1.) Write a program to find the reverse of a given number using recursive.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

```
def rev_num(n, rev=0):
    if n == 0:
        return rev
    else:
        return rev_num(n // 10, rev * 10 + n % 10)
num = 12345
print("The reverse of the number is.....",rev_num(num))
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023,
14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for
more information.
>>>
= RESTART: C:\Users\Admin\AppData\Local\Progr
ams\Python\Python311\recursion and others.py
The reverse of the number is..... 54321
>>>
```

2.) Write a program to find the perfect number.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py

File Edit Format Run Options Window Help

```
a=int(input("Enter a number"))
s=0
for i in range(1,a):
    if a%i==0:
        s=s+i
    else:
        continue
if s==a:
    print("It is a perfect number")
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:
34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more i
nformation.
>>>
= RESTART: C:\Users\Admin\AppData\Local\Programs\Py
thon\Python311\recursion and others.py
Enter a number6
It is a perfect number
>>>
```

3.) Write C program that demonstrates the usage of these notations by analyzing the time complexity of some example algorithms.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

```python
#o(n)
def linear_search(data, value):
    for index in range(len(data)):
        if data[index] == value:
            return index
    return -1
data = [2, 4, 6, 8, 10]
value = 6
print(linear_search(data,value))

#o(n^2)
def bubble_sort(data):
    n = len(data)
    for i in range(n):
        for j in range(0, n - i - 1):
            if data[j] > data[j + 1]:
                data[j], data[j + 1] = data[j + 1], data[j]
data = [64, 34, 25, 12, 22, 11, 90]
bubble_sort(data)
print("the sorted array is.....",data)
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py
2
the sorted array is..... [11, 12, 22, 25, 34, 64, 90]
>>>
```

4.) Write C programs that demonstrate the mathematical analysis of non-recursive and recursive algorithms.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

```python
#time : o(n)
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result
num = 5
print("The factorial of the number is ",factorial(num))

#time: o(2^n)
def fibonacci(n):
    if n <= 1:
        return n
    else:
        return fibonacci(n - 1) + fibonacci(n - 2)
num = 10
print("The fibonacci series is ",fibonacci(num))
```

```
= RESTART: C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py
The factorial of the number is  120
The fibonacci series is  55
```

5.) Write C programs for solving recurrence relations using the Master Theorem, Substitution Method, and Iteration Method will demonstrate how to calculate the time complexity of an example recurrence relation using the specified technique.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

```python
def master_theorem(a, b, k):
  if a < b**k:
    return "O(log n^b)"
  elif a == b**k:
    return "O(n^k)"
  else:
    return "O(n^(log a / log b))"
recurrence = "T(n) = 2T(n/2) + n^2"
a, b, k = 2, 2, 2
time_complexity = master_theorem(a, b, k)
print("The time complexity of this using master theorem is ",time_complexity)

def iteration(recurrence, n):
  if recurrence == "T(n) = T(n-1 ) + n":
    solution = 0
    for i in range(n):
      solution =solution + i
    return solution
recurrence = "T(n) = T(n-1 ) + n"
n = 3
solution = iteration(recurrence, n)
```

```python
def substitution(recurrence, n):
  if recurrence == "T(n) = T(n-1) + 1":
    if n == 0:
      return 0
    else:
      return substitution(recurrence, n-1) + 1

recurrence = "T(n) = T(n-1) + 1"
n = 3
solution = substitution(recurrence, n)
print("The time complexity using substitution method for this recurrence ")
```

```
= RESTART: C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py
The time complexity of this using master theorem is  O(log n^b)
The solution of the respective recurrence using iteration is  3
The time complexity using substitution method for this recurrence
```

6.) Given two integer arrays nums1 and nums2, return an array of their Intersection. Each element in the result must be unique and you may return the result in any order.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

