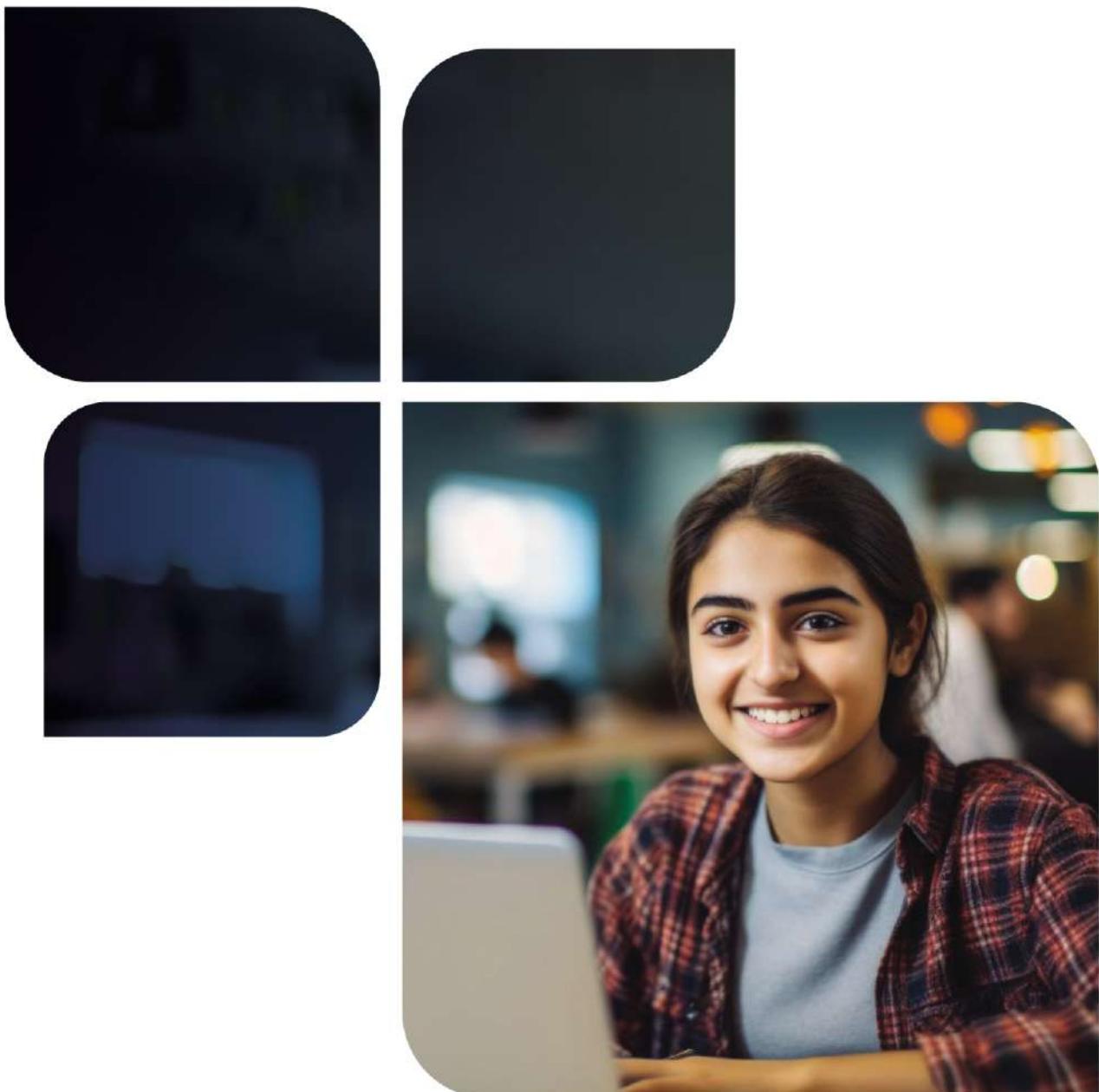


An ISO 9001 & ISO 21001
Certified Organization



The Quest
for your Dream Job
Ends Here!!

CORE PYTHON PROGRAMS

1. Write a Python Program to print "Hello World".

```
1.py
1     print("Hello World")
2

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python33\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/1.py
● Hello World
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

2. Write a Python Program to take user input and display it.

```
2.py
1     x = input()
2     print(x)
3

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python33\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/2.py
● GQT
GQT
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

3. Write a Python Program to swap two numbers.

```
3.py
1     a = int(input())
2     b = int(input())
3     a, b = b, a
4     print(a, b)
5

PROBLEMS   OUTPUT   DEBUG CONSOLE   TERMINAL   PORTS   QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python33\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/3.py
● 5
4
4 5
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

4. Write a Python Program to check if a number is even or odd.

```
4.py > ...
1 n = int(input())
2 print("Even" if n % 2 == 0 else "Odd")
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/4.py

- 4 Even
- 5 Odd
- PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

5. Write a Python Program to find the largest of three numbers.

```
5.py > ...
1 a = int(input())
2 b = int(input())
3 c = int(input())
4 print(max(a, b, c))
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/5.py

- 5
- 6
- 8
- PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

6. Write a Python Program to calculate the factorial of a number.

```
6.py > ...
1 n = int(input())
2 f = 1
3 for i in range(1, n + 1):
4     f *= i
5 print(f)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/6.py

- 5
- 120
- PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

7. Write a Python Program to generate the Fibonacci series.

```
7.py > ...
1 n = int(input())
2 a, b = 0, 1
3 for i in range(n):
4     print(a, end=" ")
5     a, b = b, a + b
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/7.py
● 5
○ 0 1 1 2 3
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

8. Write a Python Program to reverse a number.

```
8.py > ...
1 n = input()
2 print(n[::-1])
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/8.py
● 5
○ 0 1 1 2 3
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

9. Write a Python Program to check if a number is prime.

```
9.py > ...
1 n = int(input())
2 c = 0
3 for i in range(1, n + 1):
4     if n % i == 0:
5         c += 1
6 print("Prime" if c == 2 else "Not Prime")
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/9.py
5
Prime
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/9.py
14
Not Prime
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

10. Write a Python Program to find the sum of digits of a number.

```
10.py > ...
1 n = input()
2 s = 0
3 for i in n:
4     s += int(i)
5 print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/10.py
5
Prime
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/10.py
14
Not Prime
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> []

11. Write a Python Program to reverse a string.

```
11.py > ...
1 s = input()
2 print(s[::-1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/11.py
158
851
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> []

12. Write a Python Program to check if a string is a palindrome.

```
12.py > ...
1 s = input()
2 print("Palindrome" if s == s[::-1] else "Not Palindrome")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/12.py
158
851
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> []

13. Write a Python Program to count vowels and consonants in a string.

```
13.py > ...
1 s = input().lower()
2 v = 0
3 c = 0
4 for i in s:
5     if i.isalpha():
6         if i in "aeiou":
7             v += 1
8         else:
9             c += 1
10 print(v, c)
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/13.py
9 14

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

14. Write a Python Program to find the length of a string without using 'len()'.

```
14.py > ...
1 s = input()
2 count = 0
3 for i in s:
4     count += 1
5 print(count)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/14.py
25

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

15. Write a Python Program to remove all spaces from a string.

```
15.py > ...
1 s = input()
2 print(s.replace(" ", ""))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/15.py
Global Quest Technologies

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

16. Write a Python Program to count occurrences of a substring.

```
16.py > ...
1   s = input()
2   sub = input()
3   print(s.count(sub))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/16.py
● Global Quest Technologies
Tech
1
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

17. Write a Python Program to convert a string to uppercase.

```
17.py > ...
1   s = input()
2   print(s.upper())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/17.py
● global quest technologies
GLOBAL QUEST TECHNOLOGIES
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

18. Write a Python Program to replace vowels with ***

```
18.py > ...
1   s = input()
2   for i in "aeiouAEIOU":
3       s = s.replace(i, "***")
4   print(s)
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/18.py
● global quest technologies
g***b***t ***ch***l***g***s
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

19. Write a Python Program to check if two strings are anagrams.

```
19.py > ...
1 a = input()
2 b = input()
3 print("Anagram" if sorted(a) == sorted(b) else "Not Anagram")
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/19.py
● heart
earth
Anagram
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/19.py
global
globe
Not Anagram
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

20. Write a Python Program to find the first non-repeated character in a string.

```
20.py > ...
1 s = input()
2 for i in s:
3     if s.count(i) == 1:
4         print(i)
5         break
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/20.py
● Globalquest
G
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

21. Write a Python Program to find the largest element in a list.

```
21.py > ...
1 l = list(map(int, input().split()))
2 print(max(l))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/21.py
● 5 6 8 9 10 3 5
10
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

22. Write a Python Program to find the smallest element in a list.

```
22.py > ...
1 l = list(map(int, input().split()))
2 print(min(l))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/22.py

● 6 7 2 0 4 5 11 16
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

powershell + × └ └ └ └ └ └

23. Write a Python Program to calculate the sum of elements in a list.

```
23.py > ...
1 l = list(map(int, input().split()))
2 print(sum(l))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/23.py

● 15 6 5 7 8 9
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

powershell + × └ └ └ └ └ └

24. Write a Python Program to remove duplicates from a list.

```
24.py > ...
1 l = list(map(int, input().split()))
2 print(list(set(l)))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/24.py

● 1 2 3 4 5 6 2 4 8 2
[1, 2, 3, 4, 5, 6, 8]
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

powershell + × └ └ └ └ └ └

25. Write a Python Program to sort a list in ascending order.

```
25.py > ...
1  l = list(map(int, input().split()))
2  l.sort()
3  print(l)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/25.py
● 1 5 6 7 8 2 3
[1, 5, 6, 7, 8, 2, 3]
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

26. Write a Python Program to sort a list in descending order.

```
26.py > ...
1  l = list(map(int, input().split()))
2  l.sort(reverse=True)
3  print(l)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/26.py
● 1 2 5 6 4 7 5 9
[9, 7, 6, 5, 4, 2, 1]
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

27. Write a Python Program to find the second largest element in a list.

```
27.py > ...
1  l = list(set(map(int, input().split())))
2  l.sort()
3  print(l[-2])
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/27.py
1 2 5 6 9 4 7 8
8
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

28. Write a Python Program to merge two lists.

```
28.py > ...
1 a = list(map(int, input().split()))
2 b = list(map(int, input().split()))
3 print(a + b)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/28.py
1 2 3 6 3 4
9 6 5 7 2 4
[1, 2, 3, 6, 3, 4, 9, 6, 5, 7, 2, 4]
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

29. Write a Python Program to find common elements between two lists.

```
29.py > ...
1 a = set(map(int, input().split()))
2 b = set(map(int, input().split()))
3 print(list(a & b))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/29.py
1 2 3 4 5 6 7
4 5 9 3 6 7 2
[2, 3, 4, 5, 6, 7]
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

30. Write a Python Program to rotate a list by 'k' positions.

```
30.py > ...
1 l = list(map(int, input().split()))
2 k = int(input())
3 print(l[k:] + l[:k])
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/30.py
1 5 8 6 9 7 5
[7, 5, 1, 5, 8, 6, 9]
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

31. Write a Python Program to check if a number is an Armstrong number.

```
31.py> ...
1 n = input()
2 s = 0
3 for i in n:
4     s += int(i) ** len(n)
5 print("Armstrong" if s == int(n) else "Not Armstrong")
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/31.py
● 153
Armstrong
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/31.py
● 115
Not Armstrong
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

32. Write a Python Program to check if a number is a perfect number.

