



An ISO 9001 & ISO 21001  
Certified Organization



The Quest  
**for your Dream Job**  
Ends Here!!

[www.gqtech.in](http://www.gqtech.in)

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

A screenshot of a code editor window titled "program1.py". The code contains two lines: "# program to print 'hello world'" and "print("Hello World")". Below the code editor, there is a terminal window showing the output of running the program: "Hello World".

```
program1.py X
program1.py
1 # program to print 'hello world'
2 print("Hello World")

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program1.py
Hello World
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

A screenshot of a code editor window titled "program2.py". The code contains three lines: "#program to take user input and display it", "a=input('Enter your name: ')", and "print(a)". Below the code editor, there is a terminal window showing the output of running the program and prompting for user input: "Enter your name: Geethika".

```
program2.py X
program2.py > ...
1 #program to take user input and display it
2 a=input('Enter your name: ')
3 print(a)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program2.py
Enter your name: Geethika
Geethika
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

## 3. Write a Python Program to swap two numbers.

A screenshot of a code editor window titled "program3.py". The code contains seven lines: "#program to swap two numbers", "a=520", "b=173", "print('Before Swapping: \n a = {}, b = {}'.format(a,b))", "a,b=b,a", "print('After Swapping: \n a = {}, b = {}'.format(a,b))", and "#". Below the code editor, there is a terminal window showing the output of running the program, which swaps the values of 'a' and 'b': "Before Swapping: a = 520, b = 173" and "After Swapping: a = 173, b = 520".

```
program3.py X
program3.py > ...
1 #program to swap two numbers
2 a=520
3 b=173
4 print('Before Swapping: \n a = {}, b = {}'.format(a,b))
5 a,b=b,a
6 print('After Swapping: \n a = {}, b = {}'.format(a,b))
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program3.py
Before Swapping:
a = 520, b = 173
After Swapping:
a = 173, b = 520
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program4.py > [e] n
1 n=int(input("Enter a number: "))
2 if n%2==0:
3     print("{} is an even number".format(n))
4 else:
5     print("{} is an odd number".format(n))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program4.py
Enter a number: 520
520 is an even number
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program5.py > [e] a
1 a=int(input('Enter the value for a : '))
2 b=int(input('Enter the value for b : '))
3 c=int(input('Enter the value for c : '))
4 if a>b and a>c:
5     print("{} is the largest number".format(a))
6 elif b>c:
7     print("{} is the largest number".format(b))
8 else:
9     print("{} is the largest number".format(c))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program5.py
Enter the value for a : 520
Enter the value for b : 173
Enter the value for c : 175
520 is the largest number
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program6.py > ...
1 n = int(input("Enter a number: "))
2 fact = 1
3
4 for i in range(1, n+1):
5     fact = fact * i
6
7 print("Factorial is:", fact)
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program6.py
Enter a number: 5
Factorial is: 120
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program7.py > ...
1 #Program to generate the fibonacci series
2 n = int(input("Enter the number of terms: "))
3 a, b = 0, 1
4 print("Fibonacci Series:", end=" ")
5
6 for _ in range(n):
7     print(a, end=" ")
8     a, b = b, a + b
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program7.py
Enter the number of terms: 8
Fibonacci Series: 0 1 1 2 3 5 8 13
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

## 8. Write a Python Program to reverse a number.

```
program8.py > ...
1 num = int(input("Enter a number: "))
2 rev = 0
3 temp = num
4
5 while temp > 0:
6     rev = rev * 10 + temp % 10
7     temp //= 10
8
9 print("Reversed Number:", rev)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program8.py
Enter a number: 173
Reversed Number: 371
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program9.py > ...
1 num = int(input("Enter a number: "))
2
3 if num > 1:
4     for i in range(2, int(num**0.5) + 1):
5         if num % i == 0:
6             print(num, "is not prime")
7             break
8         else:
9             print(num, "is prime")
10    else:
11        print(num, "is not prime")
12

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program9.py
Enter a number: 173
173 is prime
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program10.py > num
1 num = int(input("Enter a number: "))
2 sum_digits = 0
3 temp = num
4
5 while temp > 0:
6     sum_digits += temp % 10
7     temp //= 10
8
9 print("Sum of digits:", sum_digits)
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program10.py  
Enter a number: 1374  
Sum of digits: 15  
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

## 11. Write a Python Program to reverse a string.

```
program11.py > ...
1 a=input('Enter any string: ')
2 print(a[::-1])
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program11.py  
Enter any string: Rampandu  
udnapmaR  
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program12.py > a
1 a=input('Enter any string: ')
2 if a==a[::-1]:
3     print('{} is a palindrome'.format(a))
4 else:
5     print('{} is not a palindrome'.format(a))
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program12.py  
Enter any string: madam  
madam is a palindrome  
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program13.py > [text]
1 text = input("Enter a string: ").lower()
2 vowels = "aeiou"
3 vowel_count = consonant_count = 0
4
5 for char in text:
6     if char.isalpha():
7         if char in vowels:
8             vowel_count += 1
9         else:
10            consonant_count += 1
11
12 print("Vowels:", vowel_count)
13 print("Consonants:", consonant_count)
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program13.py
Enter a string: GeethikaBandaru
Vowels: 7
Consonants: 8
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program14.py > ...
1 string = input("Enter a string: ")
2 length = 0
3
4 for _ in string:
5     length += 1
6
7 print("Length of string:", length)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program14.py
Enter a string: PadmavathiBandaru
Length of string: 17
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program15.py > ...
1 string = input("Enter a string: ")
2 no_spaces = string.replace(" ", "")
3 print("String without spaces:", no_spaces)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program15.py
Enter a string: Geethika Geethu Geeth
String without spaces: GeethikaGeethuGeeth
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```

program16.py > [e] string
1   string = input("Enter a string: ")
2   substring = input("Enter the substring to count: ")
3   count = string.count(substring)
4   print(f'{substring}' ' occurs {count} times')
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program16.py
Enter a string: Geethika geeth geeth geeth
Enter the substring to count: geeth
'geeth' occurs 3 times
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

```

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

```

program17.py > [e] s
1   s=input('Enter a string: ')
2   s1=s.upper()
3   print('Before converting to uppercase: {} '.format(s))
4   print('After converting to uppercase: {}'.format(s1))

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program17.py
Enter a string: geethikabandaru
Before converting to uppercase: geethikabandaru
After converting to uppercase: GEETHIKABANDARU
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

```

## 18. Write a Python Program to replace vowels with '\*'.

```

program18.py > [e] string
1   string = input("Enter a string: ")
2   vowels = "aeiouAEIOU"
3   result = ''.join(['*' if char in vowels else char for char in string])
4   print("After replacing vowels:", result)
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program18.py
Enter a string: Geethikabandaru
After replacing vowels: G**th*k*B*nd*r*
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

```

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

```

program19.py > [e] str1
1   str1 = input("Enter first string: ").replace(" ", "").lower()
2   str2 = input("Enter second string: ").replace(" ", "").lower()
3
4   if sorted(str1) == sorted(str2):
5       print("The strings are anagrams")
6   else:
7       print("The strings are not anagrams")
8

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program19.py
Enter a string: GeethikaBandaru
After replacing vowels: G**th*k*B*nd*r*
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

```

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

```
program20.py > s
1 s = input("Enter string: ")
2
3 for ch in s:
4     if s.count(ch) == 1:
5         print("First non-repeated character:", ch)
6         break
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program20.py
Enter string: padmavathireethika
First non-repeated character: p
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program21.py > lst
1 lst = list(map(int, input("Enter list: ").split()))
2 largest = lst[0]
3
4 for i in lst:
5     if i > largest:
6         largest = i
7
8 print("Largest element:", largest)
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program21.py
Enter list: 1 2 6 4 9 10 5 6 29
Largest element: 29
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program22.py > lst
1 lst = list(map(int, input("Enter list: ").split()))
2 smallest = lst[0]
3
4 for i in lst:
5     if i < smallest:
6         smallest = i
7
8 print("Smallest element:", smallest)
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program22.py
Enter list: 3 6 8 20 5 1 3 7 0
Smallest element: 0
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program23.py > [e] lst
1 lst = list(map(int, input("Enter list: ").split()))
2 total = 0
3
4 for i in lst:
5     total += i
6
7 print("Sum:", total)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program23.py  
Enter list: 1 3 6 9 10 20 5  
Sum: 54  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program24.py > [e] lst
1 lst = list(map(int, input("Enter list: ").split()))
2 unique = []
3
4 for i in lst:
5     if i not in unique:
6         unique.append(i)
7
8 print("List without duplicates:", unique)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program24.py  
Enter list: 1 5 3 2 0 4 2 5 9 5 2 4 2 0 1 5 9 4 2  
List without duplicates: [1, 5, 3, 2, 0, 4, 9]  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program25.py > ...
1 li=list(map(int,input('Enter the elements: ').split()))
2 li.sort()
3 print(li)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program25.py  
Enter the elements: 1 2 5 2 8 2 9 4 0 2  
[0, 1, 2, 2, 2, 4, 5, 8, 9]  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program26.py > [e] lst
1   lst = list(map(int, input("Enter list: ").split()))
2   lst.sort(reverse=True)
3   print("Descending order:", lst)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program26.py
Enter list: 2 8 4 0 2 8 5 4 8 3 1 9 3
Descending order: [9, 8, 8, 8, 5, 4, 4, 3, 3, 2, 2, 1, 0]
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program27.py > [e] lst
1   lst = list(set(map(int, input("Enter list: ").split())))
2   lst.sort(reverse=True)
3   print("Second largest element:", lst[1])
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program27.py
Enter list: 5 3 2 9 1 9 2 0
Second largest element: 5
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

## 28. Write a Python Program to merge two lists.

