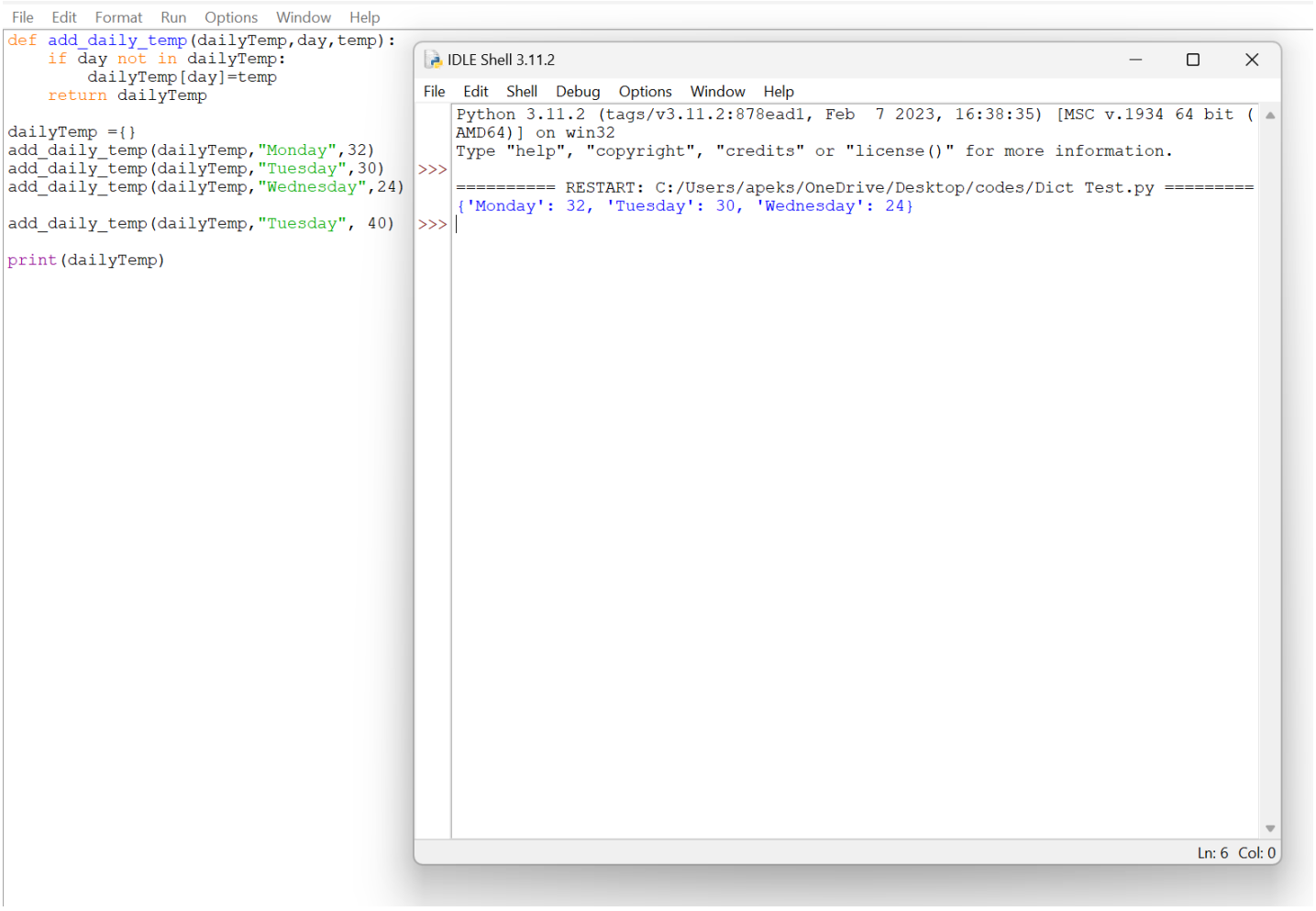
5. Write a Python program to create a dictionary of student names and their corresponding ages, and then print the age of a specific student.



6. Write a Python program that prompts the user for a list of integers and stores them in a list. For all values that are greater than 100, the string 'over' should be stored instead. The program should display the resulting list.

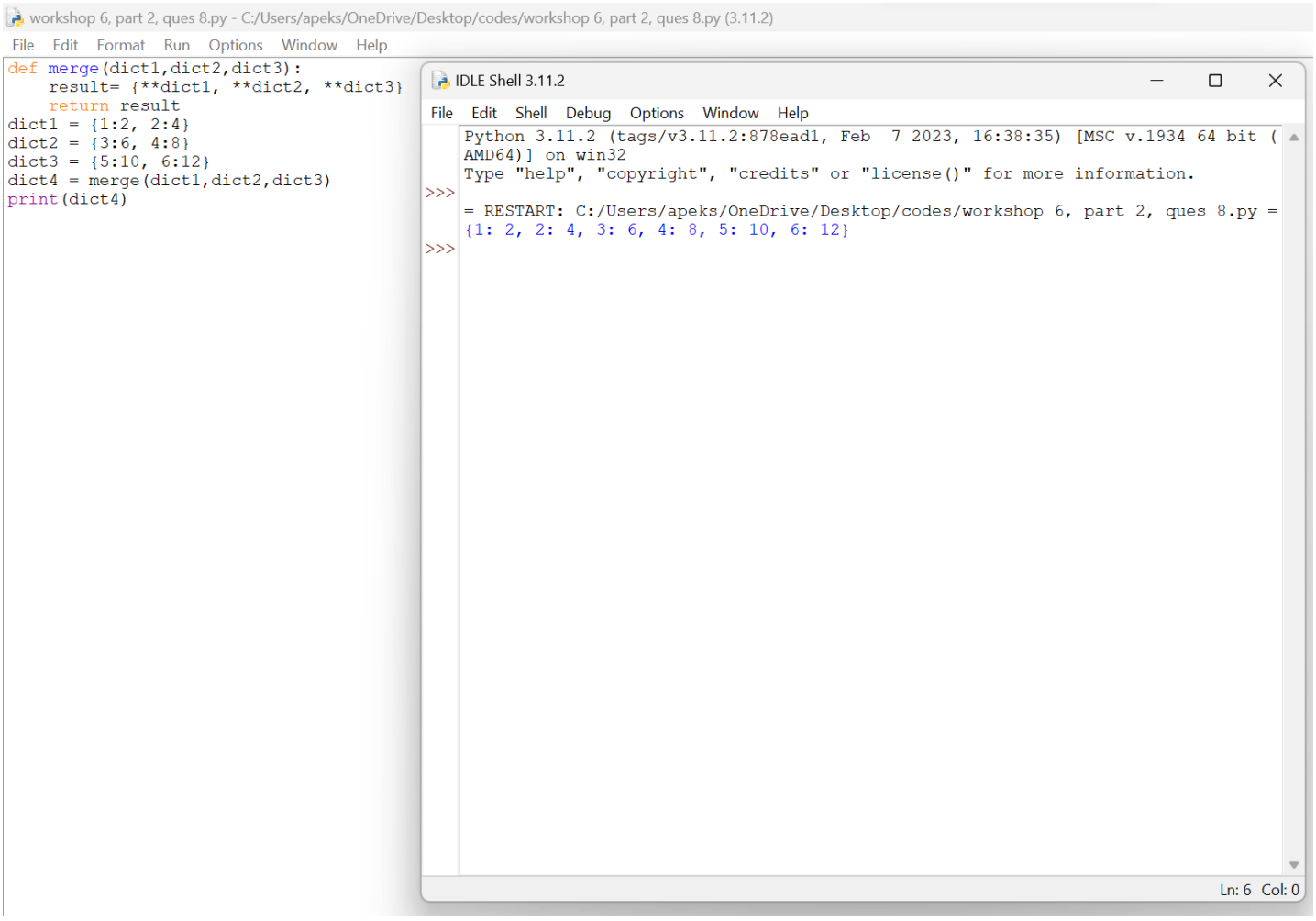


7. Write a Python function called `'add_daily_temp'` that takes three parameters: a dictionary to hold the average daily temperature for each day of the week (which may be empty), a temperature value, and the day of the week for the recorded temperature. The function should only add the temperature to the dictionary if it does not already contain a temperature for that day. If the temperature for that day is already present in the dictionary, the function should do nothing. The function should return the resulting dictionary, whether it has been updated or not.