```
32.py > ...
1 n = int(input())
2 s = 0
3 for i in range(1, n):
4     if n % i == 0:
5         s += i
6 print("Perfect" if s == n else "Not Perfect")
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/32.py

6
Perfect

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/32.py

16
Not Perfect

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> █

33. Write a Python Program to check if a number is a palindrome.

```
33.py > ...
1 n = input()
2 print("Palindrome" if n == n[::-1] else "Not Palindrome")
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
powershell + × ☰ ... ×

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/33.py
● 131
Palindrome
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/33.py
● 568
○ Not Palindrome
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

34. Write a Python Program to find the GCD of two numbers.

```
34.py > ...
1 import math
2 a = int(input())
3 b = int(input())
4 print(math.gcd(a, b))
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/34.py
● 4
5
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

35. Write a Python Program to find the LCM of two numbers.

```
35.py > ...
1 import math
2 a = int(input())
3 b = int(input())
4 print(a * b // math.gcd(a, b))
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/35.py
● 9
8
72
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

36. Write a Python Program to convert decimal to binary.

```
36.py > ...
1 n = int(input())
2 print(bin(n)[2:])
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/36.py
● 68
1000100
○ PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

37. Write a Python Program to convert binary to decimal.

```
37.py > ...
1 b = input()
2 print(int(b, 2))

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/37.py
110011
51
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>
```

38. Write a Python Program to generate prime numbers up to 'n'

```
38.py > ...
1 n = int(input("Enter n: "))
2 for num in range(2, n+1):
3     for i in range(2, int(num**0.5)+1):
4         if num % i == 0:
5             break
6     else:
7         print(num)
8

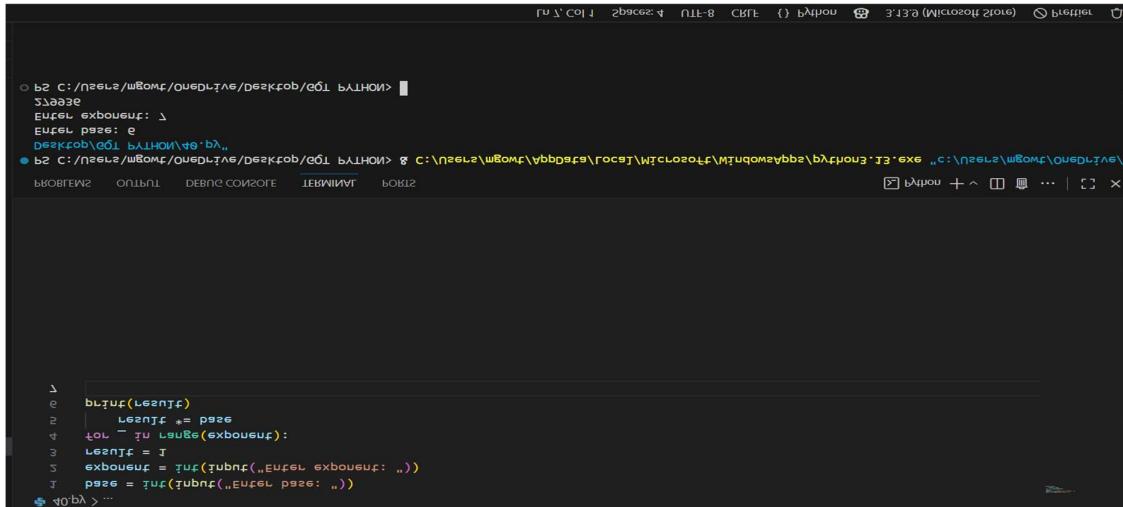
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/38.py"
Enter n: 6
2
3
5
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

39. Write a Python Program to find the sum of natural numbers up to 'n'

```
39.py > ...
1 n = int(input("Enter n: "))
2 total = n * (n + 1) // 2
3 print(total)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/39.py"
Enter n: 5
15
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

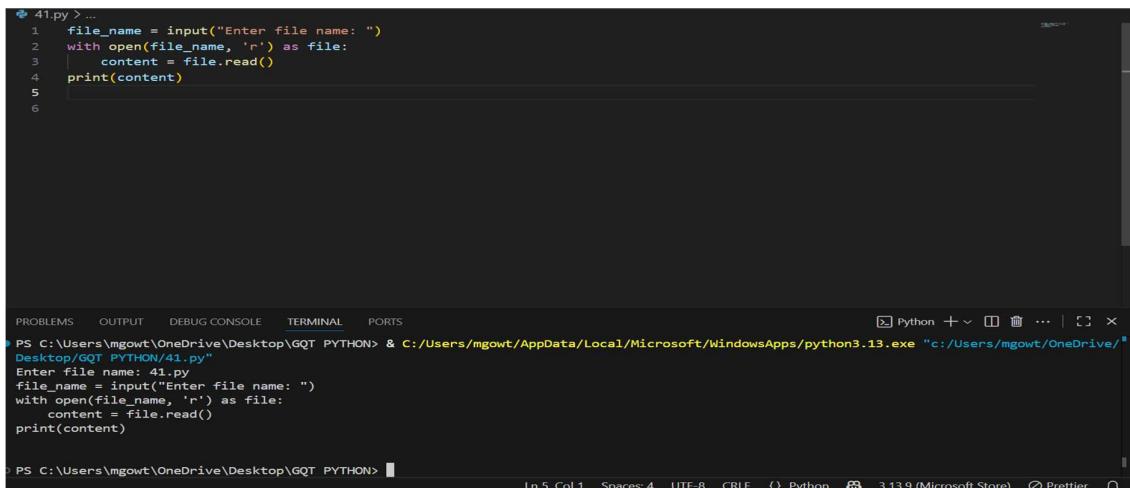
40. Write a Python Program to calculate the power of a number without using



```
PS C:\Users\mgovt\Desktop\GQT PYTHON> python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/40.py"
Enter base: 2
Enter exponent: 3
2 to the power of 3 is 8
```

The screenshot shows a terminal window with the command `python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/40.py"`. The output is `Enter base: 2` followed by `Enter exponent: 3` and then `2 to the power of 3 is 8`.

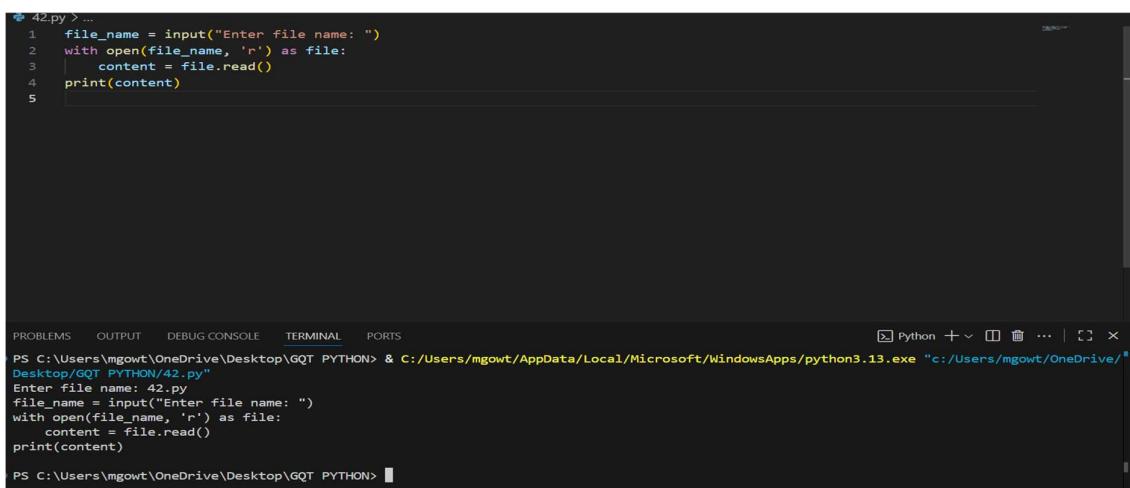
41. Write a Python Program to read a text file.



```
PS C:\Users\mgovt\Desktop\GQT PYTHON> python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/41.py"
Enter file name: 41.py
file_name = input("Enter file name: ")
with open(file_name, 'r') as file:
    content = file.read()
print(content)
```

The screenshot shows a terminal window with the command `python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/41.py"`. The output is `Enter file name: 41.py` followed by the content of the file `41.py` which is `file_name = input("Enter file name: ")`.

42. Write a Python Program to write to a text file.



```
PS C:\Users\mgovt\Desktop\GQT PYTHON> python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/42.py"
Enter file name: 42.py
file_name = input("Enter file name: ")
with open(file_name, 'r') as file:
    content = file.read()
print(content)
```

The screenshot shows a terminal window with the command `python3.13.exe "c:/Users/mgovt/Desktop/GQT PYTHON/42.py"`. The output is `Enter file name: 42.py` followed by the content of the file `42.py` which is `file_name = input("Enter file name: ")`.

43. Write a Python Program to count words in a file.

```
43.py > ...
1 file_name = input("Enter file name: ")
2 with open(file_name, 'r') as file:
3     content = file.read()
4 words = content.split()
5 print(len(words))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/43.py"
Enter file name: 43.py
18
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF ⚡ Python 3.13.9 (Microsoft Store) ⚡ Prettier ⚡

44. Write a Python Program to count lines in a file.

```
44.py > ...
1 file_name = input("Enter file name: ")
2 with open(file_name, 'r') as file:
3     lines = file.readlines()
4 print(len(lines))
5
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/44.py"
Enter file name: 44.py
5
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF ⚡ Python 3.13.9 (Microsoft Store) ⚡ Prettier ⚡

45. Write a Python Program to copy contents from one file to another.

```
45.py > ...
1 source_file = input("Enter source file name: ")
2 destination_file = input("Enter destination file name: ")
3 with open(source_file, 'r') as src:
4     data = src.read()
5 with open(destination_file, 'w') as dest:
6     dest.write(data)
7 print("File copied successfully")
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/45.py"
Enter source file name: python
Enter destination file name: python

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF ⚡ Python 3.13.9 (Microsoft Store) ⚡ Prettier ⚡

46. Write a Python Program to check if a file exists.

```
46.py
1 import os
2 print(os.path.isfile("file.txt"))
3 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/46.py"
False
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

47. Write a Python Program to append text to a file.

```
47.py > ...
1 f = open("file.txt", "a")
2 f.write("Appended text\n")
3 f.close()
4 |
5 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/47.py"
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

48. Write a Python Program to find the longest word in a file.

```
48.py > ...
1 f = open("file.txt", "r")
2 words = f.read().split()
3 f.close()
4 print(max(words, key=len))
5 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/48.py"
Appended
○ PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

49. Write a Python Program to remove blank lines from a file.

A screenshot of the Visual Studio Code interface. The code editor shows a Python script named 49.py with the following content:

```
49.py > ...
1 f = open("file.txt","r")
2 lines = f.readlines()
3 f.close()
4 f = open("file.txt","w")
5 for line in lines:
6     if line.strip():
7         f.write(line)
8 f.close()
9
10
```

The terminal below the code editor shows the command being run and its output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/49.py"
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

At the bottom of the interface, there are status indicators for Python 3.13.9 (Microsoft Store) and Prettier.

50. Write a Python Program to read a CSV file.

A screenshot of the Visual Studio Code interface. The code editor shows a Python script named 50.py with the following content:

```
50.py > ...
1 import csv
2
3 file_name = input("Enter CSV file name: ")
4 with open(file_name, 'r') as file:
5     reader = csv.reader(file)
6     for row in reader:
7         print(row)
8
```

The terminal below the code editor shows the command being run and its output. It also shows the raw input provided to the program:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Desktop/GQT PYTHON/50.py
Enter CSV file name: 50.py
['import csv']
[]
['file_name = input("Enter CSV file name: ")']
['with open(file_name, "r") as file:']
['    reader = csv.reader(file)']
['    for row in reader:']
['        print(row)']
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

At the bottom of the interface, there are status indicators for Python 3.13.9 (Microsoft Store) and Prettier.

