

### **# 1) What is a list in Python and what are some common use cases?**

# Lists are an ordered sequential mutable data type that store multiple values.

---

### **# 2) What are some common use cases for dictionaries?**

# To store a collection of key value pairs.

---

### **# 3) What is a nested data structure, and how can you access its elements?**

# Nested data structures are data structures within data structures, eg [[1,2,3],[4,5,6]]; array[0][1] gives 2.

---

### **# 4) What are the key differences between a list and a tuple in Python, and when would you use one over the other?**

# Lists are mutable, tuples are not. Tuples can be used when the collection can't or shouldn't change.

---

### **# 5) 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?**

# You can use methods like .sort() .max() .count() ... so on to perform different operations with the data structures.

---

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

```
list1 = [x for x in range(3)]
```

```
list1.append(69)
```

---

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

```
list1 = [["a","b","c"],["d","e","f"]]  
print(list1[1][0])
```

```
d
```

---

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

```
tup = (3,2,1)  
print(len(tup))
```

```
3
```

---

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

```
mahSet = {1,2,3,4}  
mahSet.add(69420)
```

---

**# 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.**

```
mahDict = {"Pam" : 69, "Sam": 420}  
print(mahDict["Pam"])
```

```
69
```

---

**# 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.**

```
uin=[]
for i in range(5):
    uin.append(int(input("Enter element to add in the list ")))
filteredArray = ["over" if x > 100 else x for x in uin]
print(filteredArray)
```

```
Enter element to add in the list 100
Enter element to add in the list 200
Enter element to add in the list 300
Enter element to add in the list 10
Enter element to add in the list 20
[100, 'over', 'over', 10, 20]
```

---

**# 7) Write a Python script to concatenate the following dictionaries to create a new one.**

```
dict1={ 1:10, 2:20}
dict2={ 3:30, 4:40}
dict3={ 5:50, 6:60}
print(**dict1,**dict2,**dict3)
```

```
{ 1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
```

---

**# 8) 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'}
]
```

```
key = "address"
value = "New York"
found = False
for dictionary in students:
    if key in dictionary:
        if dictionary[key] == value:
            print(f"key: {key} and value: {value} exist")
            found = True
if not found:
```

```
print(f"key: {key} and value: {value} don't exist")
```

```
key: address and value: New York don't exist
```

---

**# Part 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'.**

```
print([element for element in input("Enter strings with spaces: ").split() if "a" in element])
```

```
Enter strings with spaces: aaa b c d aeu  
['aaa', 'aeu']
```

---

**# 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.**

```
a = {1,2,3,4}  
b = {3,4,6}  
print({x for x in a if x in b})  
  
{3, 4}
```

---

**# 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.**

```
listofDicts = {"somedude":29,"someotherdude":38,"somegirl":65,"someothergirl":38}  
print({x:listofDicts[x] for x in listofDicts if listofDicts[x]>18})  
  
{'somedude': 29, 'someotherdude': 38, 'somegirl': 65, 'someothergirl': 38}
```

---

**# 4) Write a program to input your friend's names and their Phone Numbers and store them in the dictionary as the key-value pair.**

```
Perform the following operations on the dictionary:  
#a)Display the name and phone number of all your friends  
#b)Add a new k:v in this dictionary and display the modified dictionary  
#c)Delete the particular friend from the dictionary  
#d)Modify the phone number of an existing friend  
#e)Check if a friend is present in the dictionary or not and display it in sorted form.  
  
mahDict = { }  
for i in range(3):  
    key,value = input("Enter name and phonenum with space: ").split()
```

```

mahDict[key] = value
for element in mahDict:
    print(element)
mahDict["meow"] = "696969"
mahDict["meow"] = "595959"
print("Exists") if "meow" in mahDict else print("No Exists")

```

```

Enter name and phonenum with space: t 8888
Enter name and phonenum with space: l 888      9999
Enter name and phonenum with space: z 1000
t
l
z
Exists

```

---

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

```

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

for i in range(3):
    max_item = max(mahDict, key=mahDict.get) # mahDict.get gets values turns out ._. by
    default its the keys
    print({max_item: mahDict[max_item]})
    del mahDict[max_item]

{'item4': 55}
{'item1': 45.5}
{'item3': 41.3}

```

---

#### #6 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.