```
program28.py > [e] list1
1   list1 = list(map(int, input("Enter first list: ").split()))
2   list2 = list(map(int, input("Enter second list: ").split()))
3
4   merged = list1 + list2
5   print("Merged list:", merged)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program28.py
Enter first list: 8 0 1 4 2 6 8
Enter second list: 2 9 5 4 8 3
Merged list: [8, 0, 1, 4, 2, 6, 8, 2, 9, 5, 4, 8, 3]
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```

program29.py > [2] list1
1  list1 = list(map(int, input("Enter first list: ").split()))
2  list2 = list(map(int, input("Enter second list: ").split()))
3
4  common = []
5
6  for i in list1:
7      if i in list2 and i not in common:
8          common.append(i)
9
10 print("Common elements:", common)
11

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program29.py
 Enter first list: 8 3 8 3 6 2 8
 Enter second list: 9 2 8 5 4 8 2
 Common elements: [8, 2]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```

program29.py > [2] list1
1  lst = list(map(int, input("Enter list: ").split()))
2  k = int(input("Enter k: "))
3
4  k = k % len(lst)
5  rotated = lst[k:] + lst[:k]
6
7  print("Rotated list:", rotated)
8

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program30.py
 Enter list: 9 4 3 8 2 6 4 8 2 1
 Enter k: 3
 Rotated list: [8, 2, 6, 4, 8, 2, 1, 9, 4, 3]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```

program31.py > ...
1  num = int(input("Enter number: "))
2  temp = num
3  order = len(str(num))
4  total = 0
5
6  while temp > 0:
7      digit = temp % 10
8      total += digit ** order
9      temp //= 10
10
11 if total == num:
12     print("Armstrong number")
13 else:
14     print("Not an Armstrong number")

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program31.py
 Enter number: 153
 Armstrong number
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program32.py > num
1 num = int(input("Enter number: "))
2 total = 0
3
4 for i in range(1, num):
5     if num % i == 0:
6         total += i
7
8 if total == num:
9     print("Perfect number")
10 else:
11     print("Not a perfect number")
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program32.py  
Enter number: 6  
Perfect number  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program33.py > ...
1 num = int(input("Enter number: "))
2 rev = 0
3 temp = num
4
5 while temp > 0:
6     rev = rev * 10 + temp % 10
7     temp //= 10
8
9 if rev == num:
10    print("Palindrome")
11 else:
12    print("Not a palindrome")
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program33.py  
Enter number: 121  
Palindrome  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program34.py > ...
1 a = int(input("Enter first number: "))
2 b = int(input("Enter second number: "))
3
4 while b != 0:
5     a, b = b, a % b
6
7 print("GCD:", a)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program34.py  
Enter first number: 12  
Enter second number: 24  
GCD: 12  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program35.py > [e] a
1 a = int(input("Enter first number: "))
2 b = int(input("Enter second number: "))
3
4 x, y = a, b
5 while y:
6     x, y = y, x % y
7
8 gcd = x
9 lcm = (a * b) // gcd
10
11 print("LCM:", lcm)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program35.py
Enter first number: 12
Enter second number: 24
● LCM: 24
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program36.py > [e] num
1 num = int(input("Enter decimal number: "))
2 binary = ""
3
4 while num > 0:
5     binary = str(num % 2) + binary
6     num //= 2
7
8 print("Binary:", binary)
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program36.py
Enter decimal number: 123
Binary: 1111011
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program37.py > [e] binary
1 binary = input("Enter binary number: ")
2 decimal = 0
3 power = 0
4
5 for digit in binary[::-1]:
6     decimal += int(digit) * (2 ** power)
7     power += 1
8
9 print("Decimal:", decimal)
10

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program37.py
Enter binary number: 1010111
Decimal: 87
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
program38.py X
program38.py > [e] n
1 n = int(input("Enter n: "))
2
3 for num in range(2, n + 1):
4     for i in range(2, int(num ** 0.5) + 1):
5         if num % i == 0:
6             break
7         else:
8             print(num, end=" ")
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program38.py
Enter n: 15
2 3 5 7 11 13
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

```
program39.py X
program39.py > [e] n
1 n = int(input("Enter n: "))
2 total = n * (n + 1) // 2
3 print("Sum:", total)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program39.py
Enter n: 20
Sum: 210
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

### 40. Write a Python Program to calculate the power of a number without using '\*'.

```
program40.py X
program40.py > [e] base
● 1 base = int(input("Enter base: "))
2 exp = int(input("Enter exponent: "))
3
4 result = 1
5 for _ in range(exp):
6     result *= base
7
8 print("Result:", result)
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program40.py
Enter base: 2
Enter exponent: 4
● Result: 16
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

The screenshot shows a Python code editor interface. At the top, there are two tabs: "program41.py" and "program41.txt". The "program41.py" tab is active, displaying the following code:

```
1 file = open("program41.txt", "r")
2 print(file.read())
3 file.close()
4
```

Below the code editor are several tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following terminal session:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program41.py
- hi hello
- how are you?

At the bottom of the terminal window, there are two entries: "PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>" and "PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █".

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

The screenshot shows a Python code editor interface. At the top, there are two tabs: "program42.py" and "program42.txt". The "program42.py" tab is active, displaying the following code:

```
1 file = open("program42.txt", "w")
2 file.write("Hello, this is a text file.")
3
4 file.close()
```

Below the code editor are several tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following terminal session:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program42.py
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

Below the terminal, there is another code editor window titled "program42.txt". It contains the following text:

```
1 Hello, this is a text file.
```

At the bottom of the terminal window, there are two entries: "PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>" and "PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █".

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

The screenshot shows the VS Code interface with two tabs open: 'program43.py' and 'program43.txt'. The code in 'program43.py' reads a file named 'program43.txt' and prints its word count. The terminal shows the output: 'Word count: 4'. The contents of 'program43.txt' are also shown in the editor.

```

program43.py > ...
1 file = open("program43.txt", "r")
2 words = file.read().split()
3 print("Word count:", len(words))
4 file.close()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program43.py
Word count: 4
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
①

program43.py > ...
program43.txt > ...
1 Hi,My name is Geethika.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program43.py
Word count: 4
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
①

```

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

The screenshot shows the VS Code interface with two tabs open: 'sample.txt' and 'program44.py'. The code in 'program44.py' reads a file named 'sample.txt' and prints its line count. The terminal shows the output: 'Line count: 1'. The contents of 'sample.txt' are also shown in the editor.

```

sample.txt > ...
program44.py > ...

program44.py > ...
1 file = open("sample.txt", "r")
2 lines = file.readlines()
3 print("Line count:", len(lines))
4 file.close()

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

sample.txt > ...
program44.py > ...

sample.txt > ...
1 Hi,My name is Geethika.

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program44.py
Line count: 1
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
①
②
●

```

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

```

sample.txt X program45.py X sample1.txt program44.py
program45.py > ...
1 src = open("sample.txt", "r")
2 dest = open("sample1.txt", "w")
3
4 dest.write(src.read())

```

TERMINAL

```

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT> py program45.py
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT>

```

```

sample.txt program45.py sample1.txt program44.py
sample1.txt
1 Hi,My name is Geethika.

```

TERMINAL

```

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT> py program45.py
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT>

```

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

```

program46.py X
program46.py > ...
1 import os
2
3 filename = "sample.txt"
4 if os.path.exists(filename):
5     print("File exists")
6 else:
7     print("File does not exist")
8

```

TERMINAL

```

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT> py program46.py
File exists
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT>

```

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

```

program47.py X
program47.py > [e] file
1 file = open("sample.txt", "a")
2 file.write("\nThis text is appended.")
3 file.close()
4

```

TERMINAL

```

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT> py program47.py
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT>

```

```

sample.txt X
sample.txt
1 Hi,My name is Geethika.
2 This text is appended.

```

TERMINAL

```

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT> py program47.py
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs QOT>

```

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

```
program48.py > ...
1 file = open("sample.txt", "r")
2 words = file.read().split()
3 longest = max(words, key=len)
4 print("Longest word:", longest)
5 file.close()
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program48.py  
Longest word: Geethika.  
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program49.py > ...
1 file = open("sample.txt", "r")
2 lines = file.readlines()
3 file.close()
4
5 file = open("sample.txt", "w")
6 for line in lines:
7     if line.strip():
8         file.write(line)
9 file.close()
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program49.py  
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

```
sample.txt > ...
sample.txt
1 Hi, My name is Geethika.
2 This text is appended.
```

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

program50.py > ...  
1 import csv  
2  
3 with open("data.csv", "r") as file:  
4 reader = csv.reader(file)  
5 for row in reader:  
6 print(row)  
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program50.py  
● ['This is a csv file.'  
● ['that is related to program50 code']  
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>  
program50.py > data.csv >  
data.csv  
1 This is a csv file.  
2 that is related to program50 code  
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program50.py  
● ['This is a csv file.'  
● ['that is related to program50 code']  
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

program51.py > ...  
1 n = int(input("Enter number: "))  
2  
3 for i in range(1, 11):  
4 print(n, "x", i, "=", n \* i)  
5

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program51.py  
● Enter number: 5  
● 5 x 1 = 5  
● 5 x 2 = 10  
5 x 3 = 15  
5 x 4 = 20  
5 x 5 = 25  
② 5 x 6 = 30  
5 x 7 = 35  
5 x 8 = 40  
5 x 9 = 45  
5 x 10 = 50  
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

A screenshot of a Python code editor interface. At the top, there's a toolbar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. Below the toolbar is a terminal window showing the output of a Python script named 'program52.py'. The script prints odd numbers from 1 to 100. The terminal output is as follows:

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program52.py
Even number from 1 to 100
2 4 6 8 10 12 14 16 18 20 22 24 26 28 30 32 34 36 38 40 42 44 46 48 50 52 54 56 58 60 62 64 66 68 70 72 74 76 78 80 82 84 86 88 90 92 94 96 98
```

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

A screenshot of a Python code editor interface. At the top, there's a toolbar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. Below the toolbar is a terminal window showing the output of a Python script named 'program53.py'. The script prints odd numbers from 1 to 100. The terminal output is as follows:

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program53.py
Odd numbers from 1 to 100
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29 31 33 35 37 39 41 43 45 47 49 51 53 55 57 59 61 63 65 67 69 71 73 75 77 79 81 83 85 87 89 91 93 95 97 99
```

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

A screenshot of a Python code editor interface. At the top, there's a toolbar with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. Below the toolbar is a terminal window showing the output of a Python script named 'program54.py'. The script prompts the user to enter a number and then prints it. The terminal output is as follows:

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program54.py
Enter any number: 10
1 2 3 4 5 6 7 8 9 10
```