51. Write a Python Program to print multiplication table of a number.

A screenshot of the Visual Studio Code interface. The code editor shows a Python script named 51.py with the following content:

```
51.py > ...
1 n = int(input())
2 for i in range(1,11):
3     print(n*i)
4
5
```

The terminal below the code editor shows the command being run and its output:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/51.py"
```

At the bottom of the interface, there are status indicators for Python 3.13.9 (Microsoft Store) and Prettier.

52. Write a Python Program to print all even numbers between 1 and 100.

```
52.py > ...
1  for i in range[2,10,2]:
2      print(i)
3

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/52.py"
2
4
6
8
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

53. Write a Python Program to print all odd numbers between 1 and 100.

```
53.py > ...
1  for i in range(1,11,2):
2      print(i)
3

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/53.py"
1
3
5
7
9
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

54. Write a Python Program to calculate the sum of first 'n' natural numbers using a loop.

```
54.py > ...
1  n = int(input())
2  s = 0
3  for i in range(1,n+1):
4      s += i
5  print(s)
6
7  |
```

```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/54.py"

Ln 7, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  3.13.9 (Microsoft Store)  Prettier
```

55. Write a Python Program to print a pyramid pattern of stars.

55.py > ...

```
1 n = int(input())
2 for i in range(1,n+1):
3     print(" "**(n-i)+"*"* (2*i-1))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/55.py"

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF {} Python 🐍 3.13.9 (Microsoft Store) ⚙️ Prettier

56. Write a Python Program to print an inverted pyramid of stars.

56.py > ...

```
1 n = int(input())
2 for i in range(n,0,-1):
3     print(" "**(n-i)+"*"* (2*i-1))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/56.py"

57. Write a Python Program to print Pascal's triangle.

57.py > ...

```
1 n = int(input("Enter number of rows: "))
2 for i in range(n):
3     num = 1
4     for j in range(i + 1):
5         print(num, end=" ")
6         num = num * (i - j) // (j + 1)
7     print()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/57.py"

Enter number of rows: 5

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

58. Write a Python Program to print Floyd's triangle.

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following code in a file named 58.py:

```
58.py > ...
1 n = int(input("Enter : "))
2 num = 1
3 for i in range(1,n+1):
4     for j in range(i):
5         print(num,end=" ")
6         num += 1
7     print()
```

When run, the program prompts the user for an input value. The user enters '6'. The terminal then outputs the following Floyd's triangle:

```
Enter : 6
1
2 3
4 5 6
7 8 9 10
11 12 13 14 15
16 17 18 19 20 21
```

59. Write a Python Program to print prime numbers between 1 and 100.

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following code in a file named 59.py:

```
59.py > ...
1 for n in range(2,10):
2     for i in range(2,n):
3         if n%i==0:
4             break
5     else:
6         print(n)
```

When run, the program prints prime numbers from 2 to 9, as no numbers greater than 10 are present in the range specified in the code.

```
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/59.py"
2
3
5
7
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

60. Write a Python Program to print numbers divisible by 3 and 5 up to 100.

The screenshot shows the VS Code interface with the terminal tab selected. The terminal window displays the following code in a file named 60.py:

```
60.py > ...
1 for i in range(1,101):
2     if i%3==0 and i%5==0:
3         print(i)
4
```

When run, the program prints all numbers from 1 to 100 that are divisible by both 3 and 5 (i.e., multiples of 15).

```
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/60.py"
15
30
45
60
75
90
90
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

61. Write a Python Program to define a function that returns the square of a number.

61.py > ...

```
1 def square(n):
2     return n*n
3 print(square(int(input("Hello"))))
4 |
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/61.py"
Hello

62. Write a Python Program to define a function that checks if a number is prime.

```
62.py > ...
1 def is_prime(n):
2     if n<2:
3         return False
4     for i in range(2,n):
5         if n%i==0:
6             return False
7     return True
8 print(is_prime(int(input("GQT"))))
9
```

63. Write a Python Program to define a function that calculates factorial using recursion.

The screenshot shows a code editor with a Python script named '63.py'. The code defines a recursive factorial function and prints its result for input 'GQT'. Below the editor is a terminal window showing the script's execution and output.

```
63.py > ...
1 def fact(n):
2     if n==0:
3         return 1
4     return n*fact(n-1)
5 print(fact(int(input("GQT"))))
6
```

```
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/63.py"
GQT
```

64. Write a Python Program to define a function that finds the maximum of three numbers.

```
64.py > ...
1 def maximum(a,b,c):
2     return max(a,b,c)
3 print(maximum(*map(int,input().split())))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/64.py"

65. Write a Python Program to define a function that returns the reverse of a string.

```
65.py > ...
1 def reverse(s):
2     return s[::-1]
3 print(reverse(input("GQT")))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/65.py"

66. Write a Python Program to define a function that counts vowels in a string.

```
66.py > ...
1 def reverse(s):
2     return s[::-1]
3 print(reverse(input("GQT python")))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/66.py"

67. Write a Python Program to define a function that checks if a string is palindrome.

```
67.py > ...
1 def palindrome(s):
2     return s==s[::-1]
3 print(palindrome(input("GQT Python")))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/67.py"
GQT Python

In 3, Col 39 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

68. Write a Python Program to define a function that returns the sum of digits of a number.

```
68.py > ...
1 def sum_digits(n):
2     return sum(int(i) for i in str(n))
3 print(sum_digits(int(input())))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/67.py"
GQT Python

In 3, Col 39 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

69. Write a Python Program to define a function that generates Fibonacci series up to 'n'.

```
69.py > ...
1 def fibonacci(n):
2     a,b = 0,1
3     for _ in range(n):
4         print(a,end=" ")
5         a,b = b,a+b
6     fibonacci(int(input("GQT Python")))
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/69.py"
GQT Python

In 6, Col 36 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

70. Write a Python Program to define a function that calculates power of a number using recursion.

```
70.py > ...
1 def power(base, exp):
2     if exp == 0:
3         return 1
4     return base * power(base, exp - 1)
5
6 b = int(input("Enter base: "))
7 e = int(input("Enter exponent: "))
8 print(power(b, e))
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/70.py"
Enter base: 4
Enter exponent: 6
4096
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

Ln 6, Col 31 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

71. Write a Python Program to calculate factorial using recursion.

```
71.py > ...
1 def factorial(n):
2     if n == 0 or n == 1:
3         return 1
4     return n * factorial(n - 1)
5
6 num = int(input("Enter a number: "))
7 print(factorial(num))
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/71.py"
Enter a number: 5
120
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

Ln 6, Col 31 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

72. Write a Python Program to generate Fibonacci series using recursion.

```
72.py > ...
1 def fib(n):
2     if n <= 1:
3         return n
4     return fib(n-1) + fib(n-2)
5
6 n = int(input("Enter number of terms: "))
7 for i in range(n):
8     print(fib(i), end=" ")
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/72.py"
Enter number of terms: 5
0 1 1 2 3
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>

Ln 6, Col 31 Spaces: 4 UTF-8 CRLF {} Python 3.13.9 (Microsoft Store) ⚡ Prettier

73. Write a Python Program to find the sum of natural numbers using recursion.

The screenshot shows a Python code editor with a dark theme. The code in the editor is:

```
1 def sum_n(n):
2     if n == 0:
3         return 0
4     return n + sum_n(n-1)
5
6 n = int(input("Enter n: "))
7 print(sum_n(n))
8
```

Below the code editor is a terminal window showing the execution of the script:

```
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/73.py"
Enter n: 5
15
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> []
```

74. Write a Python Program to reverse a string using recursion.

The screenshot shows a code editor window with a dark theme. At the top, there's a status bar with the file name "74.py > ...". The main area contains the following Python code:

```
1 def reverse(s):
2     if s == "":
3         return s
4     return reverse(s[1:]) + s[0]
5
6 text = input("Enter string: ")
7 print(reverse(text))
8
```

Below the code editor, there's a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is currently selected. The terminal window displays the following session:

```
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON> & C:/Users/mgowt/AppData/Local/Microsoft/WindowsApps/python3.13.exe "c:/Users/mgowt/OneDrive/Desktop/GQT PYTHON/74.py"
Enter string: gqt
tag
PS C:\Users\mgowt\OneDrive\Desktop\GQT PYTHON>
```

At the bottom of the screen, there are several status icons and labels: Ln 8, Col 1, Spaces: 4, UTF-8, CRLF, {}, Python, 3.13.9 (Microsoft Store), Prettier, and a file icon.

75. Write a Python Program to check if a string is palindrome using recursion.

The screenshot shows a dark-themed instance of Visual Studio Code. The top navigation bar includes 'File', 'Edit', 'Selection', 'View', 'Go', 'Run', and a file list. The title bar says 'GQT Python'. The left sidebar has icons for file operations like Open, Save, Find, and others. The main editor area contains the following Python code:

```
1 def is_palindrome(s):
2     if len(s) <= 1:
3         return True
4     if s[0] != s[-1]:
5         ret (function) def is_palindrome(s: Any) -> (Any | bool)
6     return is_palindrome(s[1:-1])
7 print(is_palindrome("radar"))
```

The code defines a recursive function `is_palindrome` that checks if a string is a palindrome by comparing its first and last characters and then calling itself on the substring between them. A test call is made at the bottom.

Below the editor, the 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS' tabs are visible. The 'TERMINAL' tab is active, showing command-line output:

```
PS C:\Users\...> & C:\...\\python.exe "c:/.../75.py"
True
PS C:\Users\...>
```

The bottom status bar shows 'Ln 7, Col 30', 'Spaces: 4', 'UTF-8', 'CRLF', and file status indicators for Python files 75.py through 84.py. A 'Finish Setup' button is also present.

76. Write a Python Program to find GCD of two numbers using recursion.

The screenshot shows a code editor window titled "GQT Python". The file "76.py" contains the following code:

```
def gcd(a, b):
    if b == 0:
        return a
    return gcd(b, a % b)
print(gcd(48, 18))
```

The terminal tab shows the output of running the script:

```
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/76.py"
6
```

77. Write a Python Program to find LCM of two numbers using recursion.

The screenshot shows a code editor window titled "GQT Python". The file "77.py" contains the following code:

```
def lcm(a, b):
    def gcd(a, b):
        if b == 0:
            return a
        return gcd(b, a % b)
    return abs(a*b)//gcd(a,b)
print(lcm(4, 6))
```

78. Write a Python Program to calculate sum of digits using recursion.

The screenshot shows a code editor window titled "GQT Python". The file "78.py" contains the following code:

```
def sum_digits(n):
    if n == 0:
        return 0
    return n % 10 + sum_digits(n // 10)
print(sum_digits(123))
```

The terminal tab shows the output of running the script:

```
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/78.py"
6
```

79. Write a Python Program to find the length of a string using recursion.

The screenshot shows a code editor interface with multiple tabs at the top, each containing a file name: 75.py, 76.py, 77.py, 78.py, 79.py (highlighted in blue), 80.py, 81.py, 82.py, 83.py, and 84.py. The main workspace displays the content of 79.py:

```
79.py > string.length
1 def string.length(s):
2     if s == "":
3         return 0
4     return 1 + string.length(s[1:])
5 print(string.length("hello"))
6
```

The code defines a recursive function `string.length` that calculates the length of a string by adding 1 to the length of the substring from index 1 to the end. A call to this function is made with the argument "hello".

