data structures in python

```
#Creating list with same data type
a = [1,2,3,4,5,6,7,8,9,10]
print(a)
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

```
#creating list with differat data type
b = [1,2.5,'manju']
print(b)
```

```
[1, 2.5, 'manju']
```

list operations

"Accessing Items"

```
print(a[8])
print(b[-1])
```

```
10
manju
```

modifying **Items**

```
a[3]=12
print(a)
```

```
[1, 2, 3, 12, 5, 6, 7, 8, 9, 10]
```

```
b[2] = 'manju'
print(b)
```

```
[1, 2.5, 'manju']
```

adding items

```
#append
a.append(10)
a
```

```
[1, 2, 3, 12, 5, 7, 8, 9, 10, 10, 10, 10, 10]
```

```
#insert
b.insert(6,'manju')
b

[1, 2.5, 'manju', 'manju', 'manju', 'manju']
```

Removing items

```
#remove
a.remove(10)
a

[1, 2, 3, 12, 5, 7, 8, 9, 10]

#pop
b.pop(2)
b

[1, 2.5, 'manju', 'manju']
```

Other operations

```
#len
len(a)

9

a.sort()
a.reverse()
a

[77, 77, 66, 6, 5, 3, 3, 2, 1]
```

Iterating through a list

```
a = (1,3,5,76,56,5,9,4)
for i in a:
  print(i)

1
3
5
76
56
5
9
4
```

Tuples

```
a = ( 1,2,3,4,'hello',5.5)
print(a[0])
print(a[-1])
```

```
1
5.5
```

```
#accessing items in a tuple
print(a[3])
```

```
4
```

Dictionary

```
student ={
    "name": "manju",
    "age": 18,
    "grade": "A",
    "city": "new york"
}
print(student)
```

```
{'name': 'manju', 'age': 18, 'grade': 'A', 'city': 'new york'}
```

Accessing and modifying Items

```
#Accessing
print(student["age"])
```

```
18
```

```
#Modifying
student["age"] = 21
print(student)
```

```
{'name': 'manju', 'age': 21, 'grade': 'A', 'city': 'new york'}
```

```
#adding
student["gender"] = "male"
print(student)
```

```
{'name': 'manju', 'age': 21, 'grade': 'A', 'city': 'new york',
'gender': 'male'}
```

```
#removing
del student["grade"]
print(student)
```

```
{'name': 'manju', 'age': 21, 'city': 'new york', 'gender': 'male'}
```

Iterating Thought a Dictionary

```
for key, value in student.items():
  print(key,value)
```

```
name manju
age 21
city new york
gender male
```

Set

Creating a set

```
#creating a set
num = {1,2,3,4,5,6,7,8,9}
print(num)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9}
```

Set Operations

```
#Adding Items
num.add(10)
print(num)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

```
#Removing Items
num.remove(8)
print(num)
```

```
{1, 2, 3, 4, 5, 6, 7, 9, 10}
```

```
#set operations
#union
a = {1,2,3,4,5}
b = {4,5,6,7,8,}
a | b
```

```
{1, 2, 3, 4, 5, 6, 7, 8}
```

```
#intersection
a & b
```

```
{4, 5}
```

```
#Difference
a - b
```

```
{1, 2, 3}
```

Hands on practice

```
#manipulating Lists
fruits = ['apple','banana','orange','cherry']
fruits.append('mango')
fruits.remove('banana')
print(fruits)
```

```
['apple', 'orange', 'cherry', 'mango']
```

```
#Creating a Dictionary
book = {
    "title": "Python Basics",
    "author": "John Doe",
    "year": 2023

}
print(book["title"])
book["year"]=2024
print(book)
```

```
Python Basics
{'title': 'Python Basics', 'author': 'John Doe', 'year': 2024}
```

```
#Working with Sets
set1 = {1,2,3,4}
set2 = {3,4,5,6}
print("Union:",set1 | set2)
print("Intersection:",set1 & set2)
print("Difference:",set1 - set2)
```