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

```
program55.py X
program55.py > ...
1
2 n = int(input("Enter number of rows: "))
3
4 for i in range(1, n + 1):
5     print(" " * (n - i) + "*" * (2 * i - 1))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program55.py  
Enter number of rows: 5  
\*  
\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*\*
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program55.py X program56.py X
program56.py > [e] n
program56.py > [e] n
1 n = int(input("Enter number of rows: "))
2
3 for i in range(n, 0, -1):
4     print(" " * (n - i) + "*" * (2 * i - 1))
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program56.py  
Enter number of rows: 5  
\*\*\*\*\*  
\*\*\*\*\*  
\*\*\*\*  
\*\*\*  
\*
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program55.py program57.py X
program57.py > [n]
1 n = int(input("Enter number of rows: "))
2
3 for i in range(n):
4     num = 1
5     print(" " * (n - i), end="")
6     for j in range(i + 1):
7         print(num, end=" ")
8         num = num * (i - j) // (j + 1)
9     print()
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program57.py  
Enter number of rows: 5  
1  
1 1  
1 2 1  
1 3 3 1  
1 4 6 4 1
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program55.py program58.py X
program58.py > [n]
1 n = int(input("Enter number of rows: "))
2 num = 1
3
4 for i in range(1, n + 1):
5     for j in range(i):
6         print(num, end=" ")
7         num += 1
8     print()
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program58.py  
Enter number of rows: 5  
1  
2 3  
4 5 6  
7 8 9 10  
11 12 13 14 15
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program55.py  program59.py X
python program59.py > ...
1  for num in range(2, 101):
2      for i in range(2, int(num ** 0.5) + 1):
3          if num % i == 0:
4              break
5          else:
6              print(num, end=" ")
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program59.py  
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program55.py  program60.py X
python program60.py > ...
python program60.py > ...
1  for i in range(1,101):
2      if i % 3 == 0 and i % 5 == 0:
3          print(i,end=" ")
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program60.py  
15 30 45 60 75 90
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program55.py  program61.py X
python program61.py > ...
1  def square(n):
2      return n * n
3
4  s=square(5)
5  print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program61.py  
25
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program62.py > is_prime
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
9 s=is_prime(7)
10 print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program62.py  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program63.py > ...
1 def factorial(n):
2     if n == 0 or n == 1:
3         return 1
4     return n * factorial(n-1)
5
6 f=factorial(5)
7 print(f)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program63.py  
120  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program63.py program64.py X
python program64.py > ...
1 def max_three(a, b, c):
2     return max(a, b, c)
3
4 m=max_three(5,10,4)
5 print(m)
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program64.py  
10
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program65.py > ...
python program65.py > ...
1 def reverse_string(s):
2     return s[::-1]
3
4 r=reverse_string('gqt')
5 print(r)
6
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program65.py  
tqg
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program65.py python program66.py > ...
python program66.py > ...
1 def count_vowels(s):
2     vowels = "aeiouAEIOU"
3     return sum(1 for ch in s if ch in vowels)
4
5 c=count_vowels('Geethika')
6 print(c)
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program66.py  
4
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program67.py) X
```

```
program67.py > ...
1 def is_palindrome(s):
2     return s == s[::-1]
3
4 p=is_palindrome('katak')
5 print(p)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program67.py
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program68.py) X
```

```
program68.py > ...
1 def sum_digits(n):
2     return sum(int(d) for d in str(n))
3
4 s=sum_digits(173)
5 print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Warning: PowerShell detected that you might be using a screen reader and I enable it, run 'Import-Module PSReadLine'.

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program68.py
- 11
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program69.py) X
```

```
program69.py > ⏺ fibonacci
1 def fibonacci(n):
2     a, b = 0, 1
3     for i in range(n):
4         print(a, end=" ")
5         a, b = b, a + b
6
7 f=fibonacci(10)
8 print(f)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program69.py
- 0 1 1 2 3 5 8 13 21 34 None
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program70.py > ...
1 def power(a, b):
2     if b == 0:
3         return 1
4     return a * power(a, b-1)
5
6 p=power(2,3)
7 print(p)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program70.py
- 8
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program71.py > ...
1 n = 5
2 def fact(n):
3     if n <= 1:
4         return 1
5     return n * fact(n-1)
6
7 print(fact(n))
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program71.py
- 120
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program72.py > ...
1 def fib(n):
2     if n <= 1:
3         return n
4     return fib(n-1) + fib(n-2)
5
6 n = 5
7 for i in range(n):
8     print(fib(i), end=" ")
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program72.py
- 0 1 1 2 3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program73.py > ...
1 def sum_n(n):
2     if n == 1:
3         return 1
4     return n + sum_n(n-1)
5
6 s=sum_n(10)
7 print(s)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program73.py  
55
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program74.py > ...
1 def reverse_rec(s):
2     if len(s) == 0:
3         return s
4     return reverse_rec(s[1:]) + s[0]
5
6 r=reverse_rec('Rajasree')
7 print(r)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program74.py  
eersajaR
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program75.py > ...
1 def palindrome_rec(s):
2     if len(s) <= 1:
3         return True
4     if s[0] != s[-1]:
5         return False
6     return palindrome_rec(s[1:-1])
7
8 p=palindrome_rec('madam')
9 print(p)
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program75.py  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program76.py X
program76.py > C:\Users\reeth\OneDrive\Desktop\Python Programs GQT\program76.py
1 def gcd(a, b):
2     if b == 0:
3         return a
4     return gcd(b, a % b)
5
6 g=gcd(5,10)
7 print(g)
8
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program76.py  
5
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
- █

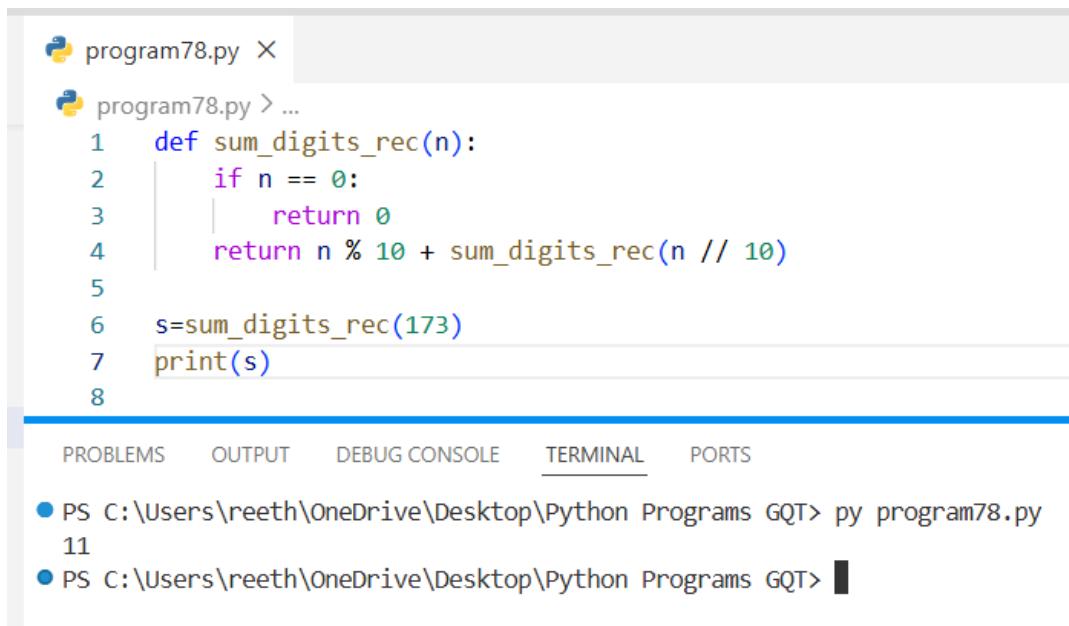
## 77. Write a Python Program to find LCM of two numbers using recursion.

```
program76.py X program77.py X
program77.py > lcm
1 def gcd(a, b):
2     if b == 0:
3         return a
4     return gcd(b, a % b)
5
6 def lcm(a, b):
7     return (a * b) // gcd(a, b)
8
9 # Example
10 a = 12
11 b = 18
12 print("LCM is:", lcm(a, b))
13
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program77.py  
LCM is: 36
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

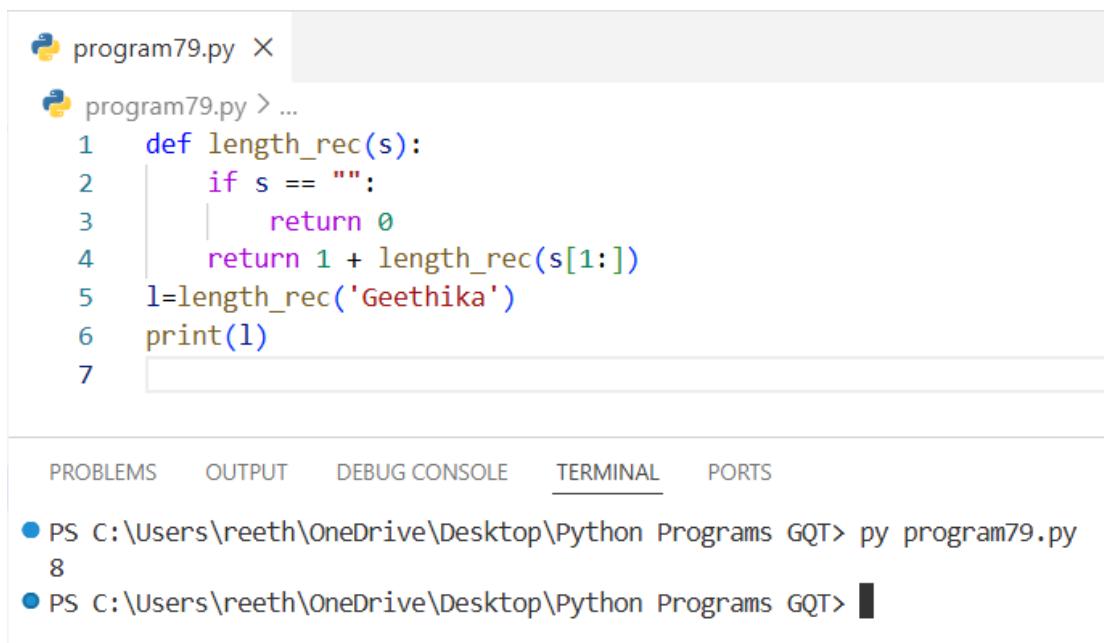


```
program78.py X
program78.py > ...
1 def sum_digits_rec(n):
2     if n == 0:
3         return 0
4     return n % 10 + sum_digits_rec(n // 10)
5
6 s=sum_digits_rec(173)
7 print(s)
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program78.py  
11
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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