Below the code editor, there is a navigation bar with icons for back, forward, search, and other file operations. The bottom of the screen features a terminal window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell + × └─ ... | C ×

PS C:\Users\Del1\Desktop\GQT Python> & C:\Users\Del1\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Del1/Desktop/GQT Python/79.py"
5
PS C:\Users\Del1\Desktop\GQT Python> [ ]
```

The terminal output shows the execution of the script and the resulting value of 5.

80. Write a Python Program to print numbers from 'n' to 1 using recursion.

The screenshot shows a Python code editor with multiple tabs open, all showing variations of a recursive function named `print_n_to_1`. The code is as follows:

```
def print_n_to_1(n):
    if n == 0:
        return
    print(n)
    print_n_to_1(n-1)

print_n_to_1(5)
```

The tab bar includes files 75.py through 84.py. The code editor interface has various icons for file operations like save, copy, and delete.

Below the code editor is a terminal window showing the command-line interface. The prompt is PS C:\Users\... . It shows the command being run: `python.exe "c:/Users/Dell/Desktop/GQT Python/88.py"`. The output of the script is displayed, showing the numbers 5, 4, 3, 2, and 1 printed sequentially.

81. Write a Python Program to find the maximum element in a tuple.

The screenshot shows a dark-themed Python development environment. The top navigation bar includes File, Edit, Selection, View, Go, Run, and a terminal input field. Below the navigation bar is a toolbar with icons for file operations like Open, Save, and Print. The main workspace displays several Python files (75.py, 76.py, 77.py, 78.py, 79.py, 80.py, 81.py, 82.py, 83.py, 84.py) and their content. A code editor window is open, showing the following code:

```
81.py > [e] t
1 t = (3, 5, 1, 9)
2 print(max(t))
3
```

The bottom of the screen features a tab bar with PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, displaying a command-line interface with the following text:

```
PS C:\Users\...> & C:\Users\...Python.exe "c:/.../81.py"
9
PS C:\Users\...>
```

The status bar at the bottom provides information about the current file (Ln 1, Col 1), encoding (UTF-8), and Python version (Python 3.13 (64-bit)). There are also icons for file save, setup, and help.

82. Write a Python Program to find the minimum element in a tuple.

The screenshot shows the GQT Python IDE interface. The top menu bar includes File, Edm, Selection, View, Go, Run, and a series of back/forward arrows. The title bar says "GQT Python". The left sidebar contains icons for file operations like Open, Save, Find, and others. The main workspace shows a list of Python files (75.py, 76.py, 77.py, 78.py, 79.py, 80.py, 81.py, 82.py, 83.py, 84.py) and a code editor window displaying the following code:

```
82.py> ...
1 t = (3, 5, 1, 9)
2 print(min(t))
3
```

The bottom navigation bar includes PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The DEBUG CONSOLE tab is active, showing the command "powershell" and its output:

```
PS C:\Users\Del1\Desktop\GQT Python> & C:\Users\Del1\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Del1/Desktop/GQT Python/82.py"
1
PS C:\Users\Del1\Desktop\GQT Python>
```

The status bar at the bottom indicates "Ln 3, Col 1" and "Python 3.13 (64-bit)".

83. Write a Python Program to convert a list into a tuple.

84. Write a Python Program to convert a tuple into a list.

```
84.py > ...
1  t = (1,2,3,4)
2  print(list(t))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS powershell + ⌂ ⌂ ... | ⌂ x

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/84.py"
[1, 2, 3, 4]
PS C:\Users\DELL\Desktop\GQT Python> █
```

85. Write a Python Program to find the union of two sets.

```
85.py > ...
1 a = {1,2,3}
2 b = {3,4,5}
3 print(a | b)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/85.py"
{1, 2, 3, 4, 5}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

86. Write a Python Program to find the intersection of two sets.

```
111.py > ...
1 year = 2024
2 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
3     print("Leap year")
4 else:
5     print("Not leap year")
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/111.py"
Leap year
PS C:\Users\DELL\Desktop\GQT Python>

Ln 6, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

87. Write a Python Program to find the difference of two sets.

```
87.py > ...
1 a = {1,2,3}
2 b = {3,4,5}
3 print(a - b)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/87.py"
{1, 2}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 4, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

88. Write a Python Program to check if a set is subset of another set.

```
88.py > ...
1 a = {1,2,3}
2 b = {3,4,5}
3 print({1,2}.issubset(a))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/88.py"
True
PS C:\Users\DELL\Desktop\GQT Python>

Ln 2, Col 12 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

89. Write a Python Program to remove duplicates from a list using set.

```
89.py > ...
1 l = [1,2,2,3]
2 print(list(set(l)))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/89.py"
[1, 2, 3]
PS C:\Users\DELL\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

90. Write a Python Program to count unique elements in a list using set.

```
90.py > ...
1 l = [1,2,2,3]
2 print(len(set(l)))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/90.py"
3
PS C:\Users\DELL\Desktop\GQT Python>

Ln 1, Col 14 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

91. Write a Python Program to create a dictionary of student names and marks.

A screenshot of the Visual Studio Code interface. The terminal window shows the following Python code:

```
91.py > [redacted] d
1   d = {"Alice":85, "Bob":90}
2   print(d)
3
```

The terminal output shows the program running and printing the dictionary:

```
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/91.py"
{'Alice': 85, 'Bob': 90}
PS C:\Users\DELL\Desktop\GQT Python>
```

The status bar at the bottom indicates the code is in Line 1, Column 1, with 4 spaces, using UTF-8 encoding, and is set to Python. It also shows Python 3.13 (64-bit) is installed.

92. Write a Python Program to access values from a dictionary.

A screenshot of the Visual Studio Code interface. The terminal window shows the following Python code:

```
92.py > ...
1   d = {"Alice":85, "Bob":90}
2   print(d)
3   print(d["Alice"])
4
```

The terminal output shows the program running and printing the value for 'Alice'.

```
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/92.py"
85
PS C:\Users\DELL\Desktop\GQT Python>
```

The status bar at the bottom indicates the code is in Line 4, Column 1, with 4 spaces, using UTF-8 encoding, and is set to Python. It also shows Python 3.13 (64-bit) is installed.

93. Write a Python Program to update values in a dictionary.

A screenshot of the Visual Studio Code interface. The terminal window shows the following Python code:

```
93.py > ...
1   d = {"Alice":85, "Bob":90}
2   print(d)
3   d["Alice"] = 95
4   print(d)
5
```

The terminal output shows the program running and printing the updated dictionary.

```
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/93.py"
{'Alice': 85, 'Bob': 90}
{'Alice': 95, 'Bob': 90}
PS C:\Users\DELL\Desktop\GQT Python>
```

The status bar at the bottom indicates the code is in Line 5, Column 1, with 4 spaces, using UTF-8 encoding, and is set to Python. It also shows Python 3.13 (64-bit) is installed.

94. Write a Python Program to delete a key from a dictionary.

```
94.py > ...
1  d = {"Alice":85, "Bob":90}
2  print(d)
3  del d["Bob"]
4  print(d)
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/94.py"
{'Alice': 85, 'Bob': 90}
{'Alice': 85}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)
```

95. Write a Python Program to merge two dictionaries.

```
95.py > ...
1  d1 = {"x":1}
2  d2 = {"y":2}
3  d1.update(d2)
4  print(d1)
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/95.py"
{'x': 1, 'y': 2}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)
```

96. Write a Python Program to count frequency of characters in a string using dictionary.

```
96.py > [s]
1  s = "hello"
2  freq = {}
3  for c in s:
4      freq[c] = freq.get(c,0)+1
5  print(freq)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/96.py"
{'h': 1, 'e': 1, 'l': 2, 'o': 1}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 5, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)
```

97. Write a Python Program to count frequency of words in a sentence using dictionary.

```
97.py > ...
1 s = "hello world hello"
2 freq = {}
3 for w in s.split():
4     freq[w] = freq.get(w,0)+1
5 print(freq)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/97.py"
{'hello': 2, 'world': 1}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

98. Write a Python Program to find the key with maximum value in a dictionary.

```
98.py > ...
1 d = {"a":5,"b":10}
2 print(max(d,key=d.get))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/98.py"
b
PS C:\Users\DELL\Desktop\GQT Python>

Ln 3, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

99. Write a Python Program to sort a dictionary by values.

```
99.py > ...
1 d = {"a":5,"b":10}
2 print(dict(sorted(d.items(), key=lambda x:x[1])))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/99.py"
{'a': 5, 'b': 10}
PS C:\Users\DELL\Desktop\GQT Python>

Ln 1, Col 19 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

100. Write a Python Program to check if a key exists in a dictionary.

```
100.py > ...
1  d = {"a":5,"b":10}
2  print("a" in d)
3
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/100.py"
True
PS C:\Users\DELL\Desktop\GQT Python>

Ln 4, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  Finish Setup  Python 3.13 (64-bit)  🔍
```

101. Write a Python Program to iterate over a list using 'for' loop.

```
101.py > ...
1  l = [1,2,3]
2  for i in l:
3      print(i)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/101.py"
1
2
3
PS C:\Users\DELL\Desktop\GQT Python>

Ln 4, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  Finish Setup  Python 3.13 (64-bit)  🔍
```

102. Write a Python Program to iterate over a list using 'while' loop.

```
102.py > ...
1  l = [1,2,3]
2  i = 0
3  while i < len(l):
4      print(l[i])
5      i += 1
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\DELL\Desktop\GQT Python> & C:\Users\DELL\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/DELL/Desktop/GQT Python/102.py"
1
2
3
PS C:\Users\DELL\Desktop\GQT Python>

Ln 6, Col 1  Spaces: 4  UTF-8  CRLF  {} Python  Finish Setup  Python 3.13 (64-bit)  🔍
```

103. Write a Python Program to find the sum of elements in a tuple.

```
103.py > [REDACTED]
1 t = (3, 5, 1, 9)
2 total = 0
3 for i in t:
4     total += i
5 print(total)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Del1\Desktop\GQT Python> & C:\Users\Del1\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Del1/Desktop/GQT Python/103.py"
18
PS C:\Users\Del1\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)

104. Write a Python Program to check if an element exists in a tuple.

```
104.py > [REDACTED]
1 t = (3, 5, 1, 9)
2 print(5 in t)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Del1\Desktop\GQT Python> & C:\Users\Del1\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Del1/Desktop/GQT Python/104.py"
True
PS C:\Users\Del1\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)

105. Write a Python Program to convert a string into a list of characters.

```
105.py > [REDACTED]
1 s = "Hello"
2 chars = []
3 for c in s:
4     chars.append(c)
5 print(chars)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\Del1\Desktop\GQT Python> & C:\Users\Del1\AppData\Local\Programs\Python\Python313\python.exe "c:/Users/Del1/Desktop/GQT Python/105.py"
['H', 'e', 'l', 'l', 'o']
PS C:\Users\Del1\Desktop\GQT Python>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF { } Python Finish Setup Python 3.13 (64-bit)

106. Write a Python Program to join a list of strings into a single string.

```
106.py > [e]
1 l = ["hello", "world"]
2 result = ""
3 for i in range(len(l)):
4     result += l[i]
5     if i != len(l)-1:
6         result += " "
7 print(result)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../106.py"
hello world
PS C:\Users\...>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

107. Write a Python Program to find the largest word in a sentence.

```
107.py > ...
1 s = "I love programming"
2 words = s.split()
3 largest = words[0]
4 for w in words:
5     if len(w) > len(largest):
6         largest = w
7 print(largest)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../107.py"
programming
PS C:\Users\...>

Ln 8, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

108. Write a Python Program to find the smallest word in a sentence.