8. Write a Python script to concatenate the following dictionaries to create a new one.

```
Sample Dictionary:
      dic1={1:10, 2:20}
      dic2={3:30, 4:40}
      dic3={5:50, 6:60}
Expected Result : {1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```



9. Write a Python program to get the top three items in a shop.

Sample data:

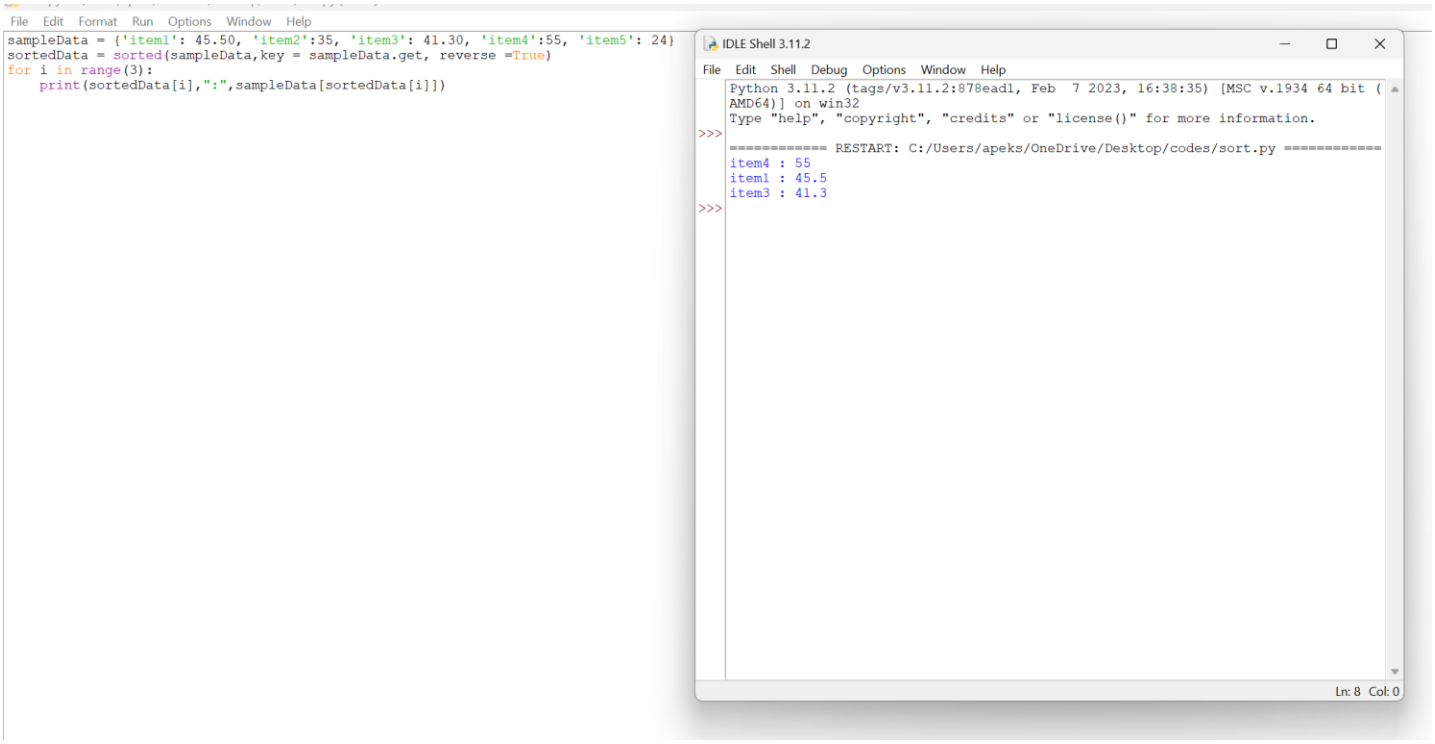
{'item1': 45.50, 'item2':35, 'item3': 41.30, 'item4':55, 'item5': 24}

Expected Output:

item4: 55

item1: 45.5

Item3: 41.3

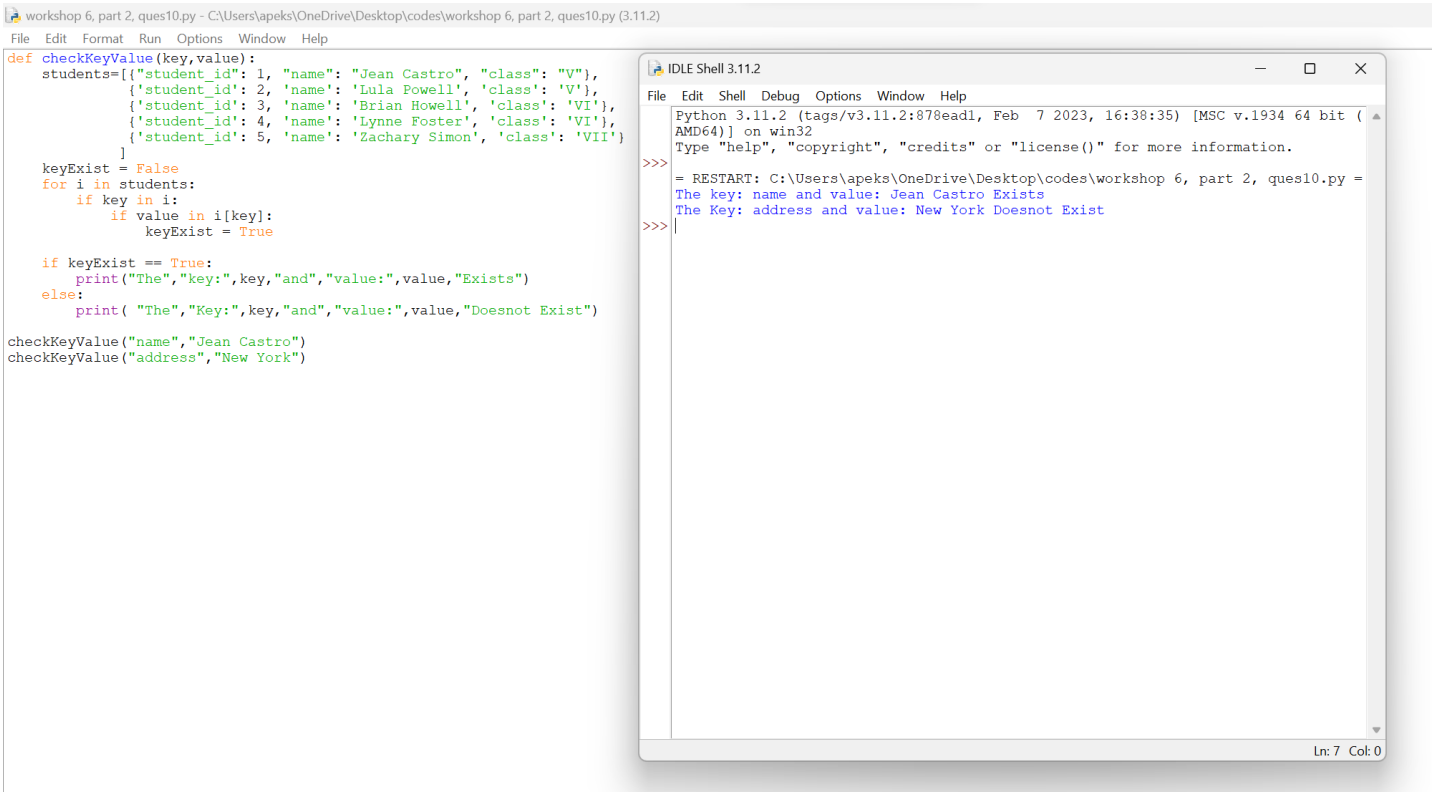


10. Write a Python program to check if a specific key and a value exist in a dictionary.

```
students=[
        {"student_id": 1, "name": "Jean Castro", "class": "V"},
        {'student_id': 2, 'name': 'Lula Powell', 'class': 'V'},
        {'student_id': 3, 'name': 'Brian Howell', 'class': 'VI'},
        {'student_id': 4, 'name': 'Lynne Foster', 'class': 'VI'},
        {'student_id': 5, 'name': 'Zachary Simon', 'class': 'VII'}
        ]
```

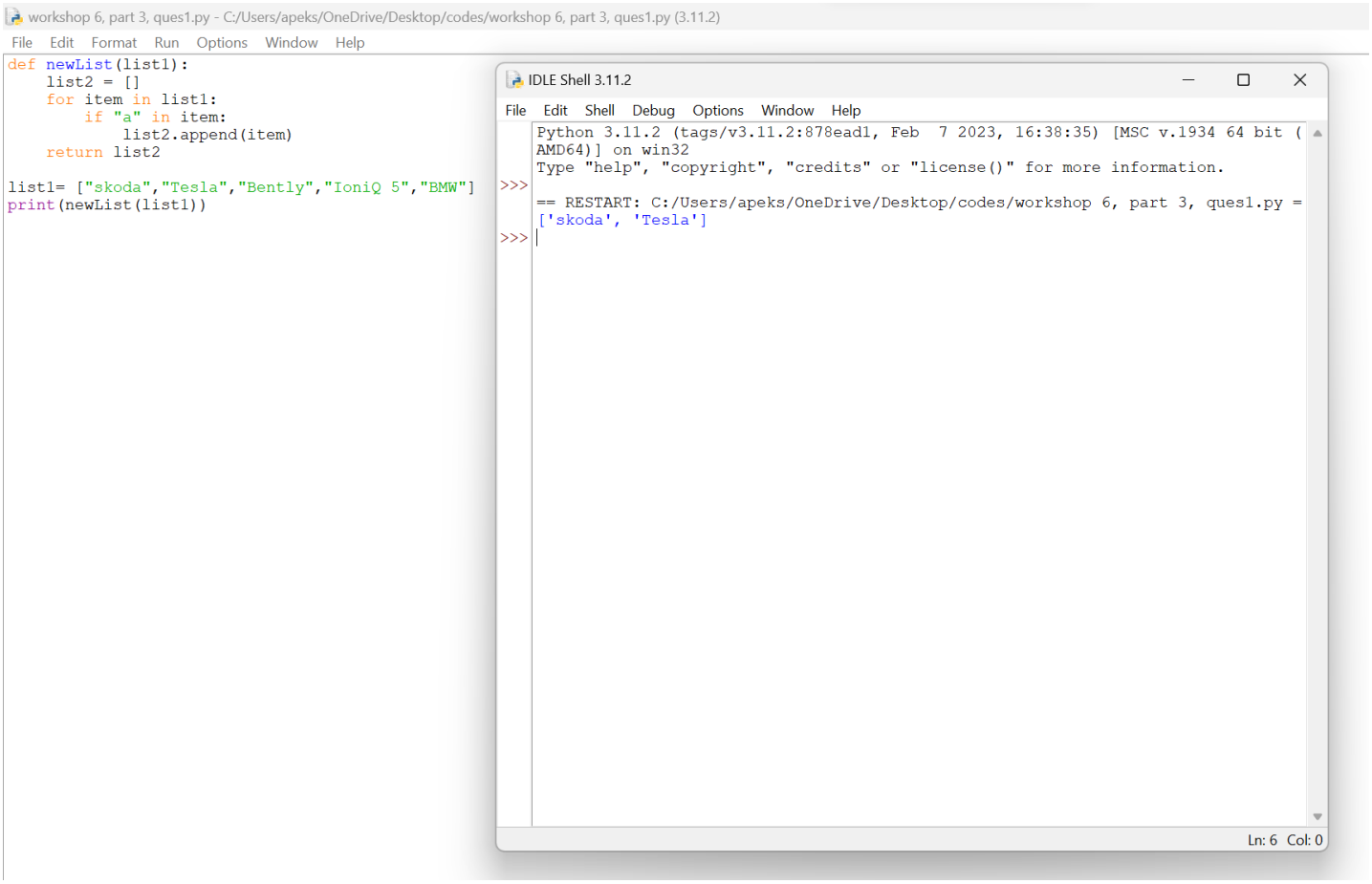
Your output should as follow:

```
Key: 'name' and Value: 'Jean Castro' do exist
Key: 'address' and Value: 'New York' do not exist
```

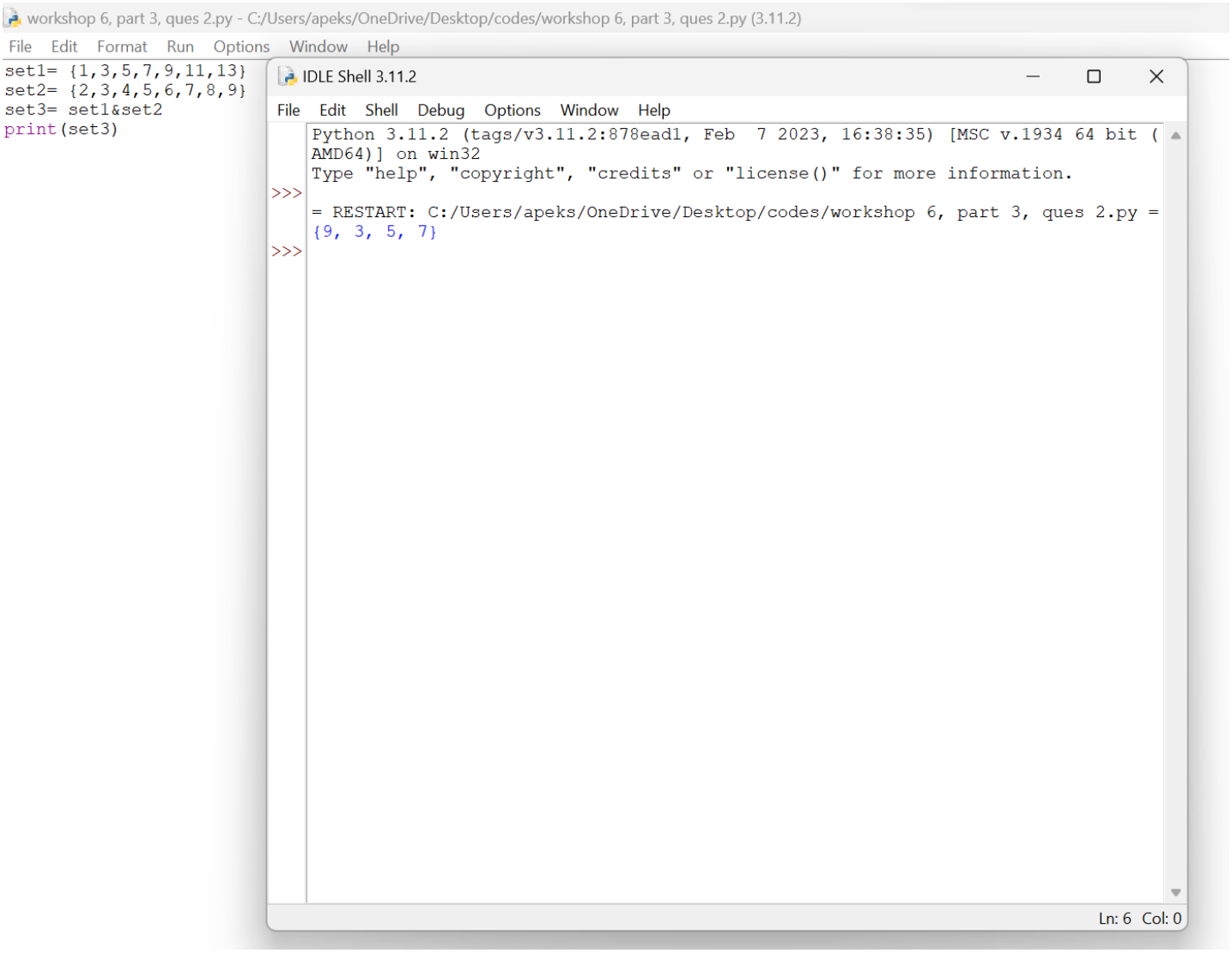


Exercise 3

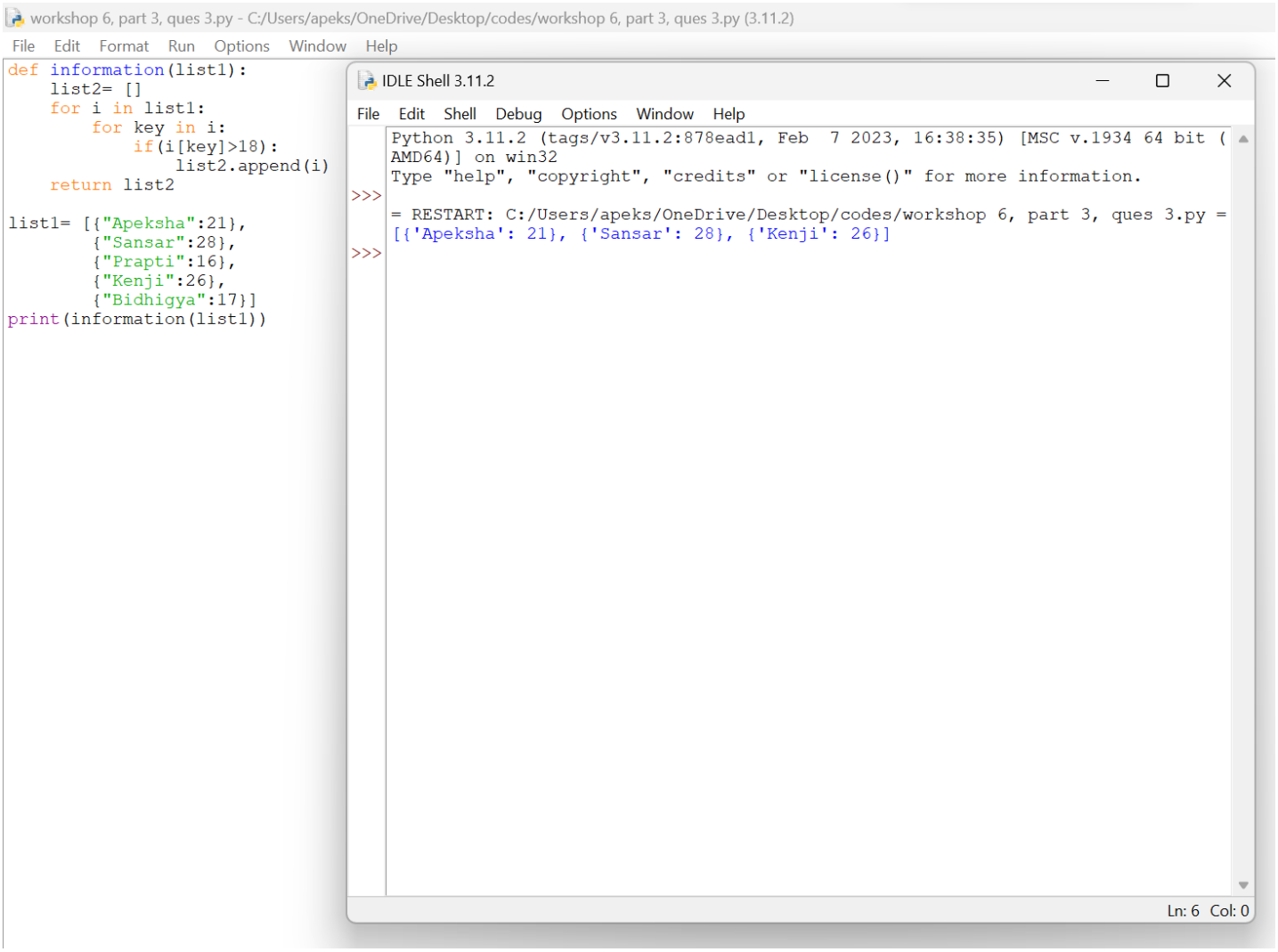
1. Write a Python program that takes in a list of strings and returns a new list with only the strings that contain the letter 'a'.



2. Write a Python program that takes in two sets of integers and returns a new set with only the common elements in both sets.



3. Write a Python program that takes in a list of dictionaries representing people with their age and returns a new list of dictionaries with only the people over the age of 18.



4. Write a Python program that takes in a set of strings and returns a new set with only the strings that start with a vowel.

workshop 6, part 3, ques4.py - C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques4.py (3.11.2)

File Edit Format Run Options Window Help