```
program79.py X
program79.py > ...
1 def length_rec(s):
2     if s == "":
3         return 0
4     return 1 + length_rec(s[1:])
5 l=length_rec('Geethika')
6 print(l)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program79.py  
8
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor interface with two tabs at the top: "program79.py" and "program80.py X". The "program80.py" tab is active, displaying the following Python code:

```
program80.py > ...
1 def print_n(n):
2     if n == 0:
3         return
4     print(n)
5     print_n(n-1)
6
7 p=print_n(10)
8 print(p)
```

Below the code editor is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal section displays the following command-line interaction:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py pr
- 10
- 9
- 8
- 7
- 6
- 5
- 4
- 3
- 2
- 1
- None

○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor interface with three tabs at the top: "program79.py", "program80.py", and "program81.py X". The "program81.py" tab is active, displaying the following Python code:

```
program81.py > [o] t
1 t = (3, 7, 2, 9)
2 print(max(t))
3
```

Below the code editor is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. The terminal section displays the following command-line interaction:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program81.py
- 9

○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program79.py program80.py program82.py X
python program82.py > ...
1   t = (3, 7, 2, 9)
2   print(min(t))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program82.py
● 2
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> ●
```

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

```
program83.py X
python program83.py > ...
1   lst = [1, 2, 3]
2   t = tuple(lst)
3   print(t)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program83.py
● (1, 2, 3)
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> ●
```

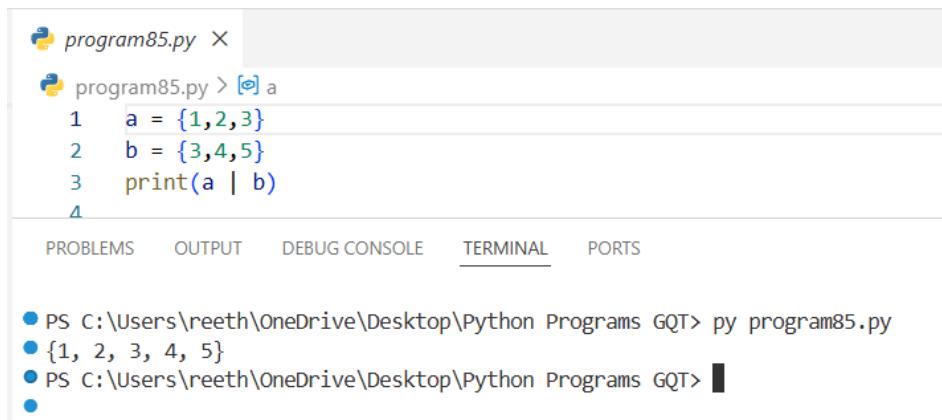
### 84. Write a Python Program to convert a tuple into a list.

```
program83.py program84.py X
python program84.py > ...
1   tuple=(1,2,3,4)
2   lst = list(tuple)
3   print(lst)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program84.py
● [1, 2, 3, 4]
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> ●
```

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

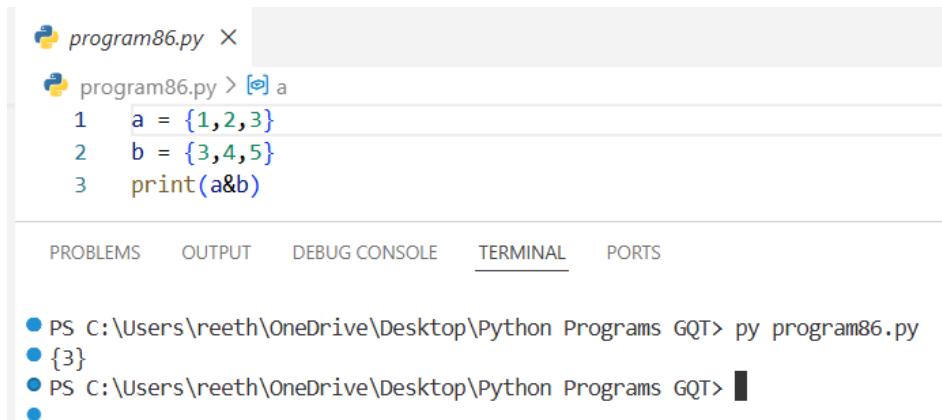


```
(program85.py) 1 a = {1,2,3} 2 b = {3,4,5} 3 print(a | b) 4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program85.py
- {1, 2, 3, 4, 5}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

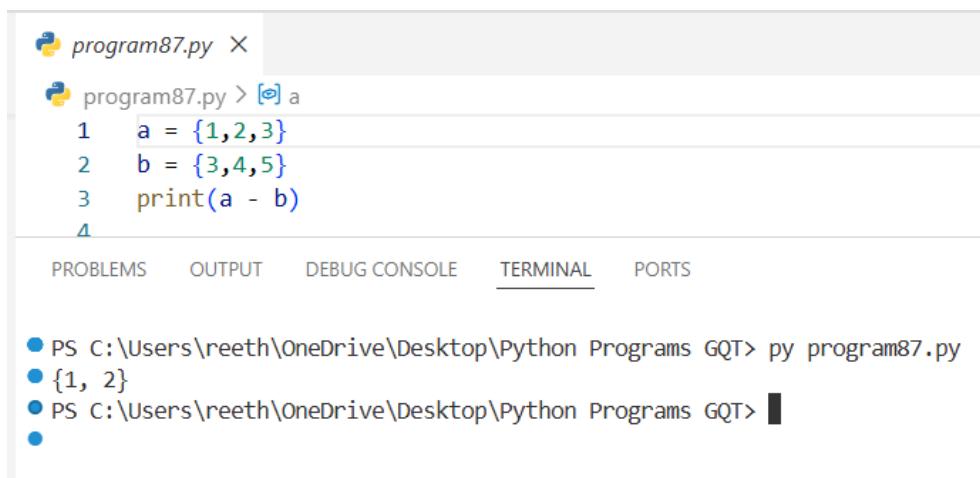


```
(program86.py) 1 a = {1,2,3} 2 b = {3,4,5} 3 print(a&b)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program86.py
- {3}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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



```
(program87.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\reeth\OneDrive\Desktop\Python Programs GQT> py program87.py
- {1, 2}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program88.py > [a]
1 a = {1,2,3}
2 b = {3,4,5}
3 print(a.issubset(b))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program87.py
- {1, 2}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
- 

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

```
program89.py > ...
1 lst = [1,2,2,3,3]
2 lst = list(set(lst))
3 print(lst)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program89.py
- [1, 2, 3]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
- 

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

```
program89.py program90.py >
program90.py > [a] lst
1 lst = [1,2,2,3,3]
2 print(len(set(lst)))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program90.py
- 3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
- 

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

```
python program91.py
```

```
program91.py > students
```

```
1 students = {"Geethika":90, "Ravi":85}  
2 print(students)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program91.py
```

```
● {'Geethika': 90, 'Ravi': 85}
```

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

```
●
```

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

```
python program92.py
```

```
program92.py > students
```

```
1 students = {"Geethika":90, "Ravi":85}  
2 print(students["Geethika"])  
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program92.py
```

```
● 90
```

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

```
●
```

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

```
python program93.py
```

```
program93.py > students
```

```
1 students = {"Geethika":90, "Ravi":85}  
2 students["Ravi"] = 95  
3 print(students)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program93.py
```

```
● {'Geethika': 90, 'Ravi': 95}
```

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

```
●
```

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

```
(program94.py) X
(program94.py) > [ɔ] students
1   students = {"Geethika":90, "Ravi":85}
2   del students["Ravi"]
3   print(students)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program94.py
- {'Geethika': 90}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
- 

## 95. Write a Python Program to merge two dictionaries.

```
(program95.py) X
(program95.py) > [ɔ] d1
1   d1 = {"a":1}
2   d2 = {"b":2}
3   d1.update(d2)
4   print(d1)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program95.py
- {'a': 1, 'b': 2}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program96.py) X
(program96.py) > ...
1   def char_freq(s):
2       d = {}
3       for ch in s:
4           d[ch] = d.get(ch,0)+1
5       return d
6
7   c=char_freq('Geethikageethugeeth')
8   print(c)
9

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program96.py
- {'G': 1, 'e': 6, 't': 3, 'h': 3, 'i': 1, 'k': 1, 'a': 1, 'g': 2, 'u': 1}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program97.py > ...
1 def word_freq(s):
2     d = {}
3     words = s.split()
4     for w in words:
5         d[w] = d.get(w,0)+1
6     return d
7
8 w=word_freq('GeethikaGeethika')
9 print(w)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program97.py  
{'Geethika': 1}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program97.py program98.py > ...
1 d = {"a":10,"b":20,"c":5}
2 print(max(d, key=d.get))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program98.py  
b
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program97.py program99.py > ...
1 d = {"a":10,"b":20,"c":5}
2 print(sorted(d.items(), key=lambda x:x[1]))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program99.py  
[('c', 5), ('a', 10), ('b', 20)]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