```
108.py > [e] s
1 s = "I love programming"
2 words = s.split()
3 smallest = words[0]
4 for w in words:
5     if len(w) < len(smallest):
6         smallest = w
7 print(smallest)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../108.py"
I
PS C:\Users\...>

Ln 1, Col 1 Spaces: 4 UTF-8 CRLF {} Python Finish Setup Python 3.13 (64-bit)

109. Write a Python Program to count the number of words in a sentence.

The screenshot shows a code editor interface with a dark theme. On the left is a code editor pane containing the following Python code:109.py > [e] s
1 s = "I love programming"
2 count = 0
3 for w in s.split():
4 count += 1
5 print(count)
6On the right is a terminal pane showing the output of running the script:PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell + v 🗑 ... | ⚡ ×
PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../109.py"
3
PS C:\Users\...>At the bottom, status bar information includes: Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, { } Python, Finish Setup, Python 3.13 (64-bit).

110. Write a Python Program to check if a number is positive, negative, or zero.

The screenshot shows a code editor interface with a dark theme. On the left is a code editor pane containing the following Python code:110.py > [e] n
1 n = 5
2 if n > 0:
3 print("Positive")
4 elif n < 0:
5 print("Negative")
6 else:
7 print("Zero")
8On the right is a terminal pane showing the output of running the script:PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell + v 🗑 ... | ⚡ ×
PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../110.py"
Positive
PS C:\Users\...>At the bottom, status bar information includes: Ln 1, Col 1, Spaces: 4, UTF-8, CRLF, { } Python, Finish Setup, Python 3.13 (64-bit).

111. Write a Python Program to check if a year is a leap year.

The screenshot shows a code editor interface with a dark theme. On the left is a code editor pane containing the following Python code:111.py > ...
1 year = 2024
2 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
3 print("Leap year")
4 else:
5 print("Not leap year")
6On the right is a terminal pane showing the output of running the script:PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
powershell + v 🗑 ... | ⚡ ×
PS C:\Users\...> & C:\Users\...Local\Programs\Python\Python313\python.exe "c:/Users/.../111.py"
Leap year
PS C:\Users\...>At the bottom, status bar information includes: Ln 6, Col 1, Spaces: 4, UTF-8, CRLF, { } Python, Finish Setup, Python 3.13 (64-bit).

112. Write a Python Program to calculate simple interest.

The screenshot shows a Python code editor interface with several files listed in the sidebar. The active file is `simpleinterest.py`. The code calculates simple interest using the formula $SI = P \times T \times R / 100$.

```
1 p=float(input())
2 t=float(input())
3 r=float(input())
4 si=(p*t*r)/100
5 print(si)
6
```

The terminal window below shows the output of running the program:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQ/simpleinterest.py
● 30000
7
2
7840.0
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

113. Write a Python Program to calculate compound interest.

The screenshot shows a Python code editor interface with several files listed in the sidebar. The active file is `Compound.py`. The code calculates compound interest using the formula $CI = P \times (1+r/100)^t$.

```
1 p,r,t=map(float,input().split())
2 ci=p*(1+r/100)**t
3 print(ci)
4
```

The terminal window below shows the output of running the program:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQ/compound.py
● 3000 5
3307.5
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

114. Write a Python Program to convert Celsius to Fahrenheit.

The screenshot shows a Python code editor interface with several files listed in the sidebar. The active file is `celsiustofahrenheit.py`. The code converts Celsius to Fahrenheit using the formula $F = (C \times 9/5) + 32$.

```
1 c=float(input())
2 f=(c*9/5)+32
3 print(f)
4
```

The terminal window below shows the output of running the program:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQ/celsiustofahrenheit.py
● 25
77.0
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

115. Write a Python Program to convert Fahrenheit to Celsius.

A screenshot of the Visual Studio Code (VS Code) interface. The terminal at the bottom shows the output of running a Python script named `fahrenheittocelsius.py`. The code itself is as follows:

```
1 fahrenheitcelsius.py > ...
2     f=float(input())
3     c=(f-32)*5/9
4     print(c)
```

The terminal output shows the conversion of 32 degrees Fahrenheit to approximately 0 degrees Celsius.

116. Write a Python Program to calculate area of a circle.

A screenshot of the Visual Studio Code (VS Code) interface. The terminal at the bottom shows the output of running a Python script named `circle.py`. The code is:

```
1 circle.py > ...
2     r=float(input())
3     area=3.14159*r**2
4     print(area)
```

The terminal output shows the calculation of the area of a circle with radius 7, resulting in approximately 153.93791 square units.

117. Write a Python Program to calculate area of a rectangle.

A screenshot of the Visual Studio Code (VS Code) interface. The terminal at the bottom shows the output of running a Python script named `rectangle.py`. The code is:

```
1 rectangle.py > ...
2     l=float(input())
3     b=float(input())
4     print(l*b)
```

The terminal output shows the calculation of the area of a rectangle with length 46 and breadth 51, resulting in 2346.0 square units.

118. Write a Python Program to calculate area of a triangle.

The screenshot shows a Python code editor interface. In the top navigation bar, the tabs are: File, Edit, Selection, View, Go, Run, Terminal, Help, and SESSIONS. The terminal tab is selected. The terminal window displays the following command and output:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/triangle.py
n/qQT/triangle.py
1
b=float(input())
2
h=float(input())
3
print(0.5*b*h)
4
```

The output shows the input values 1 and 2, followed by the calculated result 21.0.

119. Write a Python Program to calculate perimeter of a square.

The screenshot shows a Python code editor interface. In the top navigation bar, the tabs are: File, Edit, Selection, View, Go, Run, Terminal, Help, and SESSIONS. The terminal tab is selected. The terminal window displays the following command and output:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/squareP.py
squareP.py
1
s=float(input())
2
print(4*s)
3
```

The output shows the input value 6, followed by the calculated result 24.0.

120. Write a Python Program to calculate perimeter of a rectangle

The screenshot shows a Python code editor interface. In the top navigation bar, the tabs are: File, Edit, Selection, View, Go, Run, Terminal, Help, and SESSIONS. The terminal tab is selected. The terminal window displays the following command and output:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/rectangleP.py
rectangleP.py
1
rectangleP.py
2
l=float(input())
3
b=float(input())
4
print(2*(l+b))
5
```

The output shows the input values 45 and 21, followed by the calculated result 132.0.

121. Write a Python Program to calculate perimeter of a circle

The screenshot shows a dark-themed version of the Visual Studio Code interface. In the center, there's a code editor window titled "cubeV.py" containing the following Python code:

```
GGT > cubeV.py >...
1 s=float(input())
2 print(s*3)
3
```

Below the code editor is a terminal window showing the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/cubeV.py
216.0
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

122. Write a Python Program to calculate volume of a cube

The screenshot shows a dark-themed version of the Visual Studio Code interface. In the center, there's a code editor window titled "circleP.py" containing the following Python code:

```
GGT > circleP.py >...
1 r=float(input())
2 print(2*3.14159*r)
3
```

Below the code editor is a terminal window showing the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/circleP.py
7
43.9826
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

123. Write a Python Program to calculate volume of a cylinder

The screenshot shows a dark-themed version of the Visual Studio Code interface. In the center, there's a code editor window titled "cylinderV.py" containing the following Python code:

```
GGT > cylinderV.py >...
1 r=float(input())
2 h=float(input())
3 print(3.14159*r*r*h)
4
```

Below the code editor is a terminal window showing the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/cylinderV.py
7
351.8888
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

124. Write a Python Program to calculate volume of a sphere.

The screenshot shows the Visual Studio Code interface with a dark theme. On the left is the sidebar with icons for file operations like Open, Save, Find, and Run. The main area is a code editor with the file 'sphereV.py' open. The code contains three lines of Python:1 r=float(input())
2 print((4/3)*3.14159*r**3)The status bar at the bottom indicates the file is saved. Below the code editor is the terminal window, which shows the command PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/sphereV.py and the output 113.09723999999999. The bottom right corner of the terminal has a tooltip: 'sphereV.py' and 'Describe what to build next'.

125. Write a Python Program to calculate surface area of a cube

The screenshot shows the Visual Studio Code interface with a dark theme. The sidebar and code editor are identical to the previous screenshot, showing 'cubeS.py'. The code contains three lines of Python:1 s=float(input())
2 print(6*s*s)
3The terminal window shows the command PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/cubeS.py and the output 150.0. The bottom right corner of the terminal has a tooltip: 'cubeS.py' and 'Describe what to build next'.

126. Write a Python Program to calculate surface area of a cylinder.

The screenshot shows the Visual Studio Code interface with a dark theme. The sidebar and code editor are identical to the previous screenshots, showing 'cylinderS.py'. The code contains four lines of Python:1 r=float(input())
2 h=float(input())
3 print(2*3.14159*r*(r+h))
4The terminal window shows the command PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/cylinderS.py and the output 408.4067. The bottom right corner of the terminal has a tooltip: 'cylinderS.py' and 'Describe what to build next'.

127. Write a Python Program to calculate surface area of a sphere.

The screenshot shows the Visual Studio Code interface with a dark theme. The code editor displays a file named `spheres.py` containing the following code:

```
GGT > spheres.py > ...
1   r=float(input())
2   print(4*3.14159*r*r)
3
```

The terminal below shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/Desktop/Gangothri/Python/n/Q07/spheres.py
804.24704
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

128. Write a Python Program to check if a character is uppercase.

The screenshot shows the Visual Studio Code interface with a dark theme. The code editor displays a file named `uppercase.py` containing the following code:

```
GGT > uppercase.py > ...
1   ch=input()
2   print(ch.isupper())
3
```

The terminal below shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/Q07/uppercase.py
False
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

129. Write a Python Program to check if a character is lowercase

The screenshot shows the Visual Studio Code interface with a dark theme. The code editor displays a file named `lowercase.py` containing the following code:

```
GGT > lowercase.py > ...
1   ch=input()
2   print(ch.islower())
3
```

The terminal below shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/Q07/lowercase.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

130. Write a Python Program to check if a character is a digit

The screenshot shows the VS Code interface with a dark theme. A file named `digit.py` is open in the editor. The code contains three lines of Python: `ch=input()`, `print(ch.isdigit())`, and `3`. The terminal below shows the output of running the program: `PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/digit.py`, followed by the output `True`.

```
digit.py
1 ch=input()
2 print(ch.isdigit())
3

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/digit.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

131. Write a Python Program to check if a character is an alphabet

The screenshot shows the VS Code interface with a dark theme. A file named `alphabet.py` is open in the editor. The code contains three lines of Python: `ch=input()`, `print(ch.isalpha())`, and `3`. The terminal below shows the output of running the program: `PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/alphabet.py`, followed by the output `True`.

```
alphabet.py
1 ch=input()
2 print(ch.isalpha())
3

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/alphabet.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

132. Write a Python Program to check if a character is a special symbol

The screenshot shows the VS Code interface with a dark theme. A file named `symbol.py` is open in the editor. The code contains three lines of Python: `ch=input()`, `print(not ch.isalnum())`, and `3`. The terminal below shows the output of running the program: `PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/symbol.py`, followed by the output `False`.