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

problems **solving**

```
#Merage Two Lists
list1 = [1,2,3]
list2 = [4,5,6]
merged_list = list1 + list2
print("Merged List:",merged_list)
```

```
Merged List: [1, 2, 3, 4, 5, 6]
```

```
#Dictionary Operations
student = {"name": "john","age":21, "marks":85}
print("Name:",student["name"])
student["marks"] = 90
print("Updated Marks:",student["marks"])
```

```
Name: john
Updated Marks: 90
```

```
#find the maximum and minimum in a list
num = [10,20,30,40,50,]
print("Maximum:",max(num))
print("Minimum:",min(num))
```

```
Maximum: 50
Minimum: 10
```

```
#Count Frequency of Elements

numbers = [1,2,2,3,3,3,4,4,4,4]
frequency = {}
```

find wheather it is **palindrome**

```
number = int(input("Enter a number:"))
reverse_number = 0
temp = number

while temp > 0:
  digit = temp % 10
  reverse_number = reverse_number * 10 + digit
  temp //= 10

if number == reverse_number:
  print(f"{number} is a palindrome")
else:
  print(f"{number} is not a palindrome")
```

```
Enter a number:212
212 is a polindrome
```

```
#palindrome
number = input("Enter a number: ")
if number == number[::-1]:
  print(f"{number} is a  palindrone")
else:
  print(f"{number} is not a palindrome")
```

```
Enter a number: nayan
nayan is a  polindrone
```

```python
class Solution(object):
    def isPalindrome(self, x):
        """
        :type x: int
        :rtype: bool
        """
        if x < 0 or (x % 10 == 0 and x != 0):
            return False

        reversed_half = 0
        while x > reversed_half:

            reversed_half = reversed_half * 10 + x % 10
            x //= 10


        return x == reversed_half or x == reversed_half // 10


solution = Solution()
print(solution.isPalindrome(121))
print(solution.isPalindrome(-121))
print(solution.isPalindrome(10))
print(solution.isPalindrome(0))
```

```
True
False
False
True
```

```python
fruits = ['apple', 'banana', 'cherry']
print(fruits[0])
```

```
apple
```

pretty print

```python
import pprint
data = {"name": "Alice", "subjects": ["Math", "Science"], "grades":
{"Math": "A",
"Science": "B"}}
pprint.pprint(data)
```

```
{'grades': {'Math': 'A', 'Science': 'B'},
 'name': 'Alice',
 'subjects': ['Math', 'Science']}
```