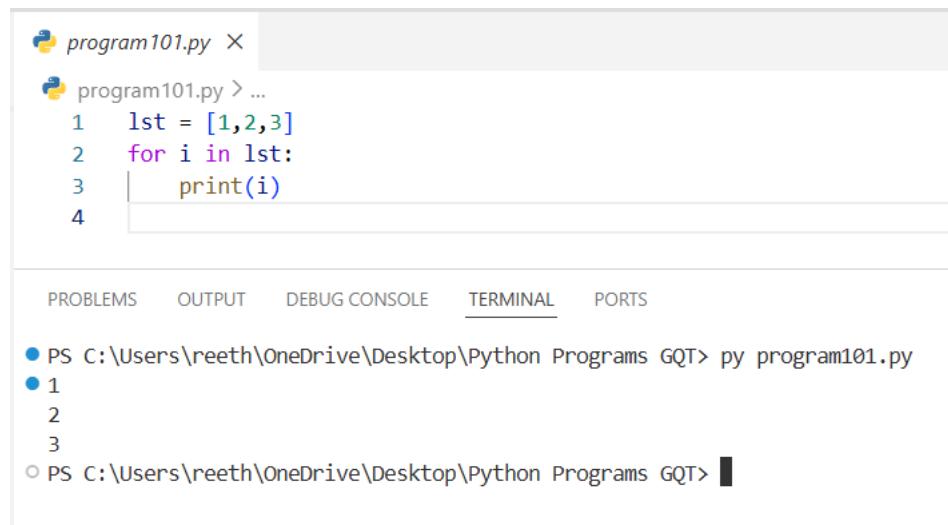


```
program100.py > 
1  d = {"a":10,"b":20,"c":5}
2  print("a" in d)
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program100.py  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

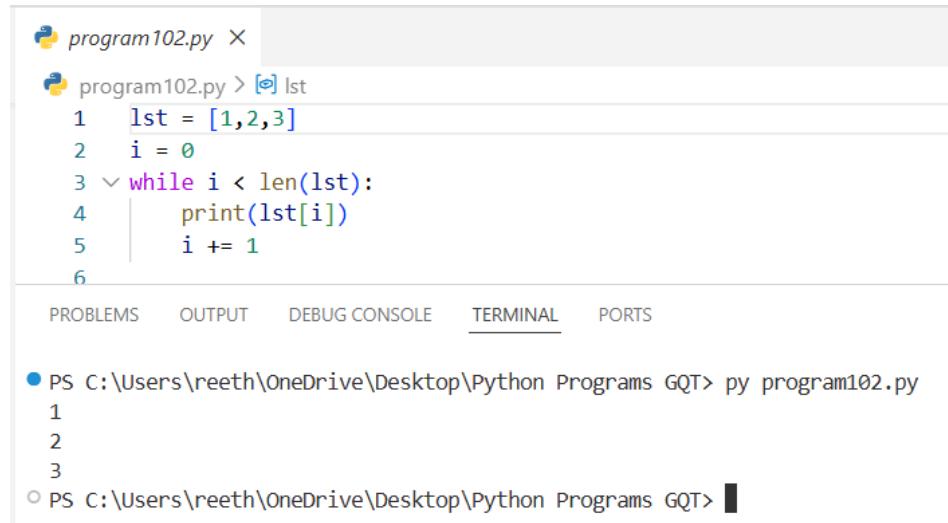


```
program101.py > ...
1  lst = [1,2,3]
2  for i in lst:
3      print(i)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program101.py  
1  
2  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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



```
program102.py > 
1  lst = [1,2,3]
2  i = 0
3  while i < len(lst):
4      print(lst[i])
5      i += 1
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program102.py  
1  
2  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program103.py X
program103.py > [ɔ] t
1   t = (1,2,3)
2   print(sum(t))
3

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program103.py  
6
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program104.py X
program104.py > [ɔ] lst
1   lst = [1,2,3]
2   print(2 in lst)
3

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program104.py  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program105.py X
program105.py > ...
1   s = "hello"
2   lst = list(s)
3   print(lst)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program105.py  
['h', 'e', 'l', 'l', 'o']
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program106.py > ...
1   lst = ["I", "love", "Python"]
2   print(" ".join(lst))
3
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program106.py  
I love Python
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program107.py > [s]
1   s = "I love programming in python"
2   print(max(s.split(), key=len))
3
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program107.py  
programming
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

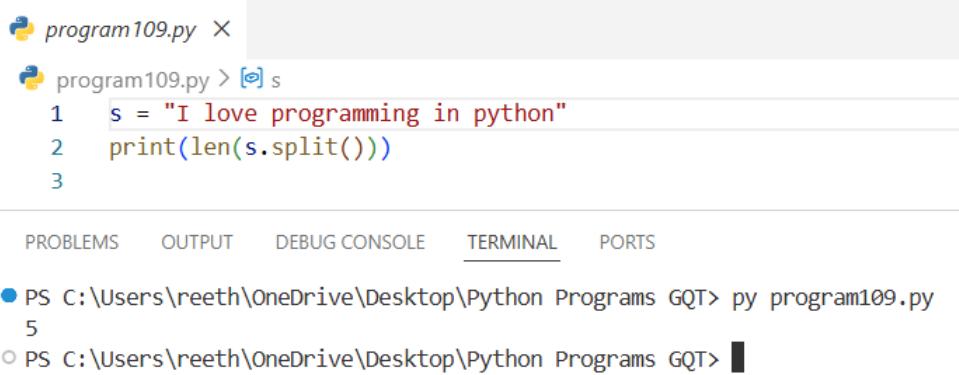
## 108. Write a Python Program to find the smallest word in a sentence.

```
program108.py > [s]
1   s = "I love programming in python"
2   print(min(s.split(), key=len))
3
```

TERMINAL

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program108.py  
I
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

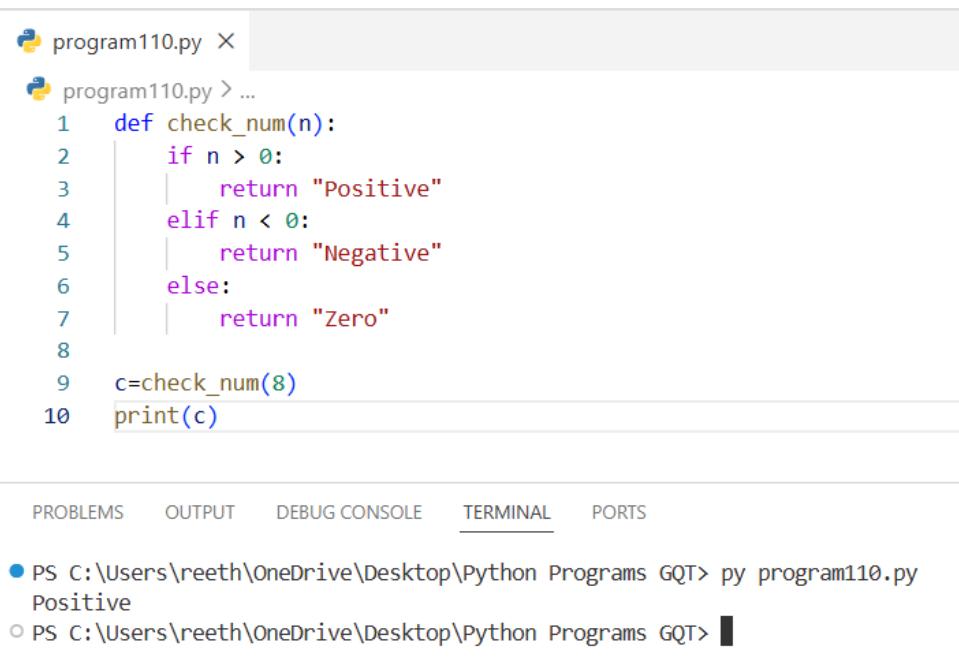


```
program109.py > 
program109.py > [s]
1   s = "I love programming in python"
2   print(len(s.split()))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program109.py
5
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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



```
program110.py > ...
program110.py > ...
1  def check_num(n):
2      if n > 0:
3          return "Positive"
4      elif n < 0:
5          return "Negative"
6      else:
7          return "Zero"
8
9  c=check_num(8)
10 print(c)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program110.py
Positive
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
(program111.py) X
(program111.py) > [?] year
1   year = int(input("Enter year: "))
2   if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
3       print("Leap Year")
4   else:
5       print("Not a Leap Year")
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program111.py  
Enter year: 2000  
Leap Year
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

## 112. Write a Python Program to calculate simple interest.

```
(program112.py) X
(program112.py) > ...
1   p = float(input("Principal: "))
2   r = float(input("Rate: "))
3   t = float(input("Time: "))
4   si = (p * r * t) / 100
5   print("Simple Interest:", si)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program112.py  
Principal: 5000  
Rate: 2  
Time: 12  
Simple Interest: 1200.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

## 113. Write a Python Program to calculate compound interest.

```
(program113.py) X
(program113.py) p
1 p = float(input("Principal: "))
2 r = float(input("Rate: "))
3 t = float(input("Time: "))
4 ci = p * (1 + r/100) ** t - p
5 print("Compound Interest:", ci)
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program113.py
Principal: 5000
Rate: 2
Time: 12
Compound Interest: 1341.2089728127276
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
(program114.py) X
(program114.py) ...
1 c = float(input("Celsius: "))
2 f = (c * 9/5) + 32
3 print("Fahrenheit:", f)
4
```

---

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program114.py
Celsius: 32
Fahrenheit: 89.6
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

```
(program115.py) X
(program115.py) > [ɔ] f
1   f = float(input("Fahrenheit: "))
2   c = (f - 32) * 5/9
3   print("Celsius:", c)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program115.py  
Fahrenheit: 48  
Celsius: 8.88888888888889
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program116.py) X
(program116.py) > [ɔ] r
1   r = float(input("Radius: "))
2   area = 3.14 * r * r
3   print("Area:", area)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program116.py  
Radius: 12  
Area: 452.1599999999997
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program117.py) X
(program117.py) > [ɔ] l
1   l = float(input("Length: "))
2   b = float(input("Breadth: "))
3   print("Area:", l * b)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program117.py  
Length: 20  
Breadth: 5  
Area: 100.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program118.py > 
program118.py > [e] b
1   b = float(input("Base: "))
2   h = float(input("Height: "))
3   print("Area:", 0.5 * b * h)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program118.py  
Base: 20  
Height: 4  
Area: 40.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program119.py > 
program119.py > [e] a
1   a = float(input("Side: "))
2   print("Perimeter:", 4 * a)
3

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program119.py  
Side: 24  
Perimeter: 96.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program120.py > 
program120.py > ...
1   l = float(input("Length: "))
2   b = float(input("Breadth: "))
3   print("Perimeter:", 2 * (l + b))
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program120.py  
Length: 20  
Breadth: 3  
Perimeter: 46.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program121.py X
```

```
python program121.py > [r] r
1   r = float(input("Radius: "))
2   print("Perimeter:", 2 * 3.14 * r)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program121.py  
Radius: 24  
Perimeter: 150.72
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program122.py X
```

```
python program122.py > ...
1   a = float(input("Side: "))
2   print("Volume:", a ** 3)
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program122.py  
Side: 6  
Volume: 216.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program123.py X
```

```
python program123.py > [r]
1   r = float(input("Radius: "))
2   h = float(input("Height: "))
3   print("Volume:", 3.14 * r * r * h)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program123.py  
Radius: 12  
Height: 8  
Volume: 3617.279999999997
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program124.py X
```

```
python program124.py > [r]
1 r = float(input("Radius: "))
2 print("Volume:", (4/3) * 3.14 * r ** 3)
3
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program124.py  
Radius: 12  
Volume: 7234.559999999995
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program125.py X
```

```
python program125.py > [a]
1 a = float(input("Side: "))
2 print("Surface Area:", 6 * a * a)
3
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program125.py  
Side: 3  
Surface Area: 54.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
python program126.py X
```

```
python program126.py > [r]
1 r = float(input("Radius: "))
2 h = float(input("Height: "))
3 print("Surface Area:", 2 * 3.14 * r * (r + h))
4
```