```
symbol.py
1 ch=input()
2 print(not ch.isalnum())
3

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/q01/symbol.py
False
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

133. Write a Python Program to count uppercase letters in a string

The screenshot shows a code editor window in VS Code with a dark theme. On the left is a sidebar with icons for file operations like Open, Save, Find, and Run. The main area contains a Python script named `countupper.py`. The code uses a for loop to iterate through each character in the input string `s`, checking if it is an uppercase letter using the `i.isupper()` method. If true, it increments a counter `c`. Finally, it prints the value of `c`. Below the code is a terminal window showing the command `python countupper.py` being run, which outputs the number 3. A sidebar on the right lists other files in the workspace.

```
countupper.py
1 s=input()
2 c=0
3 for i in s:
4     if i.isupper():
5         c+=1
6 print(c)
7
```

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/countupper.py
Ganga
3
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

134. Write a Python Program to count lowercase letters in a string

This screenshot is similar to the previous one, showing the same code editor and terminal setup. The script `countupper.py` has been modified to count lowercase letters instead. It uses the `i.islower()` method to check if a character is lowercase. The terminal output shows the command `python countupper.py` running and outputting the number 3, indicating it correctly counts the lowercase letters in the word "Ganga".

```
countupper.py
1 s=input()
2 c=0
3 for i in s:
4     if i.islower():
5         c+=1
6 print(c)
7
```

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/countupper.py
ganga
3
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

135. Write a Python Program to count digits in a string

This screenshot shows the code editor with a new Python script `countdigits.py`. The code uses a for loop to iterate through each character in the input string `s`, checking if it is a digit using the `i.isdigit()` method. If true, it increments a counter `c`. Finally, it prints the value of `c`. The terminal output shows the command `python countdigits.py` running and outputting the number 3, indicating it correctly counts the digits in the word "Ganga".

```
countupper.py  countdigits.py
countdigits.py
1 s=input()
2 c=0
3 for i in s:
4     if i.isdigit():
5         c+=1
6 print(c)
7
```

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/countdigits.py
Ganga
3
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

136. Write a Python Program to count special characters in a string

A screenshot of the Visual Studio Code interface. The left sidebar shows files: 'countupper.py' and 'countcharacters.py'. The 'countcharacters.py' tab is active, displaying the following code:

```
1 s=input()
2 c=0
3 for i in s:
4     if not i.isalnum() and i!=" ":
5         c+=1
6 print(c)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/countcharacters.py
● GQT@123
1
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

137. Write a Python Program to remove punctuation from a string

A screenshot of the Visual Studio Code interface. The left sidebar shows files: 'countupper.py' and 'remove.py'. The 'remove.py' tab is active, displaying the following code:

```
1 s=input()
2 res=""
3 for i in s:
4     if i.isalnum() or i==" ":
5         res+=i
6 print(res)
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/remove.py
● Hello world
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

138. Write a Python Program to replace spaces with hyphens in a string

A screenshot of the Visual Studio Code interface. The left sidebar shows files: 'hyphens.py'. The 'hyphens.py' tab is active, displaying the following code:

```
1 s=input()
2 print(s.replace(" ","-"))
```

The terminal at the bottom shows the output of running the script:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/hyphens.py
● Hello world
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

139. Write a Python Program to split a string into words.

```
splitstring.py
1 s=input()
2 print(s.split())
3

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/python/splitstring.py
78 20 30
['78', '20', '30']
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

140. Write a Python Program to join words into a sentence.

```
joinwords.py
1 words=input().split()
2 print(",".join(words))
3

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/joinwords.py
global quest
global quest
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

141. Write a Python Program to check if a number is divisible by another number.

```
divisible.py
1 n=int(input())
2 if n%5 == 0 and n%11 == 0:
3     print("Divisible by 5 and 11")
4 else:
5     print("not divisible by 5 and 11")

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/divisible.py
54
not divisible by 5 and 11
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

142. Write a Python Program to find the square root of a number.

The screenshot shows the Visual Studio Code interface with a dark theme. A file named `square root.py` is open in the editor. The code contains three lines of Python code:square root.py
```python
1 n=float(input())
2 print(n\*\*0.5)
3
```
The terminal below shows the output of running the program:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/gango/OneDrive/Desktop/Gangothri/Python/square root.py"
3.0
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```
A status bar at the bottom right indicates the build configuration is set to `Python`.

143. Write a Python Program to find cube root of a number.

The screenshot shows the Visual Studio Code interface with a dark theme. A file named `cuberoot.py` is open in the editor. The code contains three lines of Python code:cuberoot.py
```python
1 n=float(input())
2 print(n***(1/3))
3
```
The terminal below shows the output of running the program:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/cuberoot.py
5
1.7099750466766968
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```
A status bar at the bottom right indicates the build configuration is set to `Python`.

144. Write a Python Program to calculate power using 'pow()' function.

The screenshot shows the Visual Studio Code interface with a dark theme. A file named `power.py` is open in the editor. The code contains four lines of Python code:power.py
```python
1 a=float(input())
2 b=float(input())
3 print(pow(a,b))
4
```
The terminal below shows the output of running the program:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/power.py
7
6
117649.0
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```
A status bar at the bottom right indicates the build configuration is set to `Python`.

145. Write a Python Program to calculate absolute value of a number.

The screenshot shows the Visual Studio Code interface with a Python file named `absolute.py` open. The code contains:absolute.py
1 n=float(input())
2 print(abs(n))
3The terminal below shows the output of running the program:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/users/gango/appdata/local/programs/python/python312/python.exe c:/users/gango/OneDrive/Desktop/Gangothri/Python/absolute.py
789.0

146. Write a Python Program to generate random numbers.

The screenshot shows the Visual Studio Code interface with a Python file named `146.py` open. The code contains:146.py
1 import random
2
3 print(random.random())
4The terminal below shows the output of running the program:PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/146.py
0.51761997151883

147. Write a Python Program to generate random integers between two numbers.

The screenshot shows the Visual Studio Code interface with a Python file named `147.py` open. The code contains:147.py
1 import random
2
3 a = int(input("Enter start number: "))
4 b = int(input("Enter end number: "))
5
6 print(random.randint(a, b))
7The terminal below shows the output of running the program twice, with different user inputs:PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/147.py
Enter start number: 1
Enter end number: 100
51
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/147.py
Enter start number: 1
Enter end number: 50
41

148. Write a Python Program to shuffle elements of a list.

```
148.py > ...
1 import random
2
3 lst = [1, 2, 3, 4, 5]
4 random.shuffle(lst)
5
6 print(lst)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & C:/Python313/python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/148.py
[5, 3, 1, 2, 4]
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

149. Write a Python Program to pick a random element from a list.

```
1 import random
2
3 lst = [10, 20, 30, 40, 50]
4 print(random.choice(lst))
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT\_PythonPrograms/149.py
30
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

150. Write a Python Program to simulate rolling a dice.

```
1 import random
2
3 print(random.randint(1, 6))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT\_PythonPrograms/150.py
5
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

151. Write a Python Program to simulate tossing a coin.

```
1 import random
2
3 print(random.choice(["Heads", "Tails"]))
4
```

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT_PythonPrograms/151.py
● Heads
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

The terminal shows the program output: "Heads".

152. Write a Python Program to generate a random password.

```
1 import random
2
3 import string
4
5 chars = string.ascii_letters + string.digits + string.punctuation
6 password = ''.join(random.choice(chars) for _ in range(8))
7 print(password)
```

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT_PythonPrograms/152.py
x/JYUC*A
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> []
```

The terminal shows the generated password: "x/JYUC\*A".

153. Write a Python Program to generate a random OTP.

```
1 import random
2
3 otp = random.randint(100000, 999999)
4 print(otp)
5
```

The screenshot shows a terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT_PythonPrograms/153.py
673638
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

The terminal shows the generated OTP: "673638".

154. Write a Python Program to generate a random prime number.

```
1 import random
2
3 def is_prime(n):
4 if n < 2:
5 return False
6 for i in range(2, int(n**0.5) + 1):
7 if n % i == 0:
8 return False
9 return True
10
11 while True:
12 num = random.randint(2, 100)
13 if is_prime(num):
14 print(num)
15 break
16
```

The screenshot shows a code editor window with a dark theme. The terminal tab is selected, displaying the following command-line session:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT_PythonPrograms/154.py
29
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

155. Write a Python Program to generate a random even number.

```
1 import random
2
3 print(random.randrange(0, 100, 2))
4
```

The screenshot shows a code editor window with a dark theme. The terminal tab is selected, displaying the following command-line session:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT_PythonPrograms/155.py
46
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

156. Write a Python Program to generate a random odd number.

```
1 import random
2
3 print(random.randrange(1, 100, 2))
4
```

The screenshot shows a code editor window with a dark theme. The terminal tab is selected, displaying the following command-line session:

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT_PythonPrograms/156.py
59
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

157. Write a Python Program to generate a random string of given length.

```
1 import random
2 import string
3
4 length = 6
5 result = ''.join(random.choice(string.ascii_letters) for _ in range(length))
6 print(result)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QT/QT\_PythonPrograms/157.py
Baofwg
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

158. Write a Python Program to generate random alphanumeric string.

```
1 import random
2 import string
3
4 result = ''.join(random.choice(string.ascii_letters + string.digits) for _ in range(8))
5 print(result)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QT/QT\_PythonPrograms/158.py
oBubewS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

159. Write a Python Program to generate random floating-point numbers.

```
1 import random
2
3 print(random.uniform(1, 10))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QT/QT\_PythonPrograms/159.py
7.404514921566846
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

160. Write a Python Program to generate random numbers within a range.

```
1 import random
2
3 print(random.randint(50, 100))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT\_PythonPrograms/160.py
78
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

161. Write a Python Program to check if a list is empty.

```
1 lst = []
2 print(len(lst) == 0)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT\_PythonPrograms/161.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

162. Write a Python Program to check if a string is empty.

```
1 s = ""
2 print(len(s) == 0)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/QQT/QQT\_PythonPrograms/162.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

163. Write a Python Program to check if a tuple is empty.

```
1 t = ()
2 print(len(t) == 0)
3
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/gqt/gqt_PythonPrograms/163.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

164. Write a Python Program to check if a dictionary is empty.

```
1 d = {}
2 print(len(d) == 0)
3
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/gqt/gqt_PythonPrograms/164.py
True
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

165. Write a Python Program to remove all elements from a list.

```
1 lst = [1, 2, 3]
2 lst.clear()
3 print(lst)
4
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/gqt/gqt_PythonPrograms/165.py
[]
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

166. Write a Python Program to remove all elements from a dictionary.

```
1 d = {"a": 1, "b": 2}
2 d.clear()
3 print(d)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/q1/Q1\_PythonPrograms/166.py
()
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

167. Write a Python Program to remove all elements from a set.

```
1 s = {1, 2, 3}
2 s.clear()
3 print(s)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/q1/Q1\_PythonPrograms/167.py
set()
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

168. Write a Python Program to copy a list.

```
1 lst = [1, 2, 3]
2 new_lst = lst.copy()
3 print(new_lst)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/q1/Q1\_PythonPrograms/168.py
[1, 2, 3]
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

169. Write a Python Program to copy a dictionary.

```
1 d = {"a": 1, "b": 2}
2 new_d = d.copy()
3 print(new_d)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT_PythonPrograms/169.py
{'a': 1, 'b': 2}
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

170. Write a Python Program to copy a set.

```
1 s = {1, 2, 3}
2 new_s = s.copy()
3 print(new_s)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT_PythonPrograms/170.py
{1, 2, 3}
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

171. Write a Python Program to reverse a list.

```
1 lst = [1, 2, 3, 4]
2 lst.reverse()
3 print(lst)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT_PythonPrograms/171.py
[4, 3, 2, 1]
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

172. Write a Python Program to reverse a tuple.

```
1 t = (1, 2, 3, 4)
2 print(t[::-1])
3
```

The screenshot shows a terminal window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QQT/QQT_PythonPrograms/172.py
(4, 3, 2, 1)
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

173. Write a Python Program to reverse a dictionary.

```
1 d = {"a": 1, "b": 2, "c": 3}
2 reversed_d = {v: k for k, v in d.items()}
3 print(reversed_d)
4
```

The screenshot shows a terminal window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QQT/QQT_PythonPrograms/173.py
{1: 'a', 2: 'b', 3: 'c'}
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

174. Write a Python Program to reverse a set.

```
1 s = {1, 2, 3, 4}
2 reversed_s = set(list(s)[::-1])
3 print(reversed_s)
4
```

The screenshot shows a terminal window with the following text:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QQT/QQT_PythonPrograms/174.py
{1, 2, 3, 4}
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>
```

175. Write a Python Program to reverse words in a sentence.