```
def sets(set1):
    set2=set()
    for item in set1:
        if(item[0]=="a" or item[0]=="e" or item[0]=="i" or item[0]=="o" or item[0]=="u"):
            set2.add(item)
    return set2
set1= ["apple","banana","orange","lychee","grapes"]
print(sets(set1))
```

IDLE Shell 3.11.2

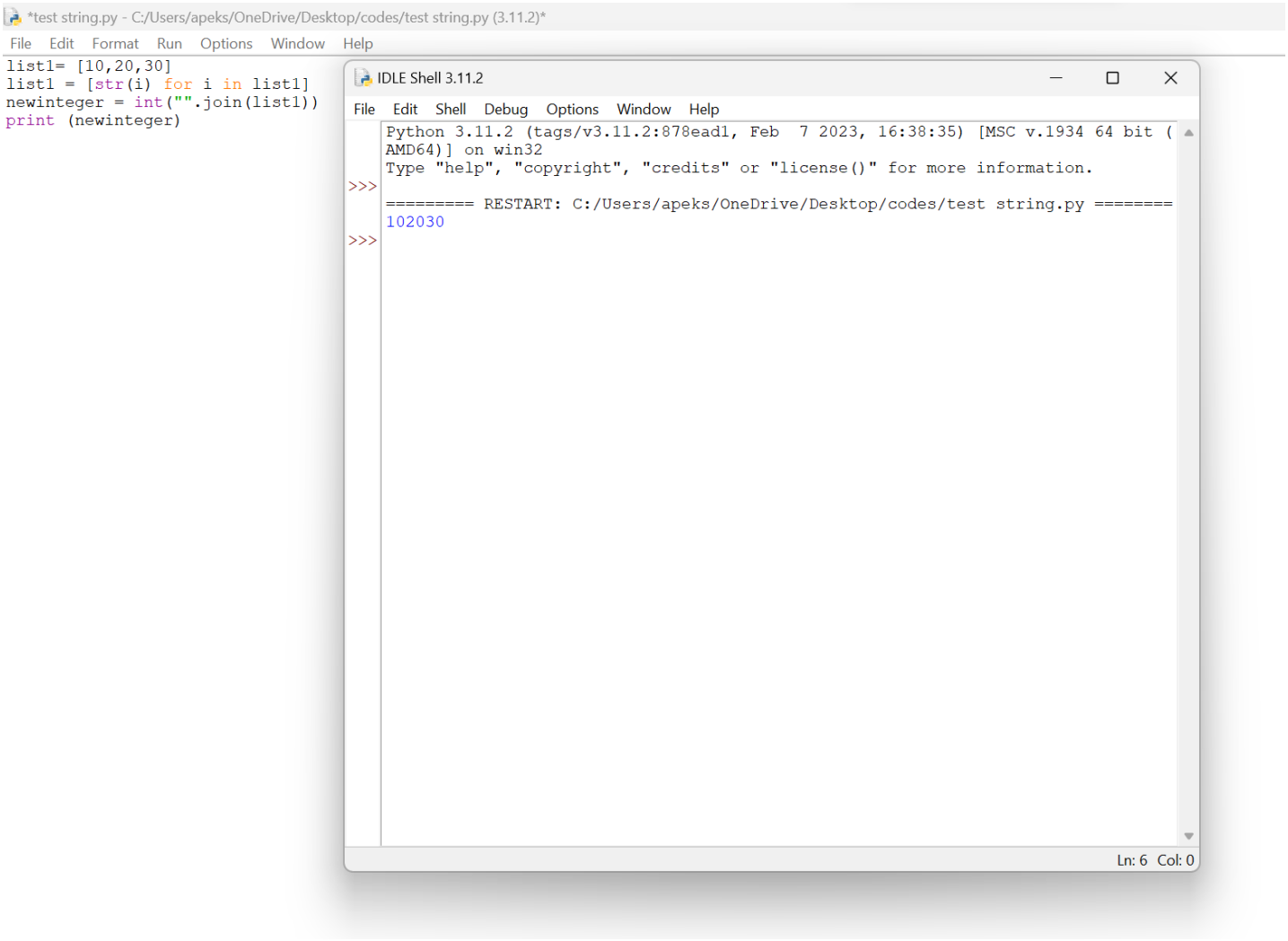
File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques4.py =
{'apple', 'orange'}
>>>
```

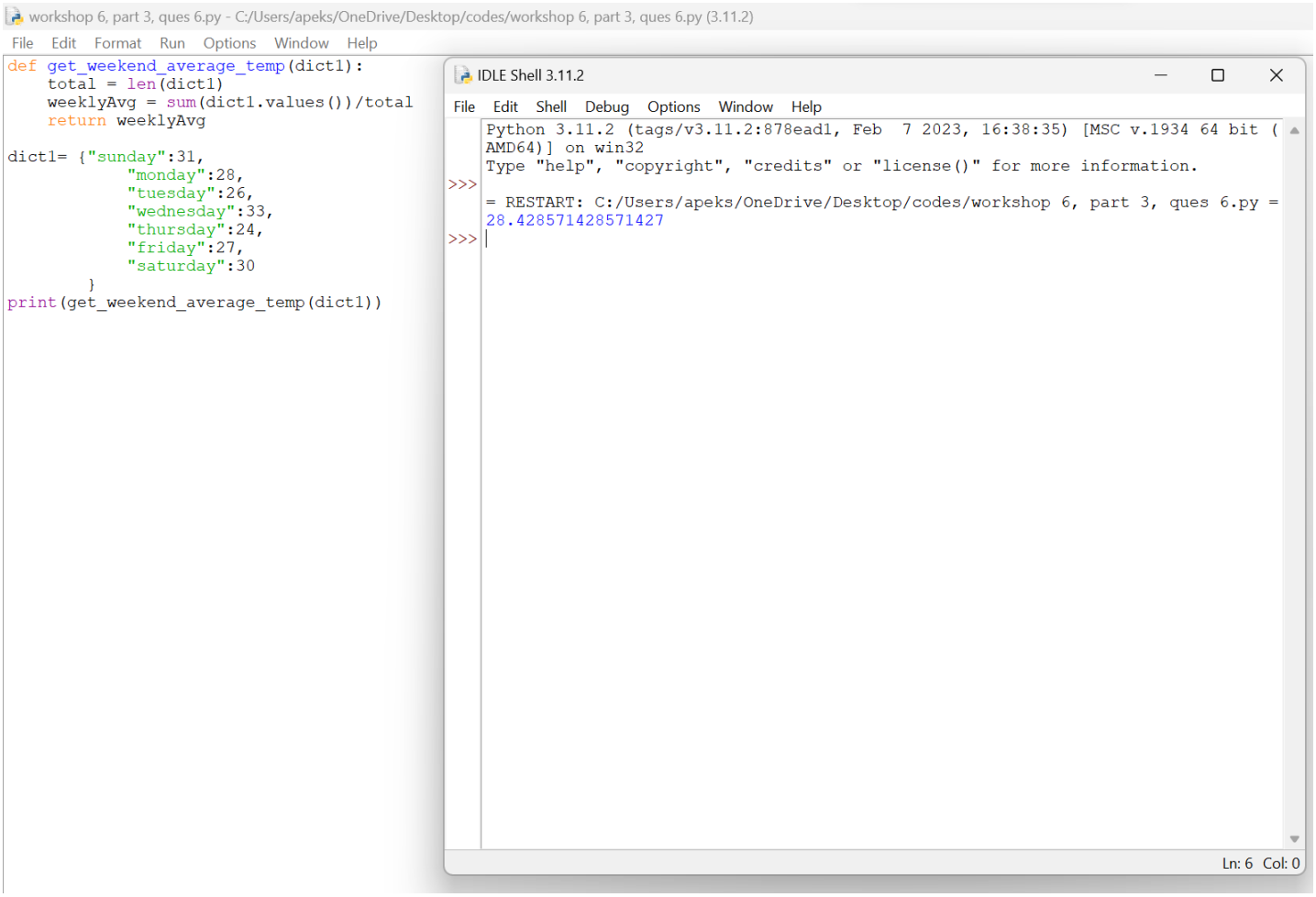
Ln: 6 Col: 0

5. Write a Python program to convert a list of multiple integers into a single integer.

```
Sample list: [11, 33, 50]
Expected Output: 113350
```



6. Write a Python function named get_weekend_average_temp that is passed a dictionary of daily temperatures and returns the average temperature over the weekend for the weekly temperatures given.



7. **Scenario Question:**

   You are a teacher and you have a **list of dictionaries** containing information about your students. Each dictionary represents a single student and contains the following keys: "**name**", "**age**", "**gender**", and "**grades**". The "**grades**" key points to a **list of grades** the student has received in **different subjects**. Your task is to write a Python program that calculates the **average grade** for each student and **prints** out their name and average grade.