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program126.py  
Radius: 14  
Height: 3  
Surface Area: 1494.64
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program127.py) X
(program127.py) r
1   r = float(input("Radius: "))
2   print("Surface Area:", 4 * 3.14 * r * r)
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program127.py  
Radius: 12  
Surface Area: 1808.639999999999  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program128.py) X
(program128.py) ch
1   ch = input("Enter character: ")
2   print(ch.isupper())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program128.py  
Enter character: GEETHIKA  
True  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program129.py) X
(program129.py) ch
1   ch = input("Enter character: ")
2   print(ch.islower())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program129.py  
Enter character: geethu  
True  
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program130.py) X
(program130.py) > [ɔ] ch
1   ch = input("Enter character: ")
2   print(ch.isdigit())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program130.py
Enter character: 2026January
False
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program130.py
Enter character: 2026
True
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

```
(program131.py) X
(program131.py) > [ɔ] ch
1   ch = input("Enter character: ")
2   print(ch.isalpha())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program131.py
Enter character: apple
True
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

```
(program132.py) X
(program132.py) > [ɔ] ch
1   ch = input("Enter character: ")
2   if not ch.isalnum():
3       print("Special Symbol")
4   else:
5       print("Not Special Symbol")
6

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program132.py
Enter character: @
Special Symbol
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program132.py
Enter character: 102
Not Special Symbol
○ PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

```
program133.py X
program133.py > [s]
1   s = input("Enter string: ")
2   print(sum(1 for i in s if i.isupper()))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program133.py  
Enter string: GeethikaGeethuBandaru  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program134.py X
program134.py > [s]
1   s = input("Enter string: ")
2   print(sum(1 for i in s if i.islower()))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program134.py  
Enter string: geethikaGeethuBandaru  
19
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program135.py X
program135.py > [s]
1   s = input("Enter string: ")
2   print(sum(1 for i in s if i.isdigit()))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program135.py  
Enter string: 100  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program135.py  
Enter string: geethu2004  
4
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program136.py X
program136.py > [s]
1 s = input("Enter string: ")
2 print(sum(1 for i in s if not i.isalnum()))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program136.py  
Enter string: @#\$Geethik  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program137.py X
program137.py > ...
1 import string
2 s = input("Enter string: ")
3 print"".join(i for i in s if i not in string.punctuation)
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program137.py  
Enter string: Geethika.Geethu,Geethika.Bandaru,  
GeethikaGeethuGeethikaBandaru
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program138.py X
program138.py > [s]
1 s = input("Enter string: ")
2 print(s.replace(" ", "-"))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program138.py  
Enter string: Geethika Bandaru Geethu  
Geethika-Bandaru-Geethu
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program139.py) s
1 s = input("Enter string: ")
2 print(s.split())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program139.py  
Enter string: Geethika Geethu Bandaru  
['Geethika', 'Geethu', 'Bandaru']
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program140.py X
(program140.py) words
1 words = ["I", "love", "Python"]
2 print(" ".join(words))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program140.py  
I love Python
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program141.py X
(program141.py) a
1 a = int(input("Enter number: "))
2 b = int(input("Enter divisor: "))
3 if a % b == 0:
4     print("Divisible")
5 else:
6     print("Not Divisible")
7

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program141.py  
Enter number: 20  
Enter divisor: 5
- Divisible
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
python program142.py X
```

```
python program142.py > [n]
```

```
1 n = float(input("Enter number: "))
2 print("Square Root:", n ** 0.5)
3
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program142.py

Enter number: 49

Square Root: 7.0

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
python program143.py X
```

```
python program143.py > [n]
```

```
1 n = float(input("Enter number: "))
2 print("Cube Root:", n ** (1/3))
3
```

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program143.py

Enter number: 27

Cube Root: 3.0

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program144.py) X
```

```
program144.py > [a]
1   a = int(input("Base: "))
2   b = int(input("Exponent: "))
3   print(pow(a, b))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program144.py  
Base: 4  
Exponent: 2
  - 16
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program145.py X
```

```
program145.py > [n]
1   n = float(input("Enter number: "))
2   print(abs(n))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program145.py  
Enter number: 25  
25.0
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

#### 146. Write a Python Program to generate random numbers.

```
program146.py X
```

```
program146.py
1   import random
2   print(random.random())
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program146.py  
0.45425066488689636
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
-

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

The screenshot shows a Jupyter Notebook interface. A code cell titled "program147.py" contains the following Python code:

```
1 import random
2
3 a = int(input("Enter first number: "))
4 b = int(input("Enter second number: "))
5
6 num = random.randint(a, b)
7
8 print("Random integer between", a, "and", b, "is:", num)
9
```

Below the code cell, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the output of running the program:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program147.py
- Enter first number: 5
- Enter second number: 9
- Random integer between 5 and 9 is: 6
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a Jupyter Notebook interface. A code cell titled "program148.py" contains the following Python code:

```
1 import random
2 lst = [1,2,3,4,5]
3 random.shuffle(lst)
4 print(lst)
5
```

Below the code cell, there are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the output of running the program:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program148.py
- [3, 5, 1, 2, 4]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program147.py program149.py < ...  
1 import random  
2 lst = [10, 20, 30, 40]  
3 print(random.choice(lst))  
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program149.py  
20
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program147.py program150.py < ...  
program150.py  
1 import random  
2 print(random.randint(1,6))  
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program150.py  
3
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program147.py program151.py < ...  
program151.py  
1 import random  
2 print(random.choice(["Head", "Tail"]))  
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program151.py  
Head
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program147.py program152.py X
program152.py > ...
1 import random, string
2 chars = string.ascii_letters + string.digits
3 password = ''.join(random.choice(chars) for i in range(8))
4 print(password)
5
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program152.py  
xoNWoSV
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

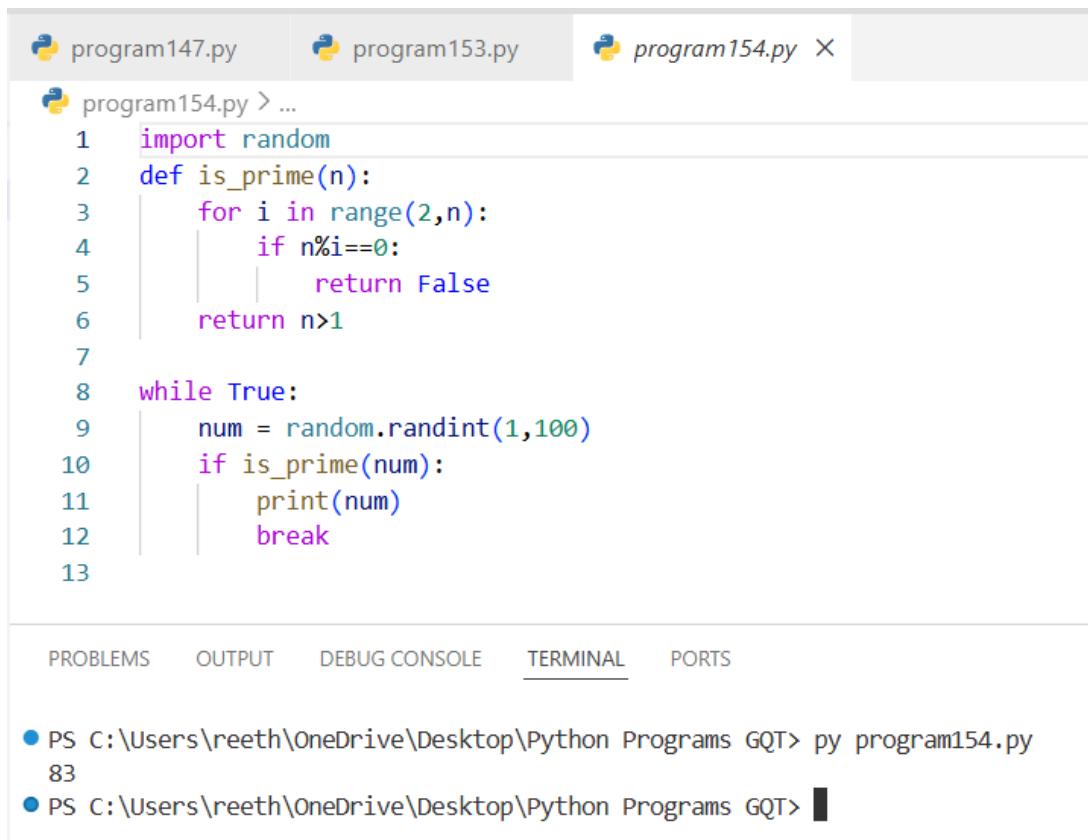
### 153. Write a Python Program to generate a random OTP.