```
1 sentence = "Python is easy"
2 print(" ".join(sentence.split()[::-1]))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT\_PythonPrograms/175.py
easy is Python

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

176. Write a Python Program to reverse characters in each word of a sentence.

```
1 sentence = "Python is easy"
2 result = " ".join(word[::-1] for word in sentence.split())
3 print(result)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT\_PythonPrograms/176.py
nohtyp si yesae

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

177. Write a Python Program to reverse order of lines in a file.

input.txt

```
1 Line one
2 Line two
3 Line three
4
```

177.py > ...
1 with open("input.txt", "r") as file:
2 lines = file.readlines()
3
4 with open("input.txt", "w") as file:
5 for line in reversed(lines):
6 file.write(line)
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS QUERY RESULTS

PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT> & c:\Python313\python.exe c:/Users/chand/OneDrive/Pictures/Desktop/GQT/177.py
PS C:\Users\chand\OneDrive\Pictures\Desktop\GQT>

input.txt

```
1 Line three
2 Line two
3 Line one
4
```

178. Write a Python Program to reverse digits of a number.

179. Write a Python Program to reverse elements of a nested list.

```
1 nested = [[1, 2], [3, 4], [5, 6]]
2 reversed_nested = [sub[::-1] for sub in nested][::-1]
3 print(reversed_nested)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/QT/QT\_PythonPrograms/179.py

- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> [[6, 5], [4, 3], [2, 1]]
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

180. Write a Python Program to reverse elements of a nested dictionary.

```
1 nested = {"a": {"x": 1, "y": 2}, "b": {"z": 3}}
2 reversed_nested = {k: {v: k2 for k2, v in val.items()} for k, val in nested.items()}
3 print(reversed_nested)
4
```

181. Write a Python Program to check if a number is prime using function.

```
1 def is_prime(n):
2 if n < 2:
3 return False
4 for i in range(2, int(n**0.5) + 1):
5 if n % i == 0:
6 return False
7 return True
8
9 print(is_prime(7))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT\_PythonPrograms/181.py

True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

182. Write a Python Program to check if a string is palindrome using function.

```
1 def is_palindrome(s):
2 return s == s[::-1]
3
4 print(is_palindrome("madam"))
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT\_PythonPrograms/182.py

True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

183. Write a Python Program to check if a number is Armstrong using function.

```
1 def is_armstrong(num):
2 order = len(str(num))
3 total = sum(int(digit) ** order for digit in str(num))
4 return total == num
5
6 print(is_armstrong(153))
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/n/GQT/GQT\_PythonPrograms/183.py

True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

184. Write a Python Program to check if a number is perfect using function.

```
1 def is_perfect(num):
2 total = sum(1 for i in range(1, num) if num % i == 0)
3 return total == num
4
5 print(is_perfect(28))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/184.py

True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python>

185. Write a Python Program to check if a number is palindrome using function.

```
1 def is_palindrome(num):
2 return str(num) == str(num)[::-1]
3
4 n = int(input("Enter number: "))
5 print("Palindrome" if is_palindrome(n) else "Not Palindrome")
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/185.py

● Enter number: 32  
Not Palindrome

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

186. Write a Python Program to check if a string is anagram using function.

```
1 def is_anagram(s1, s2):
2 return sorted(s1) == sorted(s2)
3
4 a = input("Enter first string: ")
5 b = input("Enter second string: ")
6 print("Anagram" if is_anagram(a, b) else "Not Anagram")
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/186.py

● Enter first string: 34 45  
Enter second string: 36 78  
Not Anagram

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

187. Write a Python Program to check if a string is pangram using function.

```
187.py > ...
1 import string
2
3 def is_pangram(s):
4 return set(string.ascii_lowercase).issubset(set(s.lower()))
5
6 text = input("Enter string: ")
7 print("Pangram" if is_pangram(text) else "Not Pangram")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/gangothri/Python/QQT\_PythonPrograms/187.py

Enter string: global  
Not Pangram

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms>

188. Write a Python Program to check if a string contains only digits.

```
188.py > ...
1 def only_digits(s):
2 return s.isdigit()
3
4 print(only_digits(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/gangothri/Python/QQT\_PythonPrograms/188.py

56  
True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms>

189. Write a Python Program to check if a string contains only alphabets.

```
189.py > ...
1 def only_alpha(s):
2 return s.isalpha()
3
4 print([only_alpha(input())])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/gangothri/Python/QQT\_PythonPrograms/189.py

global  
True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\QQT\_PythonPrograms>

190. Write a Python Program to check if a string contains only alphanumeric characters.

The screenshot shows a Python code editor interface with a dark theme. The code in the editor is:190.py > ...
1 def only\_alnum(s):
2 return s.isalnum()
3
4 print(only\_alnum(input()))The terminal tab at the bottom shows the output of running the script:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/190.py
● 34567
True
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

191. Write a Python Program to check if a string contains only uppercase letters.

The screenshot shows a Python code editor interface with a dark theme. The code in the editor is:191.py > ...
1 def only\_upper(s):
2 return s.isupper()
3
4 print(only\_upper(input()))The terminal tab at the bottom shows the output of running the script:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/191.py
● Global
False
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

192. Write a Python Program to check if a string contains only lowercase letters.

The screenshot shows a Python code editor interface with a dark theme. The code in the editor is:192.py > ...
1 def only\_lower(s):
2 return s.islower()
3
4 print(only\_lower(input()))The terminal tab at the bottom shows the output of running the script:PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & c:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/192.py
● global
True
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

193. Write a Python Program to check if a string contains only whitespace.

```
193.py > ...
1 def only_space(s):
2 return s.isspace()
3
4 print(only_space(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/193.py

- global quest  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

194. Write a Python Program to check if a string contains only special characters.

```
194.py > ...
1 def only_special(s):
2 return all(not ch.isalnum() for ch in s)
3
4 print(only_special(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/194.py

- gqt@123  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

195. Write a Python Program to check if a string contains both letters and digits.

```
195.py > ...
1 def letter_digit(s):
2 return any(c.isalpha() for c in s) and any(c.isdigit() for c in s)
3
4 print(letter_digit(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/195.py

- gqt@123  
True
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

196. Write a Python Program to check if a string contains both uppercase and lowercase letters.

```
196.py > ...
1 def upper_lower(s):
2 return any(c.isupper() for c in s) and any(c.islower() for c in s)
3
4 print(upper_lower(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/196.py

global  
False

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

197. Write a Python Program to check if a string contains both vowels and consonants

```
197.py > ...
1 def vowel_consonant(s):
2 vowels = "aeiouAEIOU"
3 v = any(c in vowels for c in s if c.isalpha())
4 c = any(c.isalpha() and c not in vowels for c in s)
5 return v and c
6
7 print(vowel_consonant(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/197.py

ganga  
True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

198. Write a Python Program to check if a string contains repeated characters

```
198.py > ...
1 def repeated_chars(s):
2 return len(set(s)) != len(s)
3
4 print(repeated_chars(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/198.py

global  
True

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

199. Write a Python Program to check if a string contains unique characters

```
199.py > ...
1 def unique_chars(s):
2 return len(set(s)) == len(s)
3
4 print(unique_chars(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/199.py

- global  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

200. Write a Python Program to check if a string contains all vowels

```
200.py > ...
1 def all_vowels(s):
2 return set("aeiou").issubset(set(s.lower()))
3
4 print(all_vowels(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/200.py

- global  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

201. Write a Python Program to check if a number is prime using recursion.

```
201.py > ...
1 def is_prime(n, i=2):
2 if n <= 2:
3 return n == 2
4 if n % i == 0:
5 return False
6 if i * i > n:
7 return True
8 return is_prime(n, i+1)
9
10 print(is_prime(int(input())))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/201.py

- 5  
True
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

202. Write a Python Program to check if a string is palindrome using recursion

```
202.py > ...
1 def palindrome(s):
2 if len(s) <= 1:
3 return True
4 if s[0] != s[-1]:
5 return False
6 return palindrome(s[1:-1])
7
8 print(palindrome(input()))
```

Debug Console (Ctrl+Shift+Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/202.py

● 454  
True  
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

203. Write a Python Program to check if a number is Armstrong using recursion

```
203.py > ...
1 def armstrong(n, power, temp=None):
2 if temp is None:
3 temp = n
4 if temp == 0:
5 return 0
6 return (temp % 10) ** power + armstrong(n, power, temp // 10)
7
8 num = int(input())
9 p = len(str(num))
10 print(armstrong(num, p) == num)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/203.py

● 567  
False  
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

204. Write a Python Program to check if a number is perfect using recursion.

```
204.py > ...
1 def perfect(n, i=1, s=0):
2 if i == n:
3 return s
4 if n % i == 0:
5 s += i
6 return perfect(n, i+1, s)
7
8 num = int(input())
9 print(perfect(num) == num)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/204.py

● 25  
False  
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

205. Write a Python Program to check if a number is palindrome using recursion

```
205.py > ...
1 def rev(n, r=0):
2 if n == 0:
3 return r
4 return rev(n//10, r*10 + n%10)
5
6 num = int(input())
7 print(num == rev(num))

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT_PythonPrograms/205.py
● 2345
False
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms>
```

206. Write a Python Program to check if a string is anagram using recursion

```
206.py > ...
1 def anagram(s1, s2):
2 if len(s1) == 0:
3 return True
4 if s1[0] not in s2:
5 return False
6 return anagram(s1[1:], s2.replace(s1[0], "", 1))
7
8 print(anagram(input(), input()))

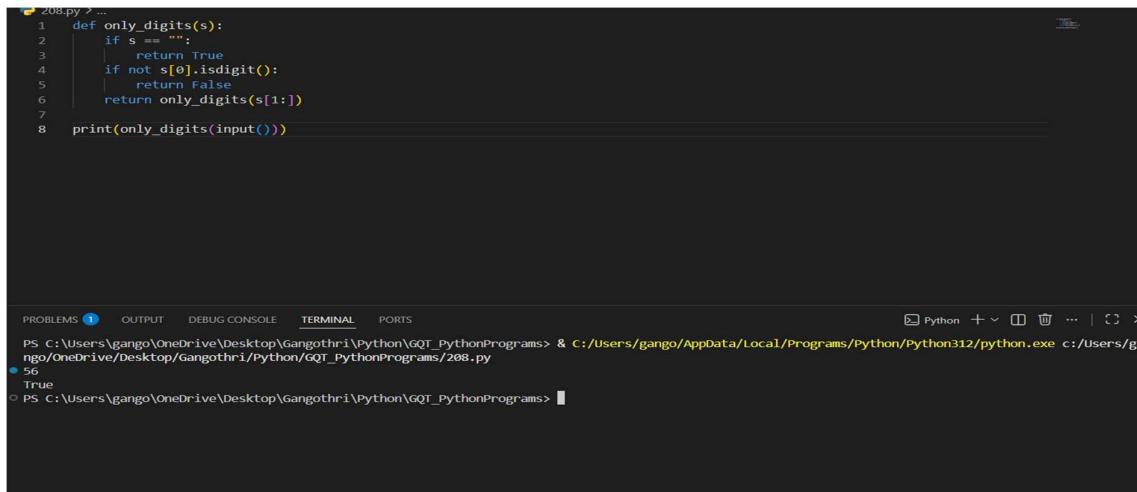
PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT_PythonPrograms/206.py
456
●
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms>
```

207. Write a Python Program to check if a string is pangram using recursion.

```
207.py > ...
1 import string
2
3 def pangram(s, letters=set(string.ascii_lowercase)):
4 if not letters:
5 return True
6 if not s:
7 return False
8 return pangram(s[1:], letters - {s[0].lower()})
9
10 print(pangram(input()))

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT_PythonPrograms/207.py
● global
False
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT_PythonPrograms>
```