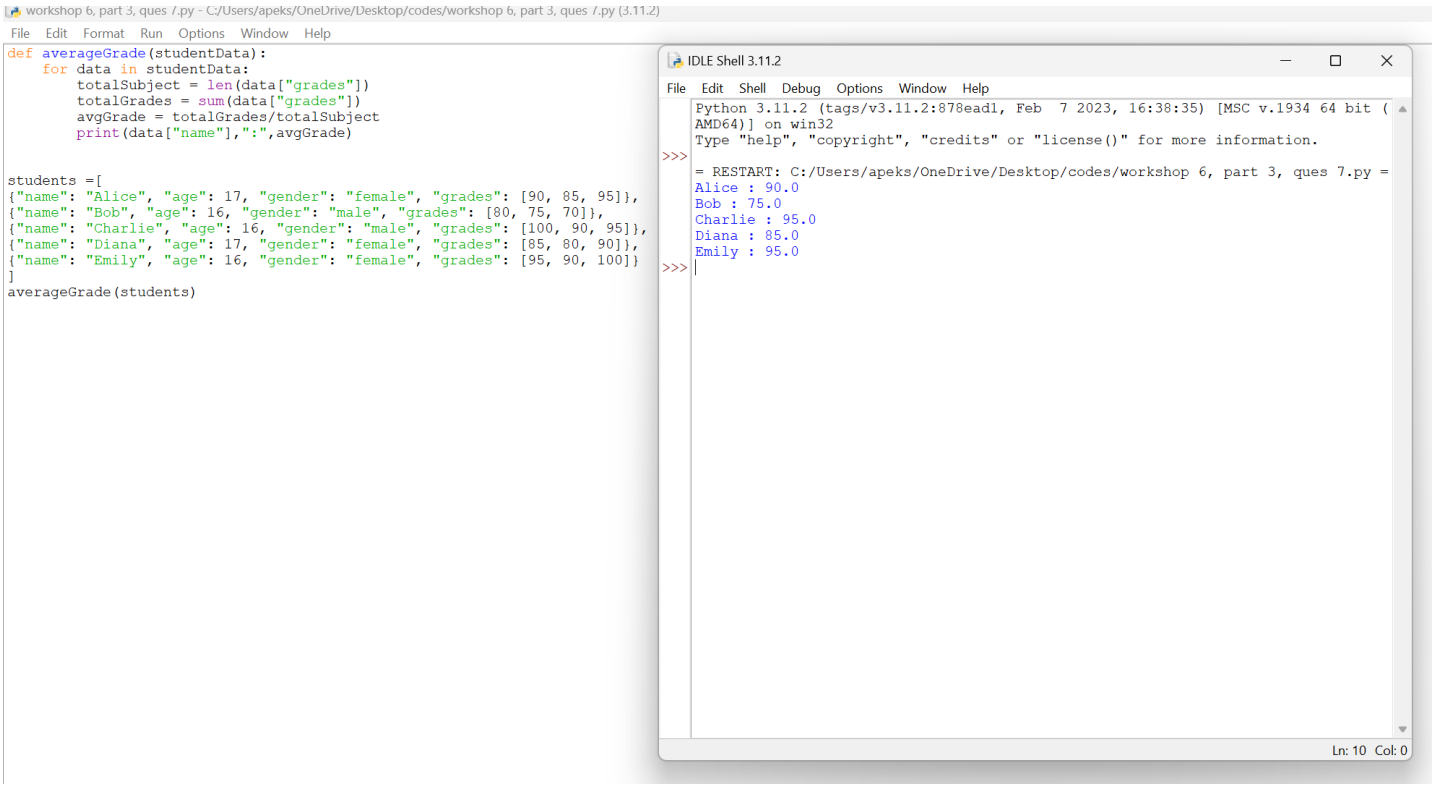
   Here's an example of **list of dictionaries** to get you started:

```
students =
[
{"name": "Alice", "age": 17, "gender": "female", "grades": [90, 85, 95]},
{"name": "Bob", "age": 16, "gender": "male", "grades": [80, 75, 70]},
{"name": "Charlie", "age": 16, "gender": "male", "grades": [100, 90, 95]},
{"name": "Diana", "age": 17, "gender": "female", "grades": [85, 80, 90]},
{"name": "Emily", "age": 16, "gender": "female", "grades": [95, 90, 100]}
]
```

   Your program should produce the following output:

```
Alice: 90.0

Bob: 75.0

Charlie: 95.0

Diana: 85.0

Emily: 95.0
```

   Hint: You'll need to use nested loops to iterate over the list of dictionaries and the list of grades for each student. You can use the len() function to get the number of grades for each student.


workshop 6, part 3, ques 7.py - C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques 7.py (3.11.2)
File Edit Format Run Options Window Help
def averageGrade(studentData):
for data in studentData:
totalSubject = len(data["grades"])
totalGrades = sum(data["grades"])
avgGrade = totalGrades/totalSubject
print(data["name"],":",avgGrade)
students =[
{"name": "Alice", "age": 17, "gender": "female", "grades": [90, 85, 95]},
{"name": "Bob", "age": 16, "gender": "male", "grades": [80, 75, 70]},
{"name": "Charlie", "age": 16, "gender": "male", "grades": [100, 90, 95]},
{"name": "Diana", "age": 17, "gender": "female", "grades": [85, 80, 90]},
{"name": "Emily", "age": 16, "gender": "female", "grades": [95, 90, 100]}
]
averageGrade(students)
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques 7.py =
Alice : 90.0
Bob : 75.0
Charlie : 95.0
Diana : 85.0
Emily : 95.0
>>>
Ln: 10 Col: 0

8. Write a Python program that takes in a list of tuples representing book titles and their corresponding authors and returns a new list of tuples sorted by author name in alphabetical order.

workshop 6, part 3, ques 8.py - C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques 8.py (3.11.2)

File Edit Format Run Options Window Help

```
def order(bookTitles):
    def get_author(book_tuple):
        return book_tuple[1]
    sortedBookTitles = sorted(bookTitles,key=get_author)
    return sortedBookTitles

booklist= [
    ("To kill a mockingbird","Harper Lee"),
    ("The palace of illusions","Chitra Banarjee"),
    ("The book thief","Markus Zusak"),
    ("The kite runner","Khaled Hosseini")
    ]
print(order(booklist))
```

IDLE Shell 3.11.2

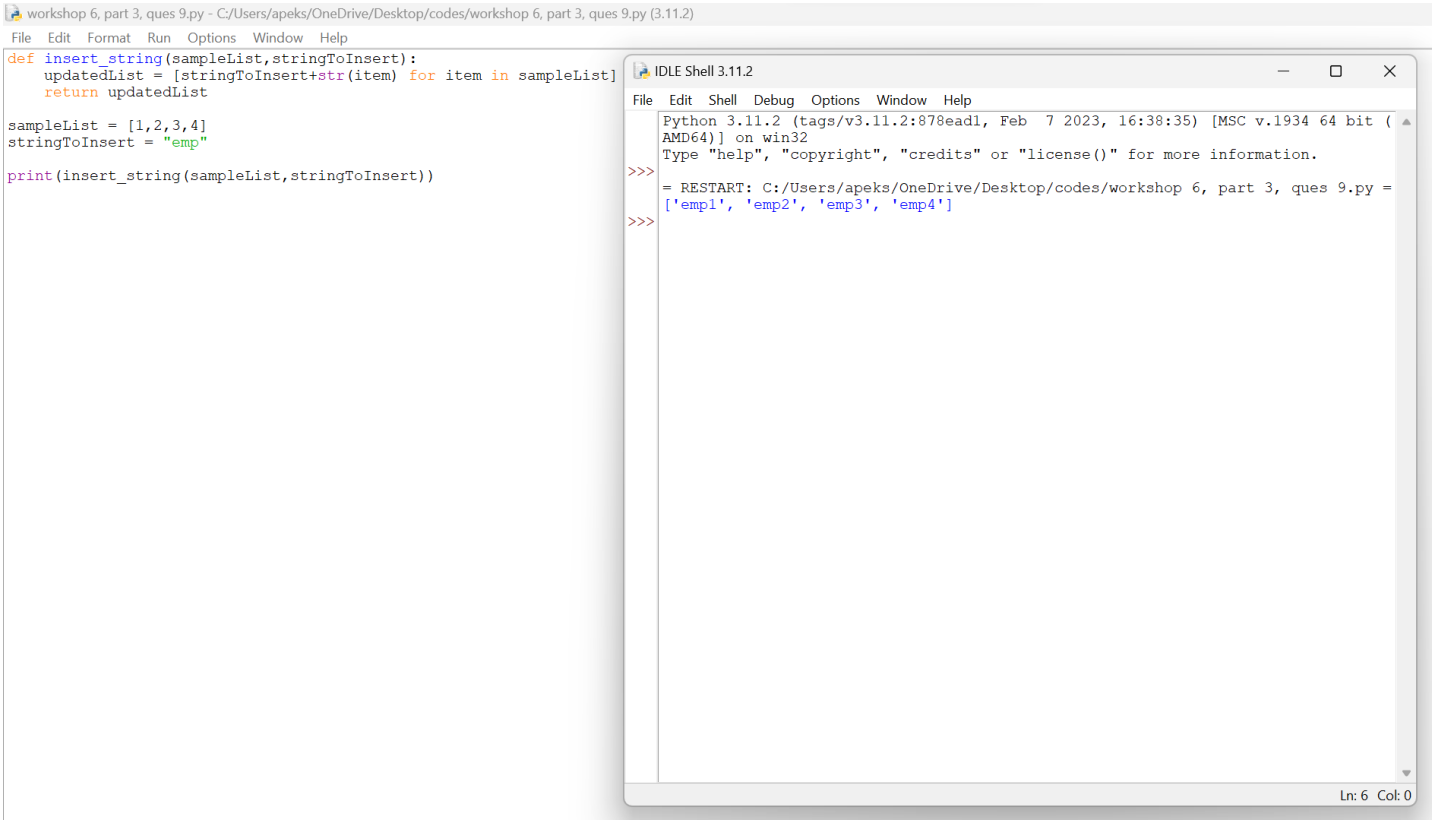
File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] o
n win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:/Users/apeks/OneDrive/Desktop/codes/workshop 6, part 3, ques 8.py =
[('The palace of illusions', 'Chitra Banarjee'), ('To kill a mockingbird', 'Harper Lee'),
('The kite runner', 'Khaled Hosseini'), ('The book thief', 'Markus Zusak')]
>>>
```

Ln: 6 Col: 0

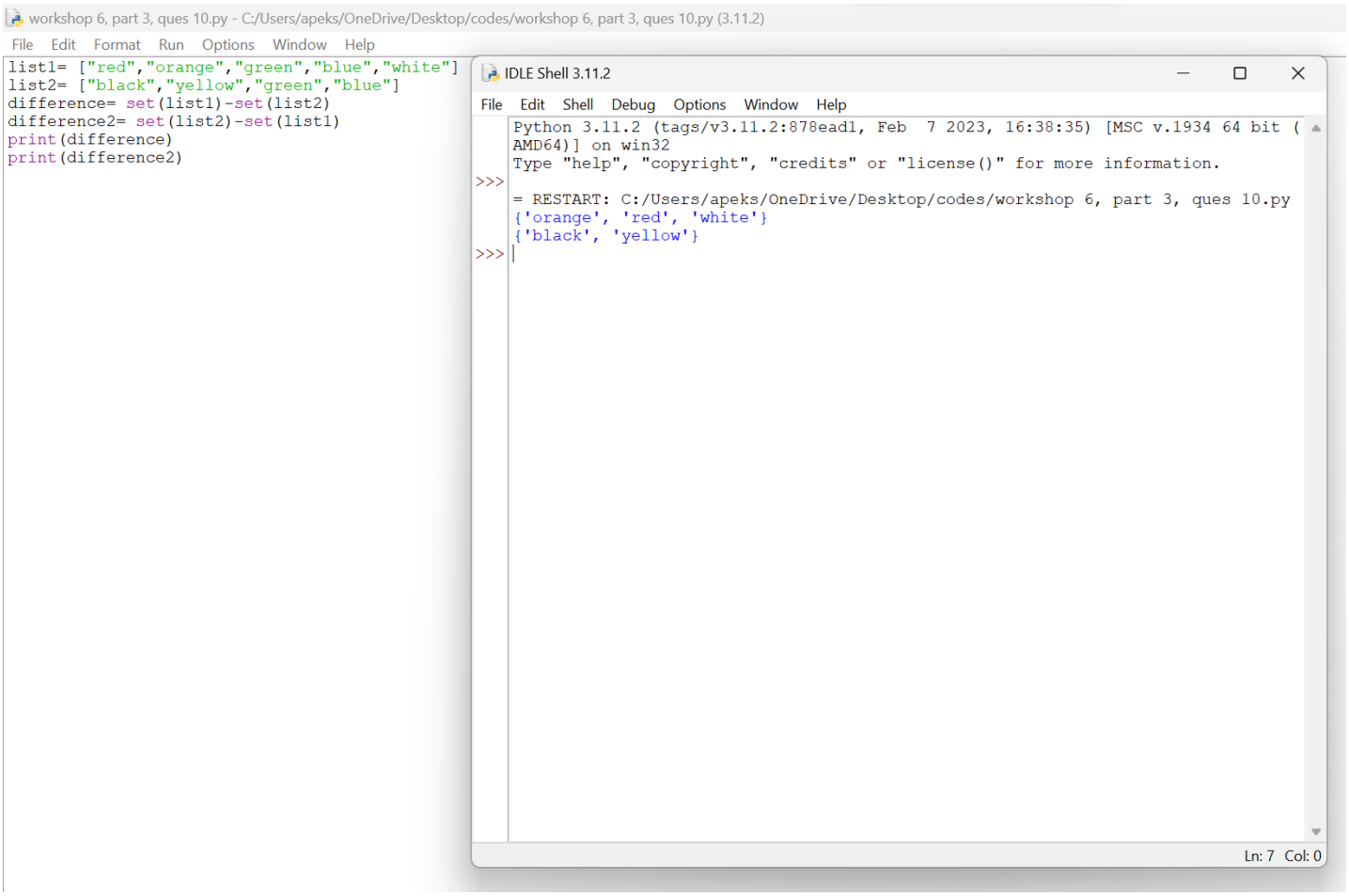
9. Write a Python program to insert a given string at the beginning of all items in a list.

