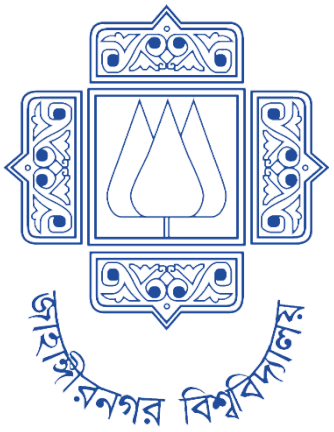
Institute of Information Technology (IIT)

Jahangirnagar University



Lab Report: 03

Submitted by:

Name: MD Zuleyenine Ibne Noman

Roll No: 2002

Lab Date:13-06-2023

Submission Date: 22-06-2023

Lab Report # Day 03

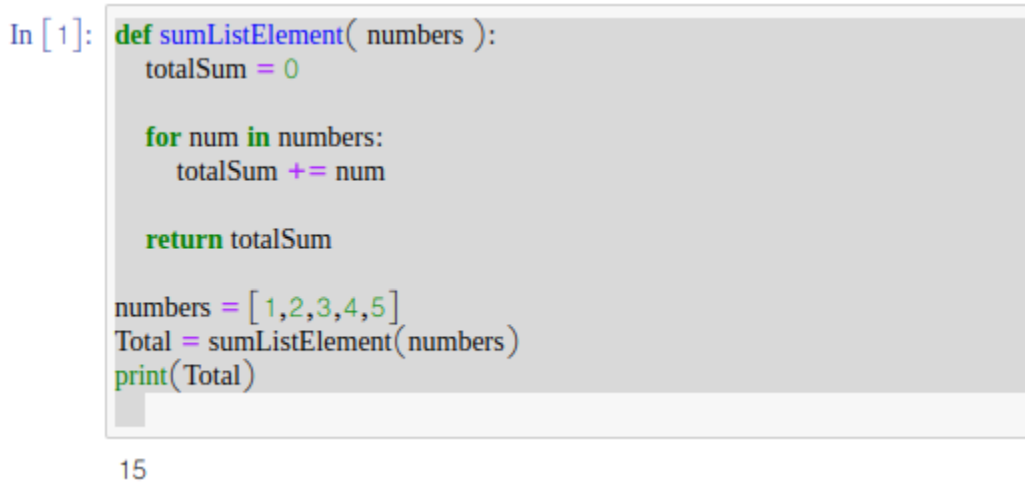
Example 1:

Problem Name: **Write a Python program to find the sum of all the elements in a list.**

Source Code:

```
def sumListElement( numbers ):
      totalSum = 0

      for num in numbers:
      totalSum += num

      return totalSum

numbers = [1,2,3,4,5]
Total = sumListElement(numbers)
print(Total)
```

Output Screen Shots:



Example :

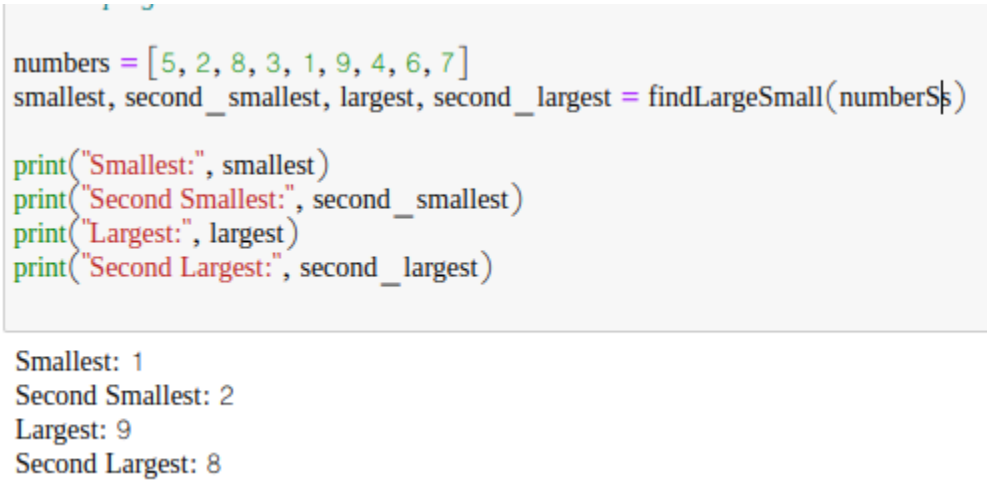
Problem Name: **Write a Python program to find the largest, smallest, second largest, and second smallest elements in a list.**