208. Write a Python Program to check if a string contains only digits using recursion.



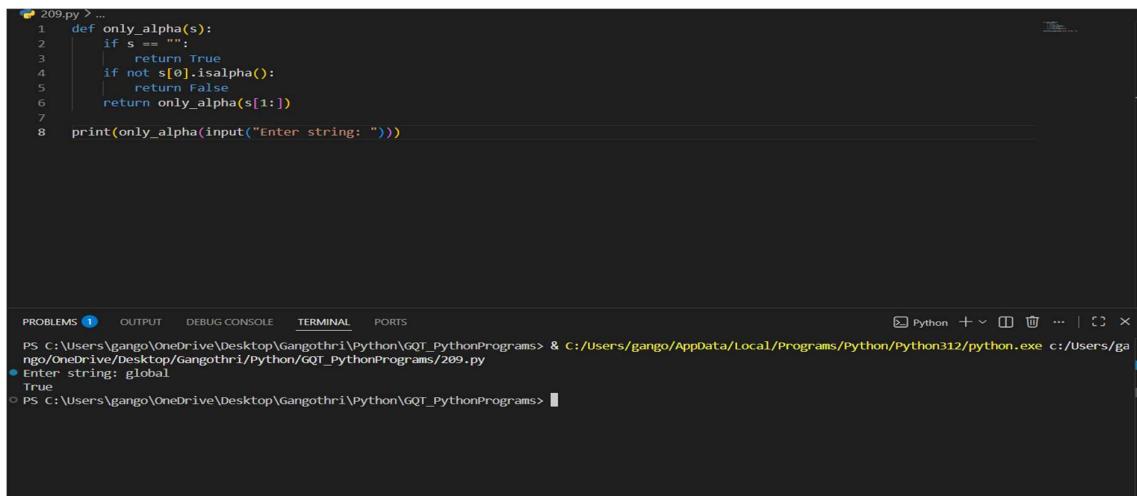
```
208.py > ...
1 def only_digits(s):
2 if s == "":
3 return True
4 if not s[0].isdigit():
5 return False
6 return only_digits(s[1:])
7
8 print(only_digits(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/208.py

● 56  
○ True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

209. Write a Python Program to check if a string contains only alphabets using recursion.



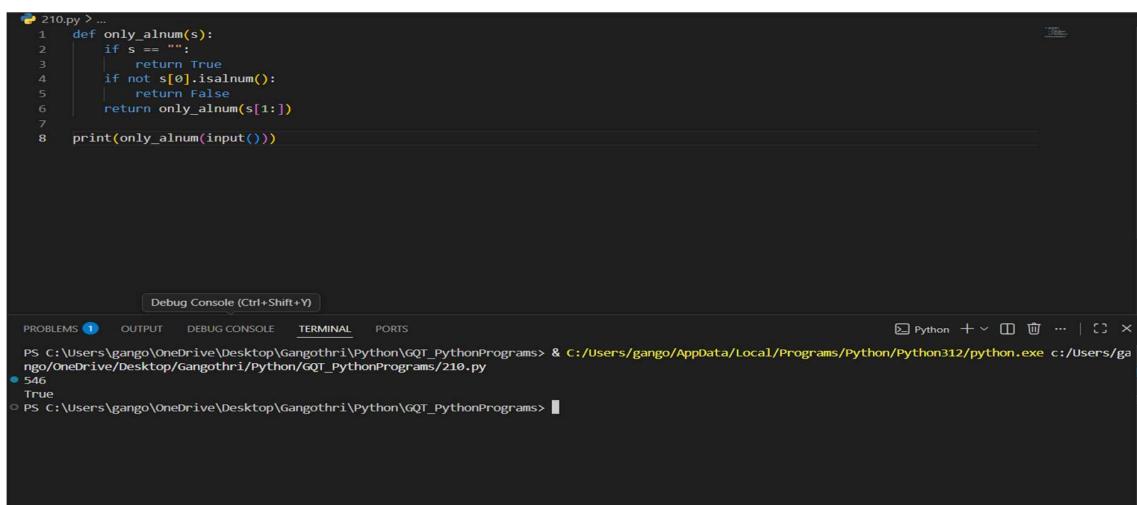
```
209.py > ...
1 def only_alpha(s):
2 if s == "":
3 return True
4 if not s[0].isalpha():
5 return False
6 return only_alpha(s[1:])
7
8 print(only_alpha(input("Enter string: ")))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/209.py

● Enter string: global  
○ True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

210. Write a Python Program to check if a string contains only alphanumeric characters using recursion.



```
210.py > ...
1 def only_alnum(s):
2 if s == "":
3 return True
4 if not s[0].isalnum():
5 return False
6 return only_alnum(s[1:])
7
8 print(only_alnum(input()))
```

Debug Console (Ctrl+Shift+Y)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/210.py

● S@#4  
○ True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

211. Write a Python Program to check if a string contains only uppercase letters using recursion.

```
211.py >...
1 def only_upper(s):
2 if s == "":
3 return True
4 if not s[0].isupper():
5 return False
6 return only_upper(s[1:])
7
8 print(only_upper(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT\_PythonPrograms/211.py

- global  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms>

212. Write a Python Program to check if a string contains only lowercase letters using recursion.

```
212.py >...
1 def only_lower(s):
2 if s == "":
3 return True
4 if not s[0].islower():
5 return False
6 return only_lower(s[1:])
7
8 print(only_lower(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT\_PythonPrograms/212.py

- global  
True
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms>

213. Write a Python Program to check if a string contains only whitespace using recursion

```
213.py >...
1 def only_space(s):
2 if s == "":
3 return True
4 if not s[0].isspace():
5 return False
6 return only_space(s[1:])
7
8 print(only_space(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GOT\_PythonPrograms/213.py

- glo bal  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GOT\_PythonPrograms>

214. Write a Python Program to check if a string contains only special characters using recursion.

```
214.py > ...
1 def only_special(s):
2 if s == "":
3 return True
4 if s[0].isalnum():
5 return False
6 return only_special(s[1:])
7
8 print(only_special(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/214.py

- @ True
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

215. Write a Python Program to check if a string contains both letters and digits using recursion

```
215.py > ...
1 def letter_digit(s, l=False, d=False):
2 if s == "":
3 return l and d
4 return letter_digit(s[1:], l or s[0].isalpha(),
5 d or s[0].isdigit())
6
7 print(letter_digit(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/215.py

- 45 True
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

216. Write a Python Program to check if a string contains both uppercase and lowercase letters using recursion.

```
216.py > ...
1 def upper_lower(s, u=False, l=False):
2 if s == "":
3 return u and l
4 return upper_lower(s[1:], u or s[0].isupper(),
5 l or s[0].islower())
6
7 print(upper_lower(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/216.py

- gTrue False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

217. Write a Python Program to check if a string contains both vowels and consonants using recursion

```
217.py > ...
1 def vowel_consonant(s, v=False, c=False):
2 vowels = "aeiouAEIOU"
3 if s == "":
4 return v and c
5 ch = s[0]
6 if ch.isalpha():
7 if ch in vowels:
8 v = True
9 else:
10 c = True
11 return vowel_consonant(s[1:], v, c)
12
13 print(vowel_consonant(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/217.py

global  
True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

218. Write a Python Program to check if a string contains repeated characters using recursion

```
218.py > ...
1 def repeated(s, seen=set()):
2 if s == "":
3 return False
4 if s[0] in seen:
5 return True
6 seen.add(s[0])
7 return repeated(s[1:], seen)
8
9 print(repeated(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/218.py

● ggtgat  
True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

219. Write a Python Program to check if a string contains unique characters using recursion.

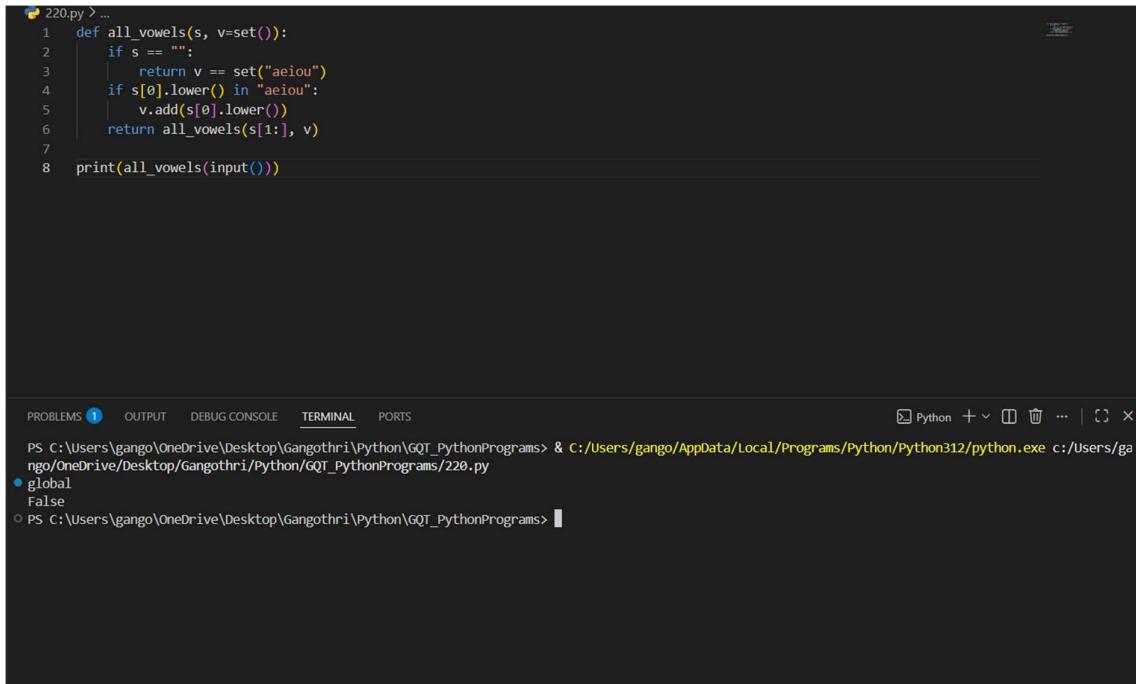
```
219.py > ...
1 def unique(s, seen=set()):
2 if s == "":
3 return True
4 if s[0] in seen:
5 return False
6 seen.add(s[0])
7 return unique(s[1:], seen)
8
9 print(unique(input()))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/219.py

● god  
True  
○ PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>

220. Write a Python Program to check if a string contains all vowels using recursion.



```
220.py > ...
1 def all_vowels(s, v=set()):
2 if s == "":
3 return v == set("aeiou")
4 if s[0].lower() in "aeiou":
5 v.add(s[0].lower())
6 return all_vowels(s[1:], v)
7
8 print(all_vowels(input()))
```

PROBLEMS 1 OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms> & C:/Users/gango/AppData/Local/Programs/Python/Python312/python.exe c:/Users/gango/OneDrive/Desktop/Gangothri/Python/GQT\_PythonPrograms/220.py

- global  
False
- PS C:\Users\gango\OneDrive\Desktop\Gangothri\Python\GQT\_PythonPrograms>



GLOBAL QUEST  
TECHNOLOGIES

fuel your  
**passion** for  
IT with  
our **guidance.**



# Global Quest Technologies



#324, 2nd Floor, 3 A Cross, Near  
Seshadripuram First Grade College,  
Above City Union Bank,  
Yelahanka New Town,  
Bengaluru-560064

+91 9448 403 469 | 080-49720009  
[info@gqtech.in](mailto:info@gqtech.in) | [www.gqtech.in](http://www.gqtech.in)