```
Sample list : [1,2,3,4], string : emp
Expected output : ['emp1', 'emp2', 'emp3', 'emp4']
```



10. Write a Python program to compute the difference between two lists.

```
Sample data:

["red", "orange", "green", "blue", "white"], ["black", "yellow", "green", "blue"]
Expected Output:
Color1-Color2: ['white', 'orange', 'red']
Color2-Color1: ['black', 'yellow']
```



Exercise 4

1. You are developing a simple game where the player needs to select a cell in a 3x3 matrix and reveal its content. Initially, all cells in the matrix are hidden and have a value of 1. When the player chooses a cell, the value in that cell is revealed and replaced with an 'X' to indicate that it has been selected. The game continues until all cells have been selected.

   As the developer, you need to implement the game logic in Python. Write a function play game() that takes no parameters and returns nothing. The function should create a 3x3 matrix of integers, initialize all cells with a value of 1, and then repeatedly prompt the player to choose a cell until all cells have been selected. Your 3x3 matrix should initially look as shown below:

```
matrix = [  [1, 1, 1],
            [1, 1, 1],
            [1, 1, 1] ]
```

   When the player chooses a cell, the function should print the updated matrix with the selected cell marked with an 'X'. If the player chooses a cell that has already been selected, the function should print an error message and prompt the player to choose again.

   Once all cells have been selected, the function should print a message indicating that the game is over and the total number of moves made by the player.
   Hint: Create a 3x3 matrix of integers, prompt the player to choose a cell and reveal its content, and keep track of the number of moves until all cells have been selected.

File Edit Format Run Options Window Help

```
def display_board(board):
    for row in board:
        print(" ".join(row))
    print("\n")

def is_full(board):
    return not any("1" in row for row in board)

def playgame():
    board = [["1" for _ in range(3)] for _ in range(3)]
    moves = 0

    while not is_full(board):
        display_board(board)
        row = int(input("Enter row (0-2): "))
        col = int(input("Enter column (0-2): "))

        if row not in range(3) or col not in range(3):
            print("Invalid input. Please choose a row and column between 0 and 2.")
            continue

        if board[row][col] == "1":
            board[row][col] = "X"
            moves += 1
        else:
            print("Cell already selected. Choose another cell.")

    print("All cells have been selected.")
    display_board(board)
    print(f"Game over! Total number of moves made: {moves}")

playgame()
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
1 1 1


Enter row (0-2): 1
Enter column (0-2): 1
1 1 1
1 X 1
1 1 1


Enter row (0-2): 2
Enter column (0-2): 2
1 1 1
1 X 1
1 1 X


Enter row (0-2): 0
Enter column (0-2): 0
X 1 1
1 X 1
1 1 X


Enter row (0-2): 1
Enter column (0-2): 2
X 1 1
1 X X
1 1 X


Enter row (0-2): 1
Enter column (0-2): 1
Cell already selected. Choose another cell.
X 1 1
1 X X
1 1 X
```

2. Imagine that you are working on a project to assist a nearby library in managing their book inventory. They want you to design a program that enables them to **add new books**, **remove old books**, and perform advanced **book searches**. You make the decision to use a **list to store** all of the library's books. Moreover, your book list will be as shown below:

```
books = [

{"title": "The Great Gatsby", "author": "F. Scott Fitzgerald", "year": 1925},

{"title": "The Hobbit", "author": "J.R.R. Tolkien", "year": 1937},

{"title": "The Lord of the Rings", "author": "J.R.R. Tolkien", "year": 1954},

{"title": "The Da Vinci Code", "author": "Dan Brown", "year": 2003}

]
```

Task: Create a Python program that prompts user with a **dashboard menu** as follows:

1. Add a new book
2. Remove a book
3. Search for a book by title
4. Search for a book by author (optional)
5. List all the books
6. Quit

Hint: You may want to use loops in your program and define a function for each task.

```python
books = []

def add_book():
    book = input("Enter the book name:")
    books.append(book)
    print(f"{book} has been added to the library.")

def remove_book():
    book = input("Enter the book name:")
    if(book in books):
        books.remove(book)
        print(f"{book} has been removed from the library.")
    else:
            print(f"{book} is not in the library")

def search_bookTitle():
    book = input("Enter the book name:")
    if(book in books):
        print(f"{book} is in the library.")
    else:
        print(f"{book} is not in the library.")

def search_bookAuthor():
    book = input("Enter the book name:")
    if(book in books):
        print(f"{book} is in the library.")
    else:
        print(f"{book} is not in the library.")

def list_books():
    for book in books:
        print(book)

while True:
    print("1. Add a new book")
    print("2. Remove a book")
    print("3. Search for a book by title")
    print("4. search for a book by author")
    print("5. list all books")
    print("6. Quit")

    choice = input("Enter your choice:")

    if(choice == "1"):
        add_book()
    elif(choice == "2"):
        remove_book()
    elif(choice == "3"):
        search_bookTitle()
    elif(choice == "4"):
        search_bookAuthor()
    elif(choice == "5"):
        list_books()
    elif(choice == "6"):
        break
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
1. Add a new book
2. Remove a book
3. Search for a book by title
4. search for a book by author
5. list all books
6. Quit
Enter your choice:1
Enter the book name:The book thief
The book thief has been added to the library.
1. Add a new book
2. Remove a book
3. Search for a book by title
4. search for a book by author
5. list all books
6. Quit
Enter your choice:1
Enter the book name:Angels and demons
Angels and demons has been added to the library.
1. Add a new book
2. Remove a book
3. Search for a book by title
4. search for a book by author
5. list all books
6. Quit
Enter your choice:1
Enter the book name:The palace of illusions
The palace of illusions has been added to the library.
1. Add a new book
2. Remove a book
3. Search for a book by title
4. search for a book by author
5. list all books
6. Quit
Enter your choice:3
Enter the book name:Angels and demons
Angels and demons is in the library.
```

Ln: 67 Col: 0

Tutorial content:

6 tryout 1.py - C:\Users\apeks\OneDrive\Desktop\codes\6 tryout 1.py (3.11.2)

File Edit Format Run Options Window Help

