Python Interview Questions - Nikhil Sharma - KirkYagami 🧑‍💻🕵️

1. Write a python program to find the 2nd maximum value in a list without using any in-built function.

```python
numbers = [23, 45, 78, 100, 3, -3, 700, 8, 9000]

first_max = float('-inf')

for n in numbers:
    if n > first_max:
        first_max = n

second_max = float('-inf')

for n in numbers:
    if n > second_max and n!=first_max:
        second_max = n

print(second_max) # Output: 700
```

if asked to write a function:

```python
numbers = [23, 45, 78, 100, 3, -3, 700, 8, 9000]

def second_max(numbers): # function
    first_max = float('-inf')

    for n in numbers:
        if n > first_max:
            first_max = n

    second_max = float('-inf')

    for n in numbers:
        if n > second_max and n!=first_max:
            second_max = n
    return second_max


print(second_max(numbers)) # function_call, Output: 700
```

2. Write a python program to reverse a list without using any in-built function or slicing.

```
nums = [23, 45, 78, 100, 3, -3, 700, 8, 9000]


rev_num = []
for i in range(1, len(nums)+1):
    rev_num.append(nums[-i])
rev_num

#output: [9000, 8, 700, -3, 3, 100, 78, 45, 23]
```

3. Write a python program to sort the list without using any function

```
nums = [23, 45, 78, 100, 3, -3, 700, 8, 9000]

for i in range(len(nums)):
    for j in range(i+1, len(nums)):
        if nums[i] > nums[j]:
            temp = nums[j]
            nums[j] = nums[i]
            nums[i] = temp

print(nums) #output: [-3, 3, 8, 23, 45, 78, 100, 700, 9000]
```

If you are asked to write a function for this:

```
nums = [23, 45, 78, 100, 3, -3, 700, 8, 9000]


def asc_sort(nums):
    for i in range(len(nums)):
        for j in range(i+1, len(nums)):
            if nums[i] > nums[j]:
                temp = nums[j]
                nums[j] = nums[i]
                nums[i] = temp
    return nums

asc_sort(nums) # function call
```

Python Interview Coding Questions - Nikhil Sharma | KirkYagami

Question 1: Write a Python program to check if a string is a palindrome.

Solution:

```python
def is_palindrome(string):
    reversed_string = string[::-1]
    return string == reversed_string

# Test the function
word = "madam"
if is_palindrome(word):
    print(f"{word} is a palindrome")
else:
    print(f"{word} is not a palindrome")
```

Question 2: Write a Python program to find the factorial of a number.

Solution:

```python
def factorial(n):
    if n == 0:
        return 1
    else:
        return n * factorial(n-1)

# Test the function
number = 5
result = factorial(number)
print(f"The factorial of {number} is {result}")
```

Question 3: Write a Python program to find the largest element in a list.

Solution:

```python
def find_largest(numbers):
    largest = numbers[0]
    for num in numbers:
        if num > largest:
            largest = num
    return largest

# Test the function
nums = [10, 5, 8, 20, 3]
```

```python
largest_num = find_largest(nums)
print(f"The largest number is {largest_num}")
```

Question 4: Write a Python program to reverse a string.

Soltuion:

```python
def reverse_string(string):
    return string[::-1]

# Test the function
text = "Hello, World!"
reversed_text = reverse_string(text)
print(reversed_text)
```

Question 5: Write a Python program to count the frequency of each element in a list.

Solution:

```python
def count_frequency(numbers):
    frequency = {}
    for num in numbers:
        if num in frequency:
            frequency[num] += 1
        else:
            frequency[num] = 1
    return frequency

# Test the function
nums = [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
frequency_count = count_frequency(nums)
print(frequency_count)
```

Question 6: Write a Python program to check if a number is prime.

Solution:

```python
def is_prime(number):
    if number < 2:
        return False
    for i in range(2, int(number**0.5) + 1):
        if number % i == 0:
            return False
    return True

# Test the function
num = 17
```

```python
if is_prime(num):
    print(f"{num} is a prime number")
else:
    print(f"{num} is not a prime number")
```

Question 7: Write a Python program to find the common elements between two lists.

Solution:

```python
def find_common_elements(list1, list2):
    common_elements = []
    for item in list1:
        if item in list2:
            common_elements.append(item)
    return common_elements

# Test the function
list_a = [1, 2, 3, 4, 5]
list_b = [4, 5, 6, 7, 8]
common = find_common_elements(list_a, list_b)
print(common)
```

Question 8: Write a Python program to sort a list of elements using the bubble sort algorithm.