```
program147.py program153.py X
program153.py > ...
1 import random
2 otp = random.randint(1000,9999)
3 print(otp)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program153.py  
8457
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

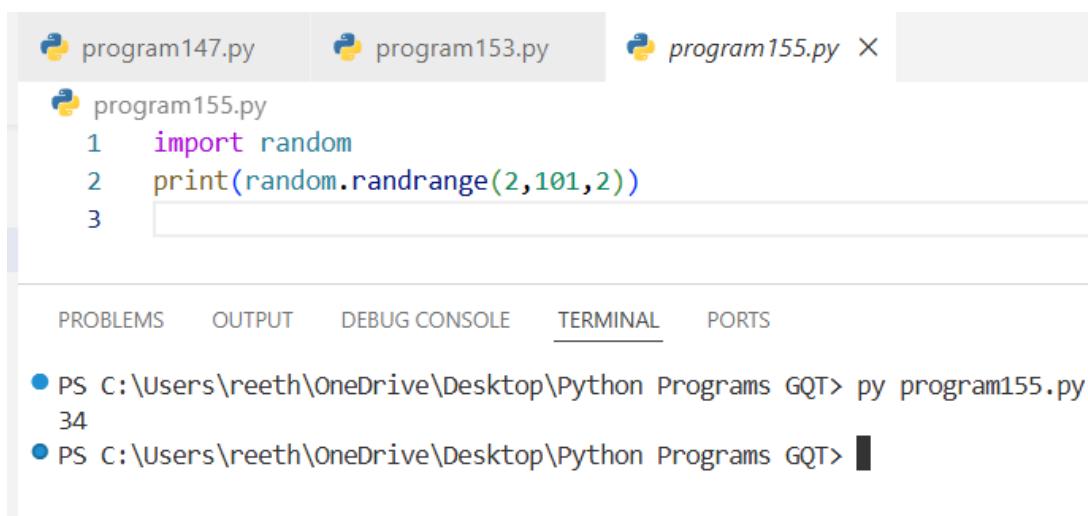


```
program154.py > ...
1 import random
2 def is_prime(n):
3     for i in range(2,n):
4         if n%i==0:
5             return False
6     return n>1
7
8 while True:
9     num = random.randint(1,100)
10    if is_prime(num):
11        print(num)
12        break
13
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program154.py  
83
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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



```
program155.py > ...
1 import random
2 print(random.randrange(2,101,2))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program155.py  
34
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program156.py) X
(program156.py)
1 import random
2 print(random.randrange(1,101,2))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program156.py  
55
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program157.py) X
(program157.py) > ...
1 import random, string
2 n = 5
3 print(''.join(random.choice(string.ascii_letters) for i in range(n)))
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program157.py  
ynUIV
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program158.py) X
(program158.py) > ...
1 import random, string
2 chars = string.ascii_letters + string.digits
3 print(''.join(random.choice(chars) for i in range(6)))
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program158.py  
XBTC18
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program159.py X
program159.py
1 import random
2 print(random.uniform(1,10))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program159.py  
7.701687530842528
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program160.py X
program160.py
1 import random
2 print(random.randint(50,100))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program160.py  
75
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program161.py X
program161.py > [lst
1 lst = []
2 if not lst:
3 | print("Empty")
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program161.py  
Empty
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program162.py) X
(program162.py) s
1 s = ""
2 if not s:
3     print("Empty")
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program162.py  
Empty
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program163.py) X
(program163.py) t
1 t = ()
2 if not t:
3     print("Empty")
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program163.py  
Empty
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
(program164.py) X
(program164.py) d
1 d = {}
2 if not d:
3     print("Empty")
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program164.py  
Empty
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
python program165.py X
python program165.py > [ɔ] lst
1   lst = [1,2,3]
2   lst.clear()
3   print(lst)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program165.py  
[]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
python program166.py X
python program166.py > [ɔ] d
1   d = {"a":1,"b":2}
2   d.clear()
3   print(d)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program166.py  
{}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
python program167.py X
python program167.py > [ɔ] s
1   s = {1,2,3}
2   s.clear()
3   print(s)
4

PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program167.py  
set()
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

## 168. Write a Python Program to copy a list.

The screenshot shows a code editor interface with a terminal window at the bottom. The terminal window displays the output of running a Python program named 'program168.py'.

```
program168.py > [?] lst1
1  lst1 = [1,2,3]
2  lst2 = lst1.copy()
3  print(lst2)
4
```

TERMINAL

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program168.py
[1, 2, 3]
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

## 169. Write a Python Program to copy a dictionary.

The screenshot shows a code editor interface with a terminal window at the bottom. The terminal window displays the output of running a Python program named 'program169.py'.

```
program169.py > [?] d1
1  d1 = {"a":1}
2  d2 = d1.copy()
3  print(d2)
4
```

TERMINAL

```
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program169.py
{'a': 1}
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

## 170. Write a Python Program to copy a set.

```
python program170.py
```

```
python program170.py > s1
1 s1 = {1,2,3}
2 s2 = s1.copy()
3 print(s2)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program170.py  
{1, 2, 3}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

### 171. Write a Python Program to reverse a list.

```
python program171.py
```

```
python program171.py > lst
1 lst = [1,2,3,4]
2 lst.reverse()
3 print(lst)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program171.py  
[4, 3, 2, 1]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

### 172. Write a Python Program to reverse a tuple.

```
python program172.py
```

```
python program172.py > t
1 t = (1,2,3,4)
2 print(t[::-1])
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program172.py  
(4, 3, 2, 1)
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

### 173. Write a Python Program to reverse a dictionary.

```
(program173.py) X
(program173.py) > [o] d
1  d = {'a': 1, 'b': 2, 'c': 3}
2  reversed_dict = {}
3  for k, v in d.items():
4      reversed_dict[v] = k
5  print(reversed_dict)
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program173.py  
{1: 'a', 2: 'b', 3: 'c'}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

## 174. Write a Python Program to reverse a set.

```
(program174.py) X
(program174.py) > ...
1  s = {1, 2, 3, 4, 5}
2  reversed_set = list(s)[::-1]
3  print(reversed_set)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

Warning: PowerShell detected that you might be using a screen reader and has enabled it, run 'Import-Module PSReadLine'.

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program174.py  
[5, 4, 3, 2, 1]
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
(program174.py) X (program175.py) X
(program175.py) > [o] sentence
1  sentence = input()
2  words = sentence.split()
3  print(" ".join(words[::-1]))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program175.py  
Geethika Geethu Bandaru  
Bandaru Geethu Geethika
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor window with two tabs: "program176.py" and "program177.py". The "program176.py" tab is active, displaying the following code:

```
1 sentence = input()
2 words = sentence.split()
3 for word in words:
4     print(word[::-1], end=" ")
5 
```

Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following terminal output:

- Warning: PowerShell detected that you might be using a screen reader and has disabled it, run 'Import-Module PSReadLine'.
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program176.py
- Geethika Geethu Bandaru  
akihteeG uhteeG uradnaB
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor window with two tabs: "program176.py" and "program177.py". The "program177.py" tab is active, displaying the following code:

```
1 with open("sample.txt", "r") as file:
2     lines = file.readlines()
3
4 lines.reverse()
5
6 for line in lines:
7     print(line, end="")
8 
```

Below the code editor are tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following terminal output:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py pr
This text is appended.Hi,My name is Geethika.
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program178.py X
program178.py > [e] num
1 num = int(input())
2 rev = 0
3 while num > 0:
4     rev = rev * 10 + num % 10
5     num //= 10
6 print(rev)
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program178.py  
254  
452
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program179.py X
program179.py > [e] nested
1 nested = [[1, 2], [3, 4], [5, 6]]
2 reversed_list = [sub[::-1] for sub in nested[::-1]]
3 print(reversed_list)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program179.py
[[6, 5], [4, 3], [2, 1]]
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

```
program180.py X
program180.py > [e] d
1 d = {'a': {'x': 1}, 'b': {'y': 2}}
2 reversed_dict = {k: {v2: k2 for k2, v2 in v.items()} for k, v in d.items()}
3 print(reversed_dict)
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program180.py  
{'a': {1: 'x'}, 'b': {2: 'y'}}
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program181.py X
program181.py > is_prime
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
9 num = int(input())
10 print(is_prime(num))
11
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program181.py  
29  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

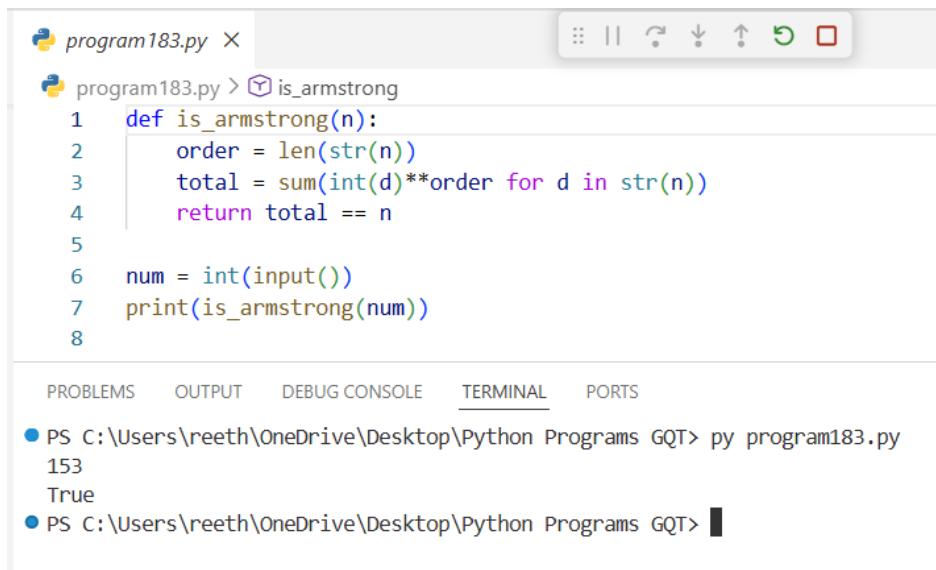
## 182. Write a Python Program to check if a string is palindrome using function.