```
#Make a list of age of friends
Age_of_friends= [20,21,22]
print(Age_of_friends)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\apeks\OneDrive\Desktop\codes\6 tryout 1.py ==========
[20, 21, 22]
>>>
```

6 tryouts 2.py - C:\Users\apeks\OneDrive\Desktop\codes\6 tryouts 2.py (3.11.2)

File Edit Format Run Options Window Help

```
#print elements of list using index number
programming_language= ["Python","Java","C++"]
print(programming_language[0])
print(programming_language[2])
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========== RESTART: C:\Users\apeks\OneDrive\Desktop\codes\6 tryouts 2.py ========
Python
C++
>>>
```

6 tryouts 3.py - C:\Users\apeks\OneDrive\Desktop\codes\6 tryouts 3.py (3.11.2)

File Edit Format Run Options Window Help

```
#using append
num=[4,5,6,7]
print("Before append:", num)
num.append(8)
print("After append:", num)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
========= RESTART: C:\Users\apeks\OneDrive\Desktop\codes\6 tryouts 3.py ========
Before append: [4, 5, 6, 7]
After append: [4, 5, 6, 7, 8]
>>>
```

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

File Edit Format Run Options Window Help

```
#using insert
odd_num= [1,3,5,7,9]
odd_num.insert(3,21)
print("List of odd#:", odd_num)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
List of odd#: [1, 3, 5, 21, 7, 9]
>>>
```

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

File Edit Format Run Options Window Help

```
#using extend and append
list1= ["a","b","c"]
list2= ["d","e","f"]
print("List 1:", list1)
print("List 2:", list2)
list2.extend(list1)
print("List after append:", list2)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
List 1: ['a', 'b', 'c']
List 2: ['d', 'e', 'f']
List after append: ['d', 'e', 'f', 'a', 'b', 'c']
>>>
```

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

File Edit Format Run Options Window Help

```
#using del
languages= ["Python", "Java", "C++", "C#", "R"]
del languages[-1]
print(languages)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
['Python', 'Java', 'C++', 'C#']
>>>
```

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

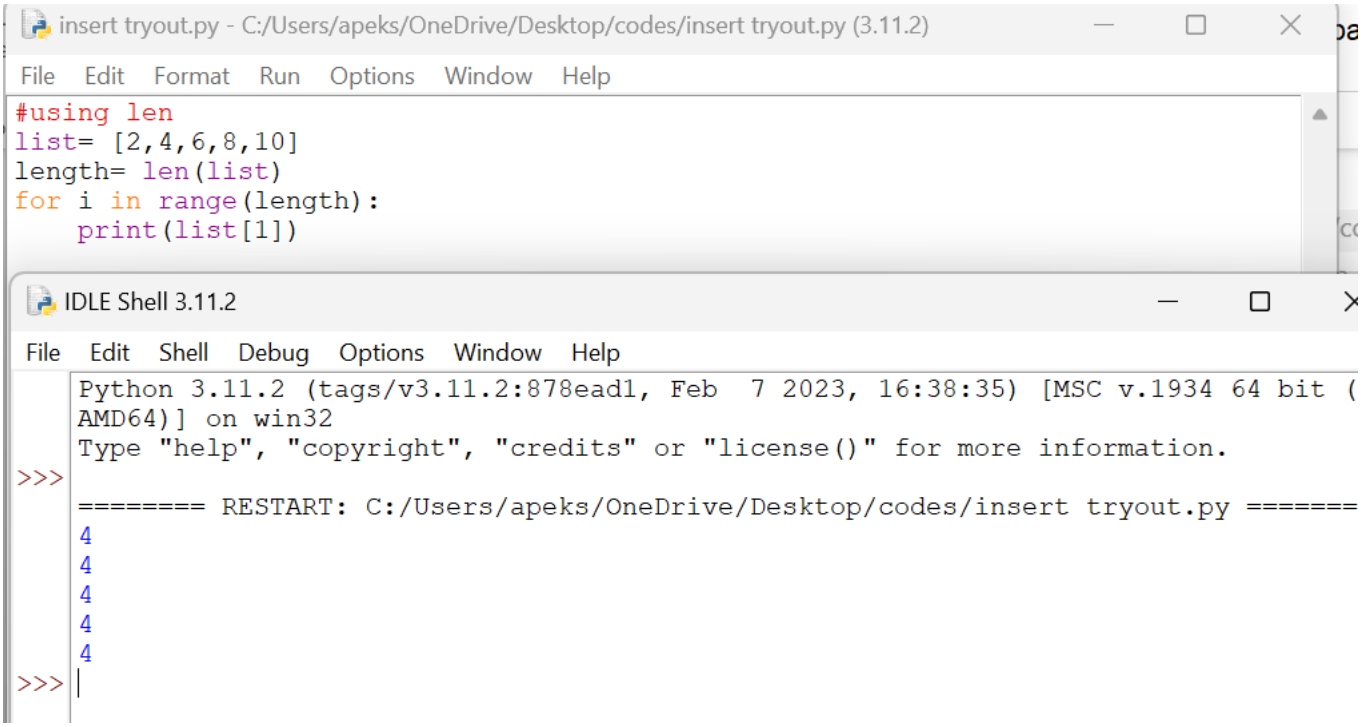
File Edit Format Run Options Window Help

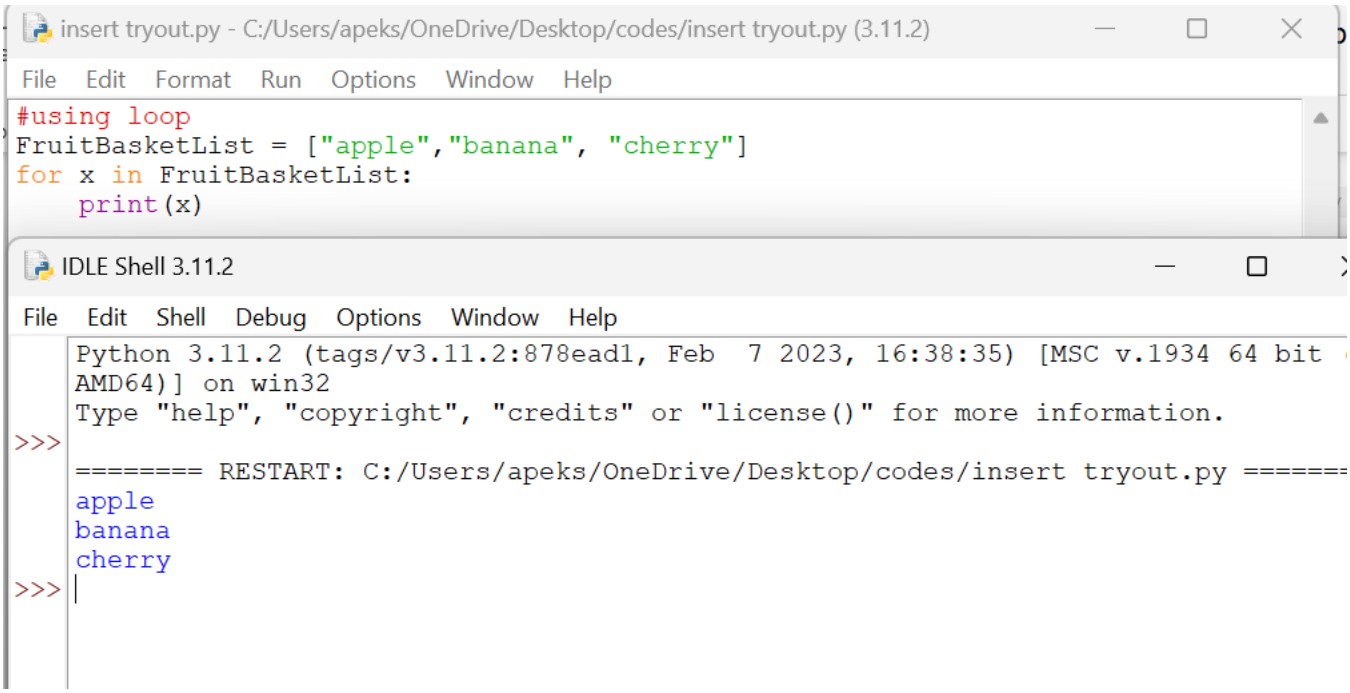
```
#using in(membership operator)
list= [2,4,6,8,10]
for i in list:
    print(i)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
2
4
6
8
10
>>>
```


insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)
File Edit Format Run Options Window Help
#using len
list= [2,4,6,8,10]
length= len(list)
for i in range(length):
print(list[1])
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
4
4
4
4
4
>>>


insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)
File Edit Format Run Options Window Help
#using loop
FruitBasketList = ["apple","banana", "cherry"]
for x in FruitBasketList:
print(x)
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
apple
banana
cherry
>>>

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

File Edit Format Run Options Window Help