Solution:

```python
def bubble_sort(elements):
    n = len(elements)
    for i in range(n - 1):
        for j in range(n - i - 1):
            if elements[j] > elements[j + 1]:
                elements[j], elements[j + 1] = elements[j + 1], elements[j]

# Test the function
nums = [5, 2, 8, 1, 9]
bubble_sort(nums)
print(nums)
```

Question 9: Write a Python program to find the second largest number in a list.

Solution:

```python
def find_second_largest(numbers):
    largest = float('-inf')
    second_largest = float('-inf')
    for num in numbers:
        if num > largest:
            second_largest = largest
            largest = num
        elif num > second_largest and num != largest:
            second_largest = num
    return second_largest

# Test the function
nums = [10, 5, 8, 20, 3]
second_largest_num = find_second_largest(nums)
print(f"The second largest number is {second_largest_num}")
```

Question 10: Write a Python program to remove duplicates from a list.

Solution:

```python
def remove_duplicates(numbers):
    unique_numbers = []
    for num in numbers:
        if num not in unique_numbers:
            unique_numbers.append(num)
    return unique_numbers

# Test the function
nums = [1, 2, 3, 2, 1, 3, 2, 4, 5, 4]
unique_nums = remove_duplicates(nums)
print(unique_nums)


# OR

def remove_dupl(a):
        return list(set(a))
```

11. Find missing number in list

```python
nums = [1, 2, 4, 5, 6]

def find_missing_number(nums):
    n = len(nums) + 1
    expected_sum = n * (n + 1) // 2
    actual_sum = sum(nums)
    return expected_sum - actual_sum

print(f"Missing number: {find_missing_number(nums)}")
```

12. Find second occurrence

```python
fruits = ['apple', 'banana', 'cherry', 'apple']

def find_second_occurrence(lst, item):
    first_index = lst.index(item)
    return lst.index(item, first_index + 1)

print(f"Index of second 'apple': {find_second_occurrence(fruits, 'apple')}")
```

13. Count frequency of each element

```python
def count_frequency(lst):
    return {item: lst.count(item) for item in set(lst)}

print(f"Frequency of elements: {count_frequency(fruits)}")
```

14. Convert negative number into positive

```python
def to_positive(num):
    return abs(num)

print(f"Positive of -5: {to_positive(-5)}")
```

---

1. Write a Python program to find the second largest element in a list without using the built-in max() function.

2.Create a Python program that takes a list of dictionaries (with keys like name, age) and sorts them by age.

3.Implement a Python function to merge two sorted lists into one sorted list without using any built-in sorting functions.

4. You need to write a program that divides a list of student names into a specified number of groups. The program should include a predefined list of student names and a specified number of groups. It should then generate a dictionary where each key represents a group name (e.g., "Group1", "Group2", etc.), and the corresponding value is a tuple containing the

names of the students assigned to that group. Ensure that the students are distributed as evenly as possible, with any remaining students included in the last group. Please provide a Python script to implement this functionality, ensuring that both the original list of student names and the resulting dictionary of groups are printed.

1. Converting an Integer into Decimals

```
import decimal
integer = 10
print(decimal.Decimal(integer))
print(type(decimal.Decimal(integer)))

> 10
> <class 'decimal.Decimal'>
```

---

2. Converting an String of Integers into Decimals

```
import decimal
string = '12345'
print(decimal.Decimal(string))
print(type(decimal.Decimal(string)))

> 12345
> <class 'decimal.Decimal'>
```

---

3. Reversing a String using an Extended Slicing Technique

```
string = "Python Programming"
print(string[::-1])

> gnimmargorP nohtyP
```

“Pasted image 20240518014208.png” could not be found.