Source Code:

```
def findLargeSmall( numbers ):
      if len(numbers) < 2:
      return "List should contain at least 2 elements."

      smallest = min(numbers[0], numbers[1])
      secondSmallest = max(numbers[0], numbers[1])
      largest = secondSmallest
      secondLargest = smallest


      for num in numbers[2: ]:
      if num < smallest:
            secondSmallest = smallest
            smallest = num
      elif num < secondSmallest:
            secondSmallest = num
      if num > largest:
            secondLargest = largest
            largest = num
      elif num > secondLargest:
            secondLargest = num
      return smallest,secondSmallest,largest,secondLargest


#main program

numbers = [5, 2, 8, 3, 1, 9, 4, 6, 7]
smallest, second_smallest, largest, second_largest =
findLargeSmall(numbers)
```

```
print("Smallest:", smallest)
print("Second Smallest:", second_smallest)
print("Largest:", largest)
print("Second Largest:", second_largest)
```

Output Screen Shots:



Example 3:

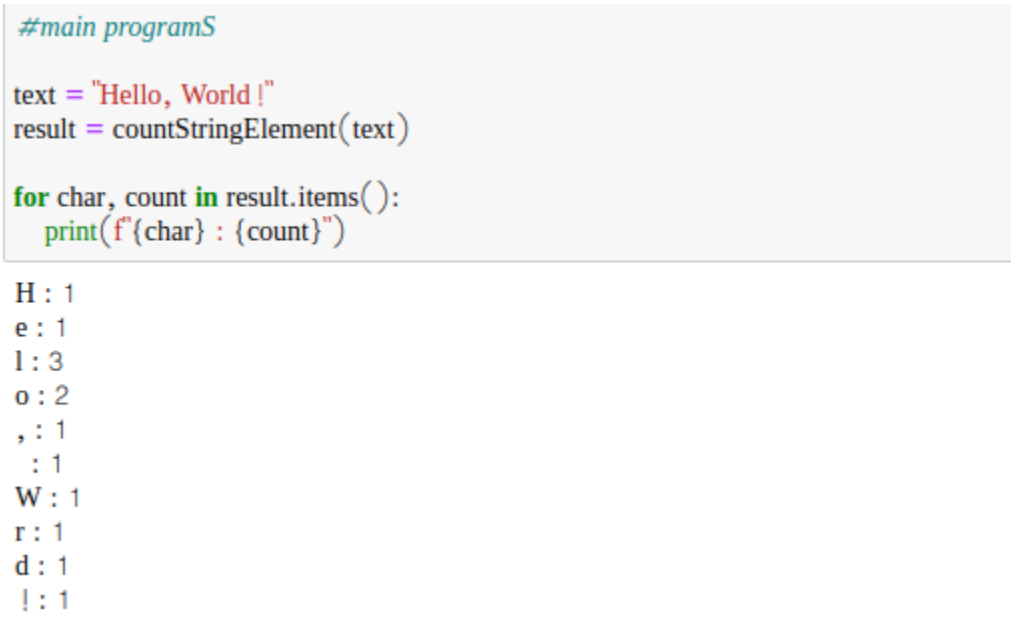
Problem Name: **Write a Python program to count the number of occurrences of each character in a string.**

Source Code:

```
def countStringElement(string):
      countChar = {}

      for char in string:
      if char in countChar:
            countChar[char] += 1
      else:
```

```
            countChar[char] = 1
      return countChar

#main program

text = "Hello, World!"
result = countStringElement(text)

for char, count in result.items():
      print(f"{char} : {count}")
```

Output Screen Shots:



```
#main programS

text = "Hello, World !"
result = countStringElement(text)

for char, count in result.items():
   print(f"{char} : {count}")

H : 1
e : 1
l : 3
o : 2
, : 1
  : 1
W : 1
r : 1
d : 1
! : 1
```

Example 4:

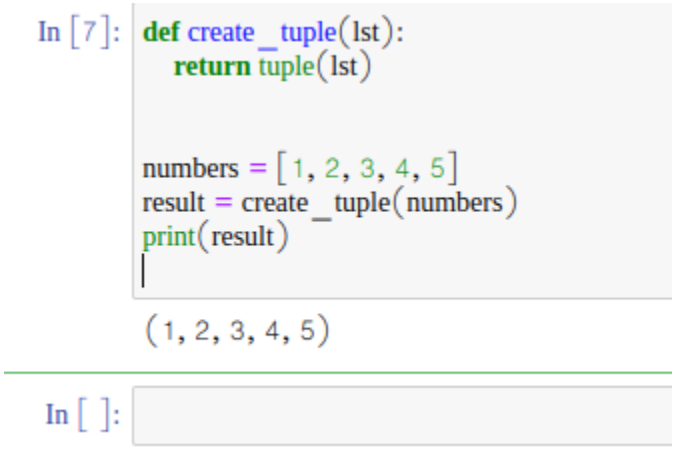
Problem Name: **Write a Python program to create a tuple with elements from a list and print it.**

Source Code:

```
def create_tuple(lst):
      return tuple(lst)
```

```
numbers = [1, 2, 3, 4, 5]
result = create_tuple(numbers)
print(result)
```

Output Screen Shots:



Example 5:

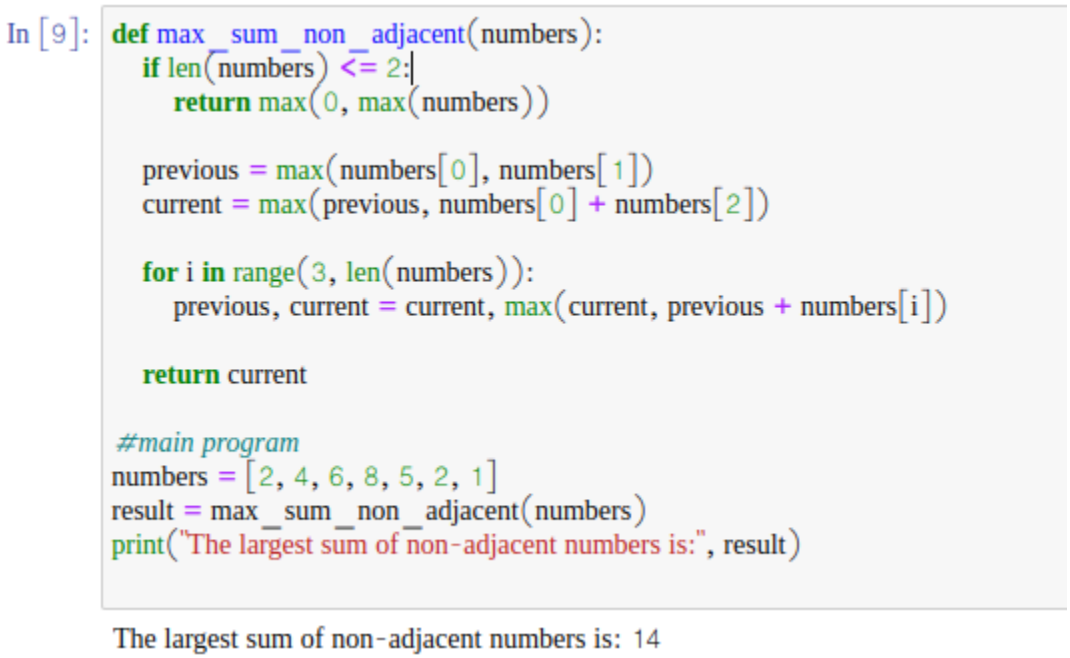
Problem Name: **Write a Python function that takes a list of numbers as input and returns the largest sum of non-adjacent numbers**.

Source Code:

```
def max_sum_non_adjacent(numbers):
    if len(numbers) <= 2:
    return max(0, max(numbers))

    previous = max(numbers[0], numbers[1])
    current = max(previous, numbers[0] + numbers[2])

    for i in range(3, len(numbers)):
    previous, current = current, max(current, previous + numbers[i])

    return current
```

```
#main program
numbers = [2, 4, 6, 8, 5, 2, 1]
result = max_sum_non_adjacent(numbers)
print("The largest sum of non-adjacent numbers is:", result)
```

Output Screen Shots:



Example 6:

Problem Name: **Write a Python program to remove duplicates from a list and return the resultant list.**