```
program182.py X
program182.py > is_palindrome
1 def is_palindrome(s):
2     return s == s[::-1]
3
4 s = input()
5 print(is_palindrome(s))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program182.py  
madam  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program182.py  
circus  
False

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

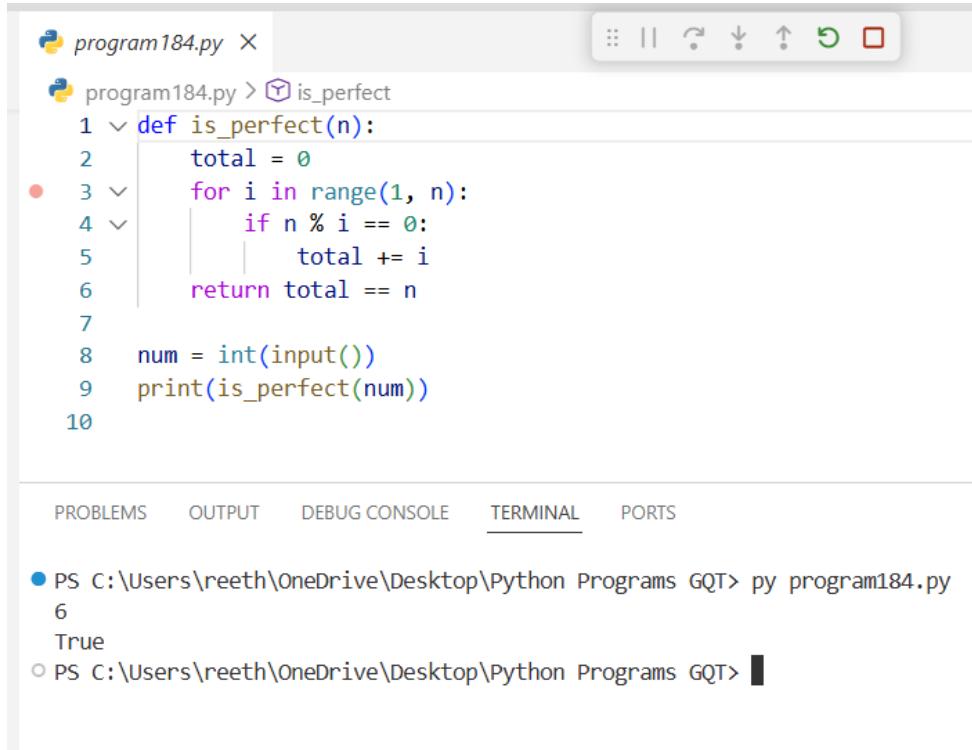


```
program183.py X
program183.py > is_armstrong
1 def is_armstrong(n):
2     order = len(str(n))
3     total = sum(int(d)**order for d in str(n))
4     return total == n
5
6 num = int(input())
7 print(is_armstrong(num))
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program183.py  
153  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

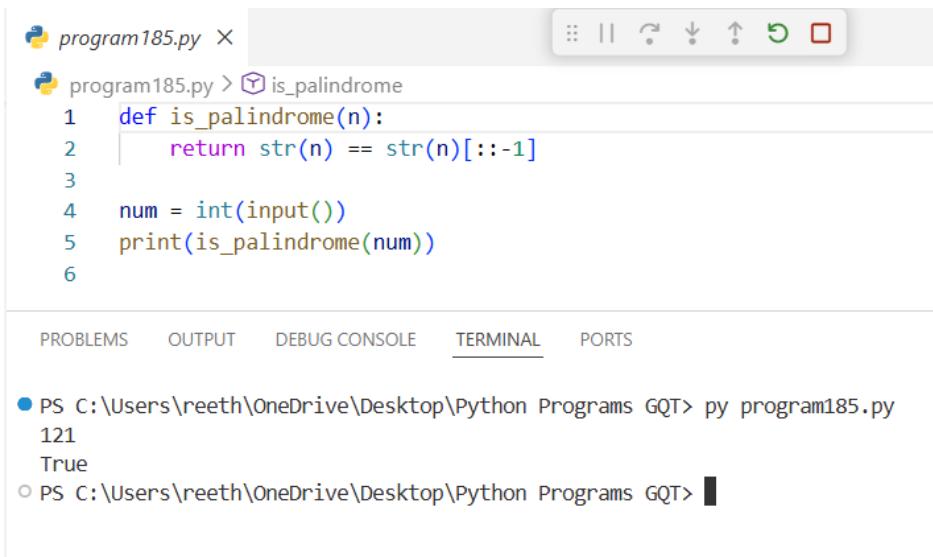


```
program184.py X
program184.py > is_perfect
1 def is_perfect(n):
2     total = 0
3     for i in range(1, n):
4         if n % i == 0:
5             total += i
6     return total == n
7
8 num = int(input())
9 print(is_perfect(num))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program184.py  
6  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

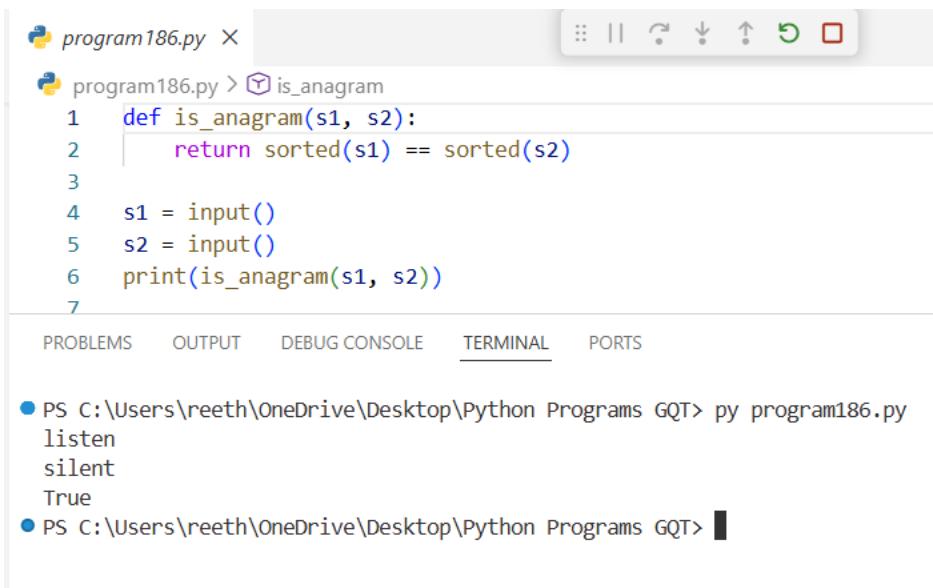


```
program185.py X
program185.py > is_palindrome
1 def is_palindrome(n):
2     return str(n) == str(n)[::-1]
3
4 num = int(input())
5 print(is_palindrome(num))
6
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program185.py  
121  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

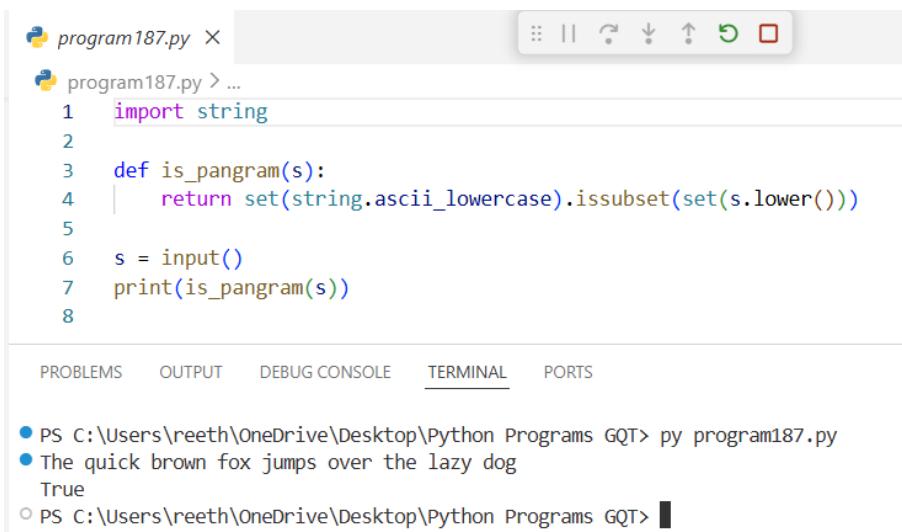


```
program186.py X
program186.py > is_anagram
1 def is_anagram(s1, s2):
2     return sorted(s1) == sorted(s2)
3
4 s1 = input()
5 s2 = input()
6 print(is_anagram(s1, s2))
7
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program186.py  
listen  
silent  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

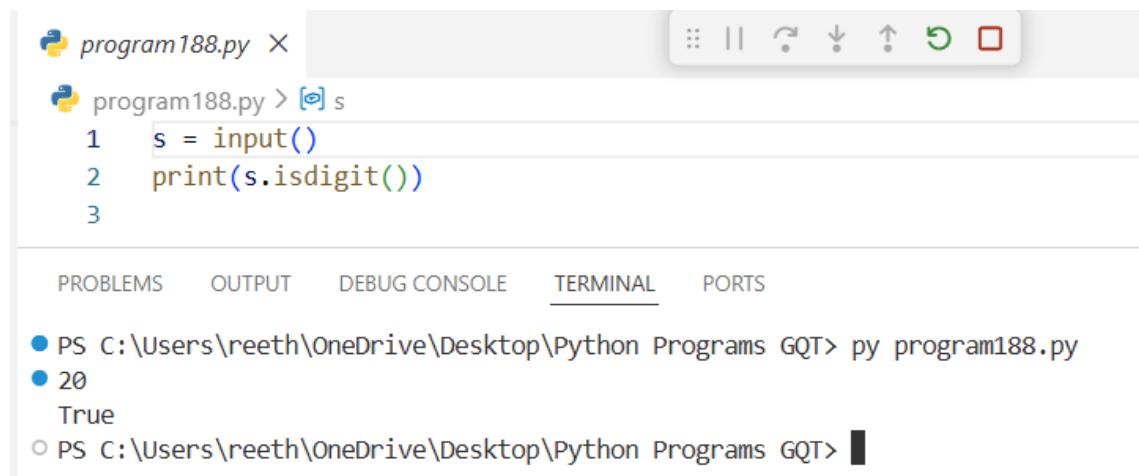


```
program187.py X
program187.py > ...
1 import string
2
3 def is_pangram(s):
4     return set(string.ascii_lowercase).issubset(set(s.lower()))
5
6 s = input()
7 print(is_pangram(s))
8
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program187.py
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> The quick brown fox jumps over the lazy dog  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