---

4. Counting Vowels in a Given Word

```
vowel = ['a', 'e', 'i', 'o', 'u']
word = "programming"
count = 0
for character in word:
    if character in vowel:
        count += 1
print(count)

> 3
```

---

5. Counting Consonants in a Given Word

```
vowel = ['a', 'e', 'i', 'o', 'u']
word = "programming"
count = 0
for character in word:
    if character not in vowel:
        count += 1
print(count)

> 8
```

---

6. Counting the Number of Occurances of a Character in a String

```
word = "python"
character = "p"
count = 0
for letter in word:
    if letter == character:
        count += 1
print(count)

> 1
```

7. Writing Fibonacci Series

```
fib = [0,1]
# Range starts from 0 by default
for i in range(5):
    fib.append(fib[-1] + fib[-2])

# Converting the list of integers to string
print(', '.join(str(e) for e in fib))

> 0, 1, 1, 2, 3, 5, 8
```

8. Finding the Maximum Number in a List

```
numberList = [15, 85, 35, 89, 125]

maxNum = numberList[0]
for num in numberList:
    if maxNum < num:
        maxNum = num
print(maxNum)

> 125
```

9. Finding the Minimum Number in a List

```
numberList = [15, 85, 35, 89, 125, 2]

minNum = numberList[0]
for num in numberList:
    if minNum > num:
        minNum = num
print(minNum)

> 2
```

10. Finding the Middle Element in a List

```
numList = [1, 2, 3, 4, 5]
midElement = int((len(numList)/2))

print(numList[midElement])

> 3
```

---

11. Converting a List into a String

```
lst = ["P", "Y", "T", "H", "O", "N"]
string = ''.join(lst)

print(string)
print(type(string))

> PYTHON
> <class 'str'>
```

---

12. Adding Two List Elements Together

```
lst1 = [1, 2, 3]
lst2 = [4, 5, 6]

res_lst = []
for i in range(0, len(lst1)):
    res_lst.append(lst1[i] + lst2[i])
print(res_lst)

> [5, 7, 9]
```

---

13. Comparing Two Strings for Anagrams

```
str1 = "Listen"
str2 = "Silent"

str1 = list(str1.upper())
```

```
str2 = list(str2.upper())
str1.sort(), str2.sort()

if(str1 == str2):
    print("True")
else:
    print("False")

> True
```

14. Checking for Palindrome Using Extended Slicing Technique

```
str1 = "Kayak".lower()
str2 = "kayak".lower()

if(str1 == str2[::-1]):
    print("True")
else:
    print("False")

> True
```

---

15. Counting the White Spaces in a String

```
string = "P r ogramm in g "
print(string.count(' '))

> 5
```

---

16. Counting Digits, Letters, and Spaces in a String

```
# Importing Regular Expressions Library
import re

name = 'Python is 1'
```

```
digitCount = re.sub("[^0-9]", "", name)
letterCount = re.sub("[^a-zA-Z]", "", name)
spaceCount = re.findall("[ \n]", name)

print(len(digitCount))
print(len(letterCount))
print(len(spaceCount))

> 1
> 8
> 2
```

17. Counting Special Characters in a String

```
# Importing Regular Expressions Library
import re
spChar = "!@#$%^&*()"

count = re.sub('[\w]+', '', spChar)
print(len(count))

> 10
```

18. Removing All Whitespace in a String

```
import re

string = "C O D E"
spaces = re.compile(r'\s+')
result = re.sub(spaces, '', string)
print(result)

> CODE
```

19. Building a Pyramid in Python

```
floors = 3
h = 2*floors-1
for i in range(1, 2*floors, 2):
    print('{:^{}}'.format('*'*i, h))

>  *
  ***
 *****
```

---

20. Randomizing the Items of a List in Python

```
from random import shuffle

lst = ['Python', 'is', 'Easy']
shuffle(lst)
print(lst)

> ['Easy', 'is', 'Python']
```

String Questions

1. Python program to remove character from string

```
str = "Python"
ch = "o"
print(str.replace(ch," ")) #output: Pyth n
```

---

2. Python Program to count occurrence of characters in string

```
string = "Python"
char = "y"
count = 0
for i in range(len(string)):
    if(string[i] == char):
        count = count + 1
```

3. Python program in to check string is anagrams or not

```
str1 = "python"
str2 = "yonthp"
if (sorted(str1) == sorted(str2)):
    print("Anagram")
else:
    print("Not an anagram")

> Anagram
```

4. Python program to check string is palindrome or not

```
string = "madam"
if(string == string[:: - 1]):
   print("Palindrome")
else:
   print("Not a Palindrome")

> Palindrome
```

5. Python code to check given character is digit or not

```
ch = 'a'
if ch >= '0' and ch <= '9':
        print("Digit")
else:
    print("Not a Digit")

> Not a Digit
```

6. Program to replace the string space with any given character

```python
string = "m d m"
result = ''
ch = "a"
for i in string:
        if i == ' ':
            i = ch
        result += i
print(result)

> madam
```