Source Code:

```
def remove_duplicates(lst):
      return list(set(lst))

# Example usage
numbers = [1, 2, 3, 2, 4, 1, 5]
result = remove_duplicates(numbers)
print(result)
```

Output Screen Shots:

```
In [10]: def remove_duplicates(lst):
             return list(set(lst))

         # Example usage
         numbers = [1, 2, 3, 2, 4, 1, 5]
         result = remove_duplicates(numbers)
         print(result)

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

Example 7:

Problem Name: **Write a Python program to find the common elements between two lists and return the resultant list.**

Source Code:

```
def findCommonElements(list1, list2):
      common_elements = []
      for element in list1:
      if element in list2 and element not in common_elements:
            common_elements.append(element)
      return common_elements


list1 = [1, 2, 3, 4, 5]
list2 = [4, 5, 6, 7, 8]
result = findCommonElements(list1, list2)
print(result)
```

Output Screen Shots:

```
In [11]: def findCommonElements(list1, list2):
             common_elements = []
             for element in list1:
                 if element in list2 and element not in common_elements:
                     common_elements.append(element)
             return common_elements


         list1 = [1, 2, 3, 4, 5]
         list2 = [4, 5, 6, 7, 8]
         result = findCommonElements(list1, list2)
         print(result)

         [4, 5]
```

Example 8:

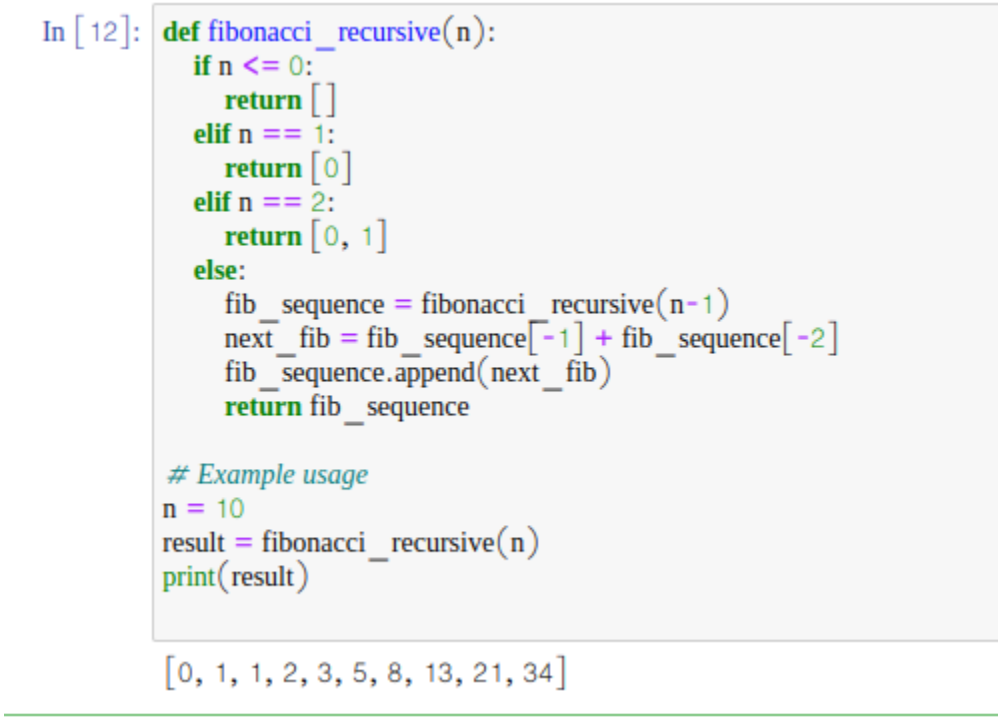
Problem Name: **Write a Python program to find the first n Fibonacci numbers using recursion.**

Source Code:

```
def fibonacci_recursive(n):
      if n <= 0:
      return []
      elif n == 1:
      return [0]
      elif n == 2:
      return [0, 1]
      else:
      fib_sequence = fibonacci_recursive(n-1)
      next_fib = fib_sequence[-1] + fib_sequence[-2]
      fib_sequence.append(next_fib)
      return fib_sequence
```