```
#using while loop
FruitList= ["apple","banana", "cherry"]
i=0
while i < len(FruitList):
    print(FruitList[i])
    i= i+1
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py ======
apple
banana
cherry
>>>
```

insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)

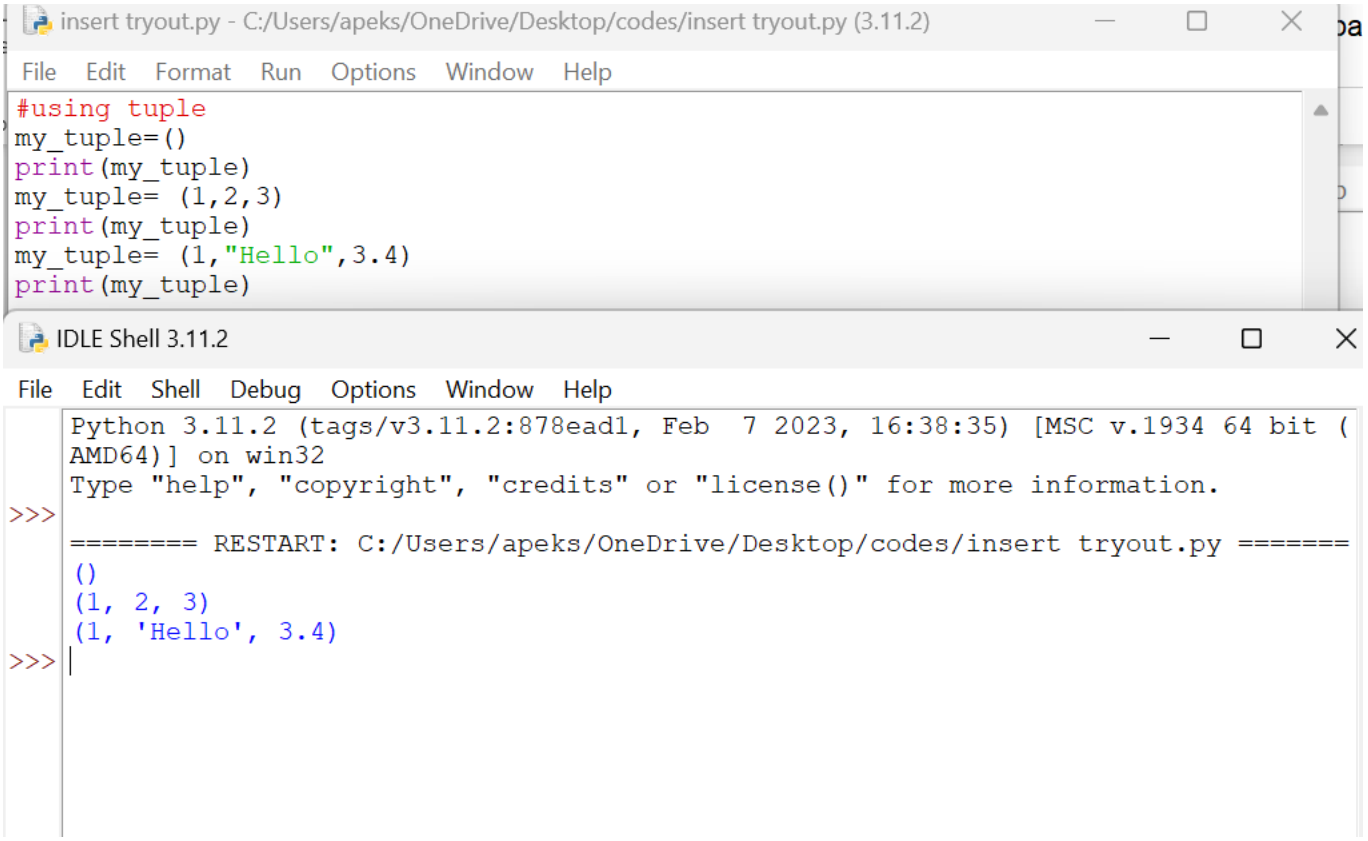
File Edit Format Run Options Window Help

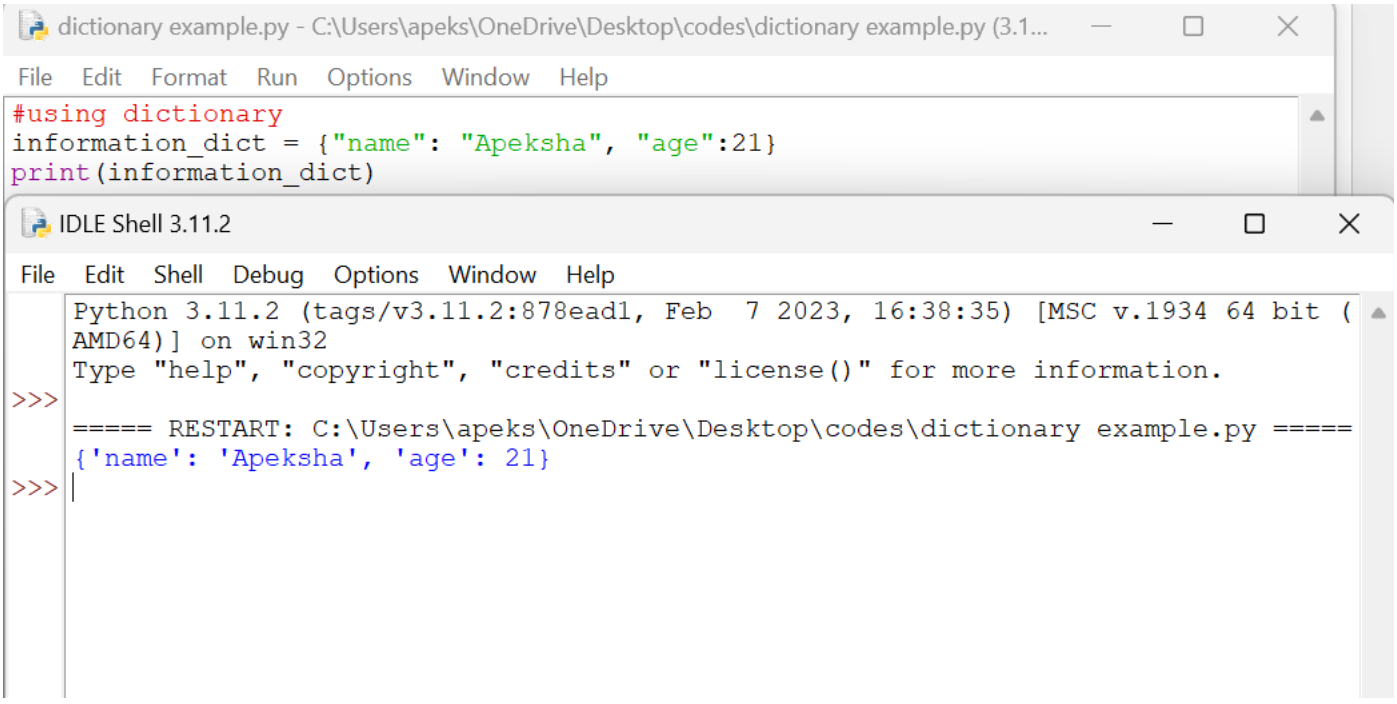
```
#using nested list
nestedList= [1,2,['a',1],3]
subList= nestedList[2]
element= nestedList[2][0]
print("List inside nested list:", subList)
print("First element of the sublist:", element)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py =======
List inside nested list: ['a', 1]
First element of the sublist: a
>>>
```


insert tryout.py - C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py (3.11.2)
File Edit Format Run Options Window Help
#using tuple
my_tuple=()
print(my_tuple)
my_tuple= (1,2,3)
print(my_tuple)
my_tuple= (1,"Hello",3.4)
print(my_tuple)
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
======== RESTART: C:/Users/apeks/OneDrive/Desktop/codes/insert tryout.py ========
()
(1, 2, 3)
(1, 'Hello', 3.4)
>>>


dictionary example.py - C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py (3.1...
File Edit Format Run Options Window Help
#using dictionary
information_dict = {"name": "Apeksha", "age":21}
print(information_dict)
IDLE Shell 3.11.2
File Edit Shell Debug Options Window Help
Python 3.11.2 (tags/v3.11.2:878ead1, Feb 7 2023, 16:38:35) [MSC v.1934 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py =====
{'name': 'Apeksha', 'age': 21}
>>>

*dictionary example.py - C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py (3....

File Edit Format Run Options Window Help

```
#using loop in dictionary
std_dict= {"name": "Apeksha",
           "age" : 21}
dict2= {x:std_dict[x] for x in std_dict}
print(dict2)

dict1 = {x: x**3 for x in (3,4,9)}
print(dict1)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py =====
{'name': 'Apeksha', 'age': 21}
{3: 27, 4: 64, 9: 729}
>>>
```

dictionary example.py - C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py (3.1...

File Edit Format Run Options Window Help

```
#using set
set1= {1,6,4,6,8,3,3,9}
set2= {7,9,1,5,8,9,3,3}
set_union= set1|set2
set_diff= set1-set2
set_intersection= set1&set2
print(set_union)
print(set_diff)
print(set_intersection)
```

IDLE Shell 3.11.2

File Edit Shell Debug Options Window Help

```
Python 3.11.2 (tags/v3.11.2:878ead1, Feb  7 2023, 16:38:35) [MSC v.1934 64 bit (
AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Users\apeks\OneDrive\Desktop\codes\dictionary example.py =====
{1, 3, 4, 5, 6, 7, 8, 9}
{4, 6}
{8, 1, 3, 9}
```

```python
#Pair programming
books = []

def add_book():
    book = input("Enter the book name:")
    books.append(book)
    print(f"{book} has been added to the library.")

def remove_book():
    book = input("Enter the book name:")
    if(book in books):
        books.remove(book)
        print(f"{book} has been removed from the library.")
    else:
            print(f"{book} is not in the library")

def search_book():
    book = input("Enter the book name:")
    if(book in books):
        print(f"{book} is in the library.")
    else:
        print(f"{book} is not in the library.")

def list_books():
    for book in books:
        print(book)

while True:
    print("1. Add a new book")
    print("2. Remove a book")
    print("3. Search for a book")
    print("4. List all books")
    print("5. Quit")

    choice = input("Enter your choice:")

    if(choice == "1"):
        add_book()
    elif(choice == "2"):
        remove_book()
    elif(choice == "3"):
        search_book()
    elif(choice == "4"):
        list_books()
    elif(choice == "5"):
        break
```