practice problems

```python
#basic list problems
favorite_movies = ["Inception", "The Dark Knight", "Interstellar",
"The Matrix"]
new_movie = "Pulp Fiction"
favorite_movies.append(new_movie)
print("After adding a new movie:", favorite_movies)
movie_to_remove = "The Matrix"
if movie_to_remove in favorite_movies:
    favorite_movies.remove(movie_to_remove)
print("After removing a movie:", favorite_movies)
specific_movie = "Inception"
if specific_movie in favorite_movies:
    index = favorite_movies.index(specific_movie)
    print(f"The index of '{specific_movie}' is:", index)
else:
    print(f"'{specific_movie}' is not in the list.")
```

```
After adding a new movie: ['Inception', 'The Dark Knight',
'Interstellar', 'The Matrix', 'Pulp Fiction']
After removing a movie: ['Inception', 'The Dark Knight',
'Interstellar', 'Pulp Fiction']
The index of 'Inception' is: 0
```

```python
#magic 8 ball Extension
import random

# Magic 8 Ball responses
responses = [
    "It is certain.", "It is decidedly so.", "Without a doubt.",
    "Yes – definitely.", "You may rely on it.", "As I see it, yes.",
    "Most likely.", "Outlook good.", "Yes.", "Signs point to yes.",
    "Reply hazy, try again.", "Ask again later.",
    "Better not tell you now.", "Cannot predict now.",
    "Concentrate and ask again.", "Don't count on it.",
    "My reply is no.", "My sources say no.",
    "Outlook not so good.", "Very doubtful."
]

print("Welcome to the Magic 8 Ball!")
print("You can ask any question or type 'quit' to exit.")

while True:
    # Get the user's question
    question = input("\nWhat is your question? ").strip()

    # Check if the user wants to quit
    if question.lower() == "quit":
        print("Goodbye! Come back soon!")
        break
    # Ensure the question is not empty
    if not question:
        print("Please ask a question!")
        continue

    # Provide a random response
    print("Magic 8 Ball says:", random.choice(responses))
```

```
Welcome to the Magic 8 Ball!
You can ask any question or type 'quit' to exit.

What is your question? quit
Goodbye! Come back soon!
```

```
#Dictionary Manipulation
library = {
    "1984": True,
    "To Kill a Mockingbird": True,
    "The Great Gatsby": True,
    "Moby Dick": False,
    "War and Peace": True
}


def is_book_available(book_title):
    return library.get(book_title, None)


def borrow_book(book_title):
    if book_title in library:
        if library[book_title]:
            library[book_title] = False
            print(f"You have successfully borrowed '{book_title}'.")
        else:
            print(f"'{book_title}' is currently unavailable.")
    else:
        print(f"'{book_title}' is not in the library.")


def return_book(book_title):
    if book_title in library:
        if not library[book_title]:
            library[book_title] = True
            print(f"Thank you for returning '{book_title}'.")
        else:
            print(f"'{book_title}' was not borrowed.")
    else:
        print(f"'{book_title}' is not in the library.")


print("Welcome to the library system!")
```

```
while True:
    print("\nOptions: 1. Check availability  2. Borrow book  3. Return
book  4. Exit")
    choice = input("Enter your choice: ").strip()

    if choice == "1":
        book = input("Enter the title of the book: ").strip()
        available = is_book_available(book)
        if available is None:
            print(f"'{book}' is not in the library.")
        elif available:
            print(f"'{book}' is available.")
        else:
            print(f"'{book}' is currently unavailable.")

    elif choice == "2":
        book = input("Enter the title of the book to borrow:
").strip()
        borrow_book(book)

    elif choice == "3":
        book = input("Enter the title of the book to return:
").strip()
        return_book(book)

    elif choice == "4":
        print("Goodbye!")
        break

    else:
        print("Invalid option. Please try again.")
```

```
Welcome to the library system!

Options: 1. Check availability  2. Borrow book  3. Return book  4.
Exit
Enter your choice: 4
Goodbye!
```

```
#Dictionary Manipulation
menu = {
    "Burger": 5.99,
    "Pizza": 8.99,
    "Pasta": 7.49,
    "Salad": 4.99,
    "Fries": 2.99,
    "Drink": 1.99,
}
def calculate_total(items):
    total = 0
```

```python
    unavailable_items = []

    for item in items:
        if item in menu:
            total += menu[item]
        else:
            unavailable_items.append(item)

    return total, unavailable_item
print("Welcome to our restaurant!")
print("Here is the menu:")
for item, price in menu.items():
    print(f"{item}: ${price:.2f}")

while True:
    print("\nEnter the items you'd like to order (comma-separated), or type 'exit' to quit.")
    order = input("Your order: ").strip()

    if order.lower() == "exit":
        print("Thank you for visiting! Goodbye!")
        break

    order_list = [item.strip() for item in order.split(",") if item.strip()]

    # Calculate total cost
    total_cost, unavailable = calculate_total(order_list)

    if unavailable:
        print(f"Sorry, these items are not on the menu: {', '.join(unavailable)}")

    print(f"Total cost for your order: ${total_cost:.2f}")
```

```
Welcome to our restaurant!
Here is the menu:
Burger: $5.99
Pizza: $8.99
Pasta: $7.49
Salad: $4.99
Fries: $2.99
Drink: $1.99

Enter the items you'd like to order (comma-separated), or type 'exit' to quit.
Your order: exit
Thank you for visiting! Goodbye!
```