```
# Example usage
n = 10
result = fibonacci_recursive(n)
print(result)
```

Output Screen Shots:



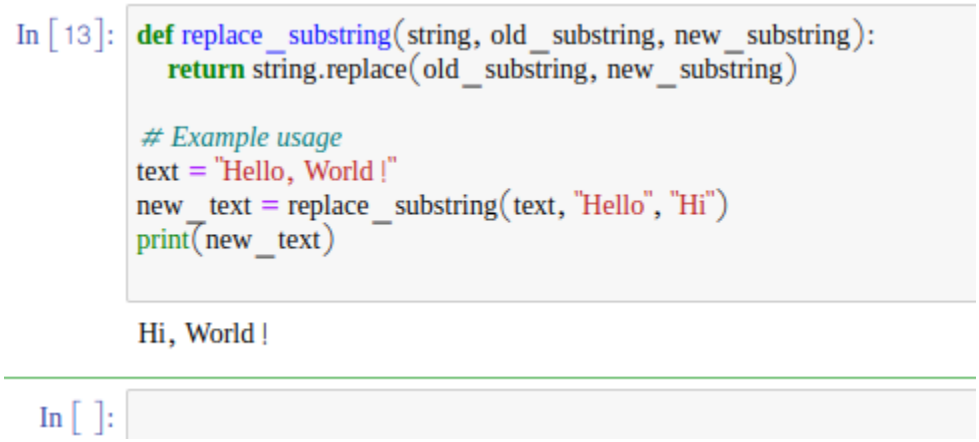
Example 9:

Problem Name: **Write a Python function to replace all occurrences of a substring in a string.**

Source Code:

```
def replace_substring(string, old_substring, new_substring):
      return string.replace(old_substring, new_substring)

# Example usage
text = "Hello, World!"
new_text = replace_substring(text, "Hello", "Hi")
print(new_text)
```

Output Screen Shots:



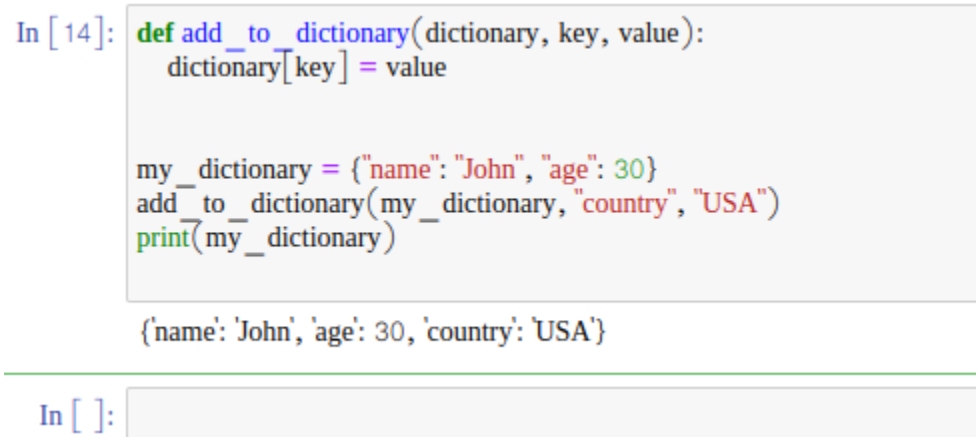
Example 10:

Problem Name: **Write a function to add a key-value pair to a dictionary in Python**

Source Code:

```
def add_to_dictionary(dictionary, key, value):
      dictionary[key] = value


my_dictionary = {"name": "John", "age": 30}
add_to_dictionary(my_dictionary, "country", "USA")
print(my_dictionary)
```

Output Screen Shots:



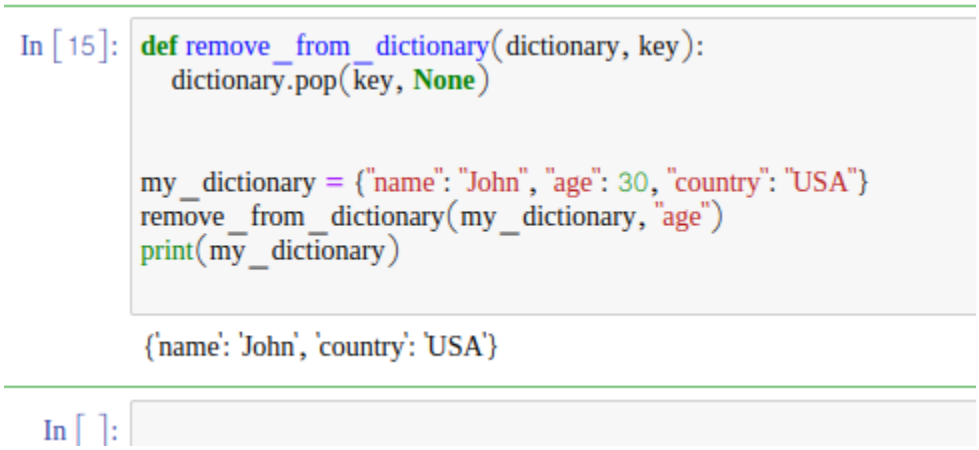
Example 11:

Problem Name: **Write a function to remove a key from a dictionary**

Source Code:

```
def remove_from_dictionary(dictionary, key):
      dictionary.pop(key, None)


my_dictionary = {"name": "John", "age": 30, "country": "USA"}
remove_from_dictionary(my_dictionary, "age")
print(my_dictionary)
```

Output Screen Shots:



Example 12:

Problem Name: **Write a function to reverse a list of numbers.**

Source Code:

```
def reverseList(numbers):
      return numbers[::-1]


numbers = [1, 2, 3, 4, 5]
reversed_numbers = reverseList(numbers)
print(reversed_numbers)
```

Output Screen Shots:

```
In [16]: def reverse_list(numbers):
             return numbers[::-1]

         # Example usage
         numbers = [1, 2, 3, 4, 5]
         reversed_numbers = reverse_list(numbers)
         print(reversed_numbers)

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

Example 13:

Problem Name: **Write a Python program to find and print the key with the maximum value in a dictionary.**

Source Code:

```
def find_key_with_maximum_value(dictionary):
      if not dictionary:
      return None
      max_value = max(dictionary.values())
      for key, value in dictionary.items():
      if value == max_value:
            return key

my_dictionary = {"apple": 20, "banana": 30, "orange": 15, "grape": 30}
max_value_key = find_key_with_maximum_value(my_dictionary)
print("Key with the maximum value:", max_value_key)
```

Output Screen Shots:

```
In [17]: def find_key_with_maximum_value(dictionary):
             if not dictionary:
                 return None
             max_value = max(dictionary.values())
             for key, value in dictionary.items():
                 if value == max_value:
                     return key

         my_dictionary = {"apple": 20, "banana": 30, "orange": 15, "grape": 30}
         max_value_key = find_key_with_maximum_value(my_dictionary)
         print("Key with the maximum value:", max_value_key)
```

Key with the maximum value: banana

Example 14:

Problem Name: **Write a Python program to merge two dictionaries and create a new dictionary.**

Source Code:

```
def merge_dictionaries(dict1, dict2):
      merged_dict = dict1.copy()
      merged_dict.update(dict2)
      return merged_dict

# Example usage
dict1 = {"apple": 2, "banana": 3}
dict2 = {"orange": 4, "grape": 5}
merged_dict = merge_dictionaries(dict1, dict2)
print(merged_dict)
```

Output Screen Shots:

```
In [18]: def merge_dictionaries(dict1, dict2):
             merged_dict = dict1.copy()
             merged_dict.update(dict2)
             return merged_dict

         # Example usage
         dict1 = {"apple": 2, "banana": 3}
         dict2 = {"orange": 4, "grape": 5}
         merged_dict = merge_dictionaries(dict1, dict2)
         print(merged_dict)

         {'apple': 2, 'banana': 3, 'orange': 4, 'grape': 5}
```

Example 15:

Problem Name: **Given a list of dictionaries, you want to sort them based on a specific key 'age' in each dictionary. Write a lambda function as the key parameter in the sorted() function to achieve this.**

Source Code:

```
people = [
      {"name": "John", "age": 25},
      {"name": "Alice", "age": 30},
      {"name": "Bob", "age": 20},
]

sorted_people = sorted(people, key=lambda x: x['age'])

print(sorted_people)
```

Output Screen Shots:

```
[19]: people = [
        {"name": "John", "age": 25},
        {"name": "Alice", "age": 30},
        {"name": "Bob", "age": 20},
      ]

      sorted_people = sorted(people, key=lambda x: x['age'])

      print(sorted_people)
```

```
[{'name': 'Bob', 'age': 20}, {'name': 'John', 'age': 25}, {'name': 'Alice', 'age': 30}]
```