```

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]}
]

for student in students:
    average_grade = sum(student["grades"]) / len(student["grades"])

```

```
print(f"{student['name']}: {average_grade:.1f}")
```

```
Alice: 90.0  
Bob: 75.0  
Charlie: 95.0  
Diana: 85.0  
Emily: 95.0
```

---

**#7 Write a Python program to compute the difference between two lists.**

```
def list_difference(list1, list2):  
    return list(set(list1) - set(list2)), list(set(list2) - set(list1))  
  
list1 = ["red", "orange", "green", "blue", "white"]  
list2 = ["black", "yellow", "green", "blue"]  
diff1, diff2 = list_difference(list1, list2)  
  
print(diff1, diff2)  
  
['orange', 'white', 'red'] ['black', 'yellow']
```

---

**#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.**

```
def sort_books_by_author(books):  
    sorted_books = sorted(books, key= lambda book: book[1])  
    return sorted_books  
  
books = [("The Great Gatsby", "F. Scott Fitzgerald"), ("To Kill a Mockingbird", "Harper Lee"),  
         ("1984", "George Orwell")]  
  
sorted_books = sort_books_by_author(books)  
print(sorted_books)  
  
[('The Great Gatsby', 'F. Scott Fitzgerald'), ('1984', 'George Orwell'), ('To Kill a Mockingbird',  
'Harper Lee')]
```

---

**# Part 4 #1 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}
]

def add_book():
    title = input("Enter book title: ")
    author = input("Enter author's name: ")
    year = int(input("Enter the publication year: "))
    books.append({"title": title, "author": author, "year": year})

def remove_book():
    title = input("Enter the title of the book: ")
    for book in books:
        if book["title"] == title:
            books.remove(book)
            return
    print("Book not found.")

def search_by_title():
    title = input("Enter the title of the book: ")
    for book in books:
        if book["title"] == title:
            print(book)
            return
    print("Book not found.")

def search_by_author():
    author = input("Enter the author's name: ")
    for book in books:
        if book["author"] == author:
            print(book)
            return
    print("Book not found.")

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

while True:
    print("Library Dashboard Menu:".center(50))
    print("1. Add new book")
    print("2. Remove book")
    print("3. Search for a book title")
    print("4. Search for a book author")
```

```

print("5. List all the books")
print("6. Quit")
choice = input("Enter your choice: ")
if choice == "1":
    add_book()
elif choice == "2":
    remove_book()
elif choice == "3":
    search_by_title()
elif choice == "4":
    search_by_author()
elif choice == "5":
    list_all_books()
elif choice == "6":
    break
else:
    print("Invalid choice. Please try again.")

```

Library Dashboard Menu:

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

Enter your choice: 1

Enter book title: Som                      PewPew

Enter author's name: ugabuga

Enter the publication year: 6969

Library Dashboard Menu:

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

Enter your choice: 6

---

**#2 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.**

```

def play_game():
    matrix = [[1, 1, 1], [1, 1, 1], [1, 1, 1]]
    moves = 0
    while any(1 in row for row in matrix):
        print("Current Matrix:")
        for row in matrix:

```



```

        print(row)

    try:
        row = int(input("Enter the row index (0-2): "))
        col = int(input("Enter the column index (0-2): "))
    except ValueError:
        print("Invalid input. Please enter a number.")
        continue

    if matrix[row][col] == 'X':
        print("This cell has already been selected. Please choose again.")
        continue

    matrix[row][col] = 'X'
    moves += 1

    print(f"End! You made {moves} moves.")

play_game()

```

Current Matrix:

[1, 1, 1]

[1, 1, 1]

[1, 1, 1]

Enter the row index (0-2): 0

Enter the column index (0-2): 0

Current Matrix:

['X', 1, 1]

[1, 1, 1]

[1, 1, 1]

Enter the row index (0-2): 0

Enter the column index (0-2): 1

Current Matrix:

['X', 'X', 1]

[1, 1, 1]

[1, 1, 1]

Enter the row index (0-2): 0

Enter the column index (0-2): 2

Current Matrix:

['X', 'X', 'X']

[1, 1, 1]

[1, 1, 1]

Enter the row index (0-2): 1

Enter the column index (0-2): 0

Current Matrix:

['X', 'X', 'X']

['X', 1, 1]

[1, 1, 1]

Enter the row index (0-2): 1

Enter the column index (0-2): 1

Current Matrix:

['X', 'X', 'X']

['X', 'X', 1]

[1, 1, 1]

```
Enter the row index (0-2): 1
Enter the column index (0-2): 2
Current Matrix:
['X', 'X', 'X']
['X', 'X', 'X']
[1, 1, 1]
Enter the row index (0-2): 2
Enter the column index (0-2): 0
Current Matrix:
['X', 'X', 'X']
['X', 'X', 'X']
['X', 1, 1]
Enter the row index (0-2): 2
Enter the column index (0-2): 1
Current Matrix:
['X', 'X', 'X']
['X', 'X', 'X']
['X', 'X', 1]
Enter the row index (0-2): 2
Enter the column index (0-2): 2
End! You made 9 moves.
```

---