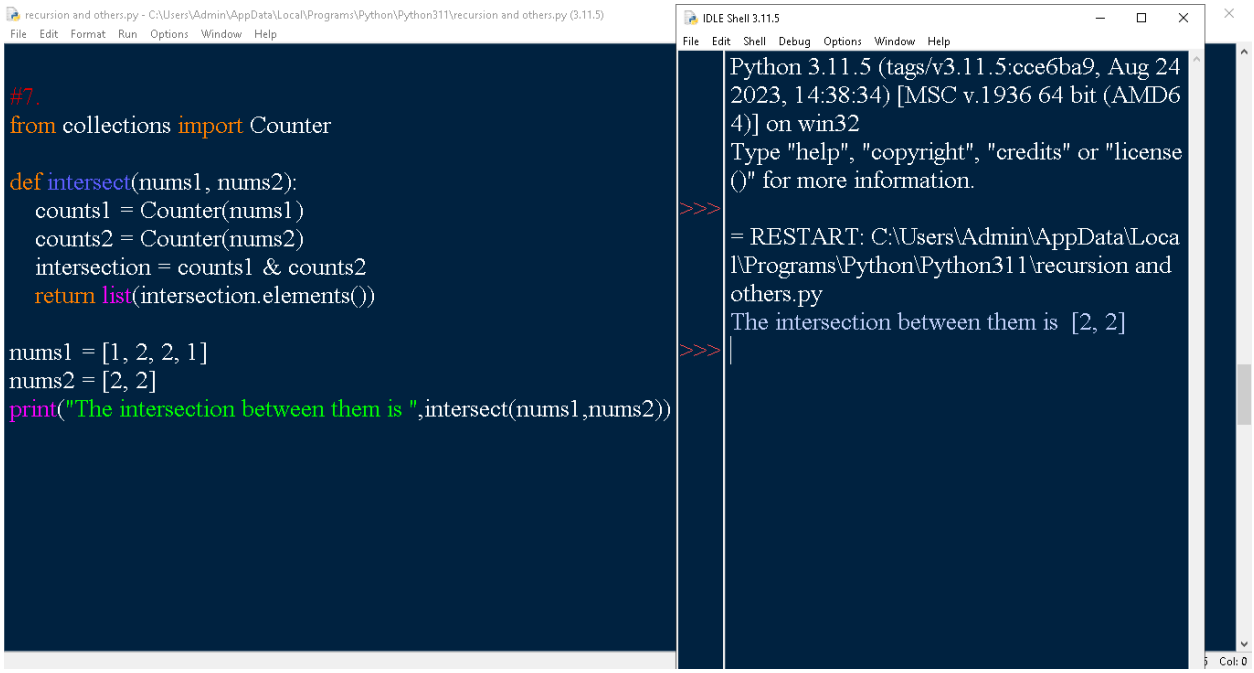
```python
#6.

def intersection(nums1, nums2):
    set1 = set(nums1)
    set2 = set(nums2)
    return list(set1 & set2)
nums1 = [1, 2, 2, 1]
nums2 = [2, 2]
print("The intersection of the two sets is ",intersection(nums1,nums2))
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>> 
= RESTART: C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py
The intersection of the two sets is  [2]
>>> 
```

7.) Given two integer arrays nums1 and nums2, return an array of their intersection. Each element in the result must appear as many times as it shows in both arrays and you may return the result in any order.



8.) Given an array of integers nums, sort the array in ascending order and return it.You must solve the problem without using any built-in functions in O(nlog(n)) time complexity and with the smallest space complexity possible.

```python
#8.
def quicksort(nums):
    if len(nums) <= 1:
        return nums
    pivot = nums[len(nums) // 2]
    left = [x for x in nums if x < pivot]
    middle = [x for x in nums if x == pivot]
    right = [x for x in nums if x > pivot]
    return quicksort(left) + middle + quicksort(right)
nums = [3,6,8,10,1,2,1]
print("The sorted array is",quicksort(nums))
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 2023
, 14:38:34) [MSC v.1936 64 bit (AMD64)] on win
32
Type "help", "copyright", "credits" or "license()" fo
r more information.
>>>
= RESTART: C:\Users\Admin\AppData\Local\Prog
rams\Python\Python311\recursion and others.py
The sorted array is [1, 1, 2, 3, 6, 8, 10]
>>>
```

9.) Given an array of integers nums, half of the integers in nums are odd, and the other half are even.

```python
def generate_array(n):
    return [i if i % 2 == 0 else i - 1 for i in range(2 * n)]
n = 10
print("The array with half odd and half even",generate_array(n))
```

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 24 202
3, 14:38:34) [MSC v.1936 64 bit (AMD64)] on wi
n32
Type "help", "copyright", "credits" or "license()" fo
r more information.
>>>
= RESTART: C:\Users\Admin\AppData\Local\Pro
grams\Python\Python311\recursion and others.py
The array with half odd and half even [0, 0, 2, 2, 4
, 4, 6, 6, 8, 8, 10, 10, 12, 12, 14, 14, 16, 16, 18, 1
8]
>>>
```

10.) Sort the array so that whenever nums[i] is odd, i is odd, and whenever nums[i] is even, i is

even. Return any answer array that satisfies this condition.

recursion and others.py - C:\Users\Admin\AppData\Local\Programs\Python\Python311\recursion and others.py (3.11.5)

File Edit Format Run Options Window Help

```python
def sort_array(nums):
    odd = [x for x in nums if x % 2 == 1]
    even = [x for x in nums if x % 2 == 0]
    result = []
    while odd or even:
        if even:
            result.append(even.pop())
        if odd:
            result.append(odd.pop())
    return result
nums = [4, 2, 5, 7]
print(f"Sorted array is: {sort_array(nums)}")
```

IDLE Shell 3.11.5

File Edit Shell Debug Options Window Help

```
Python 3.11.5 (tags/v3.11.5:cce6ba9, Aug 2
4 2023, 14:38:34) [MSC v.1936 64 bit (AM
D64)] on win32
Type "help", "copyright", "credits" or "licens
e()" for more information.
>>>
= RESTART: C:\Users\Admin\AppData\Loc
al\Programs\Python\Python311\recursion an
d others.py
Sorted array is: [2, 7, 4, 5]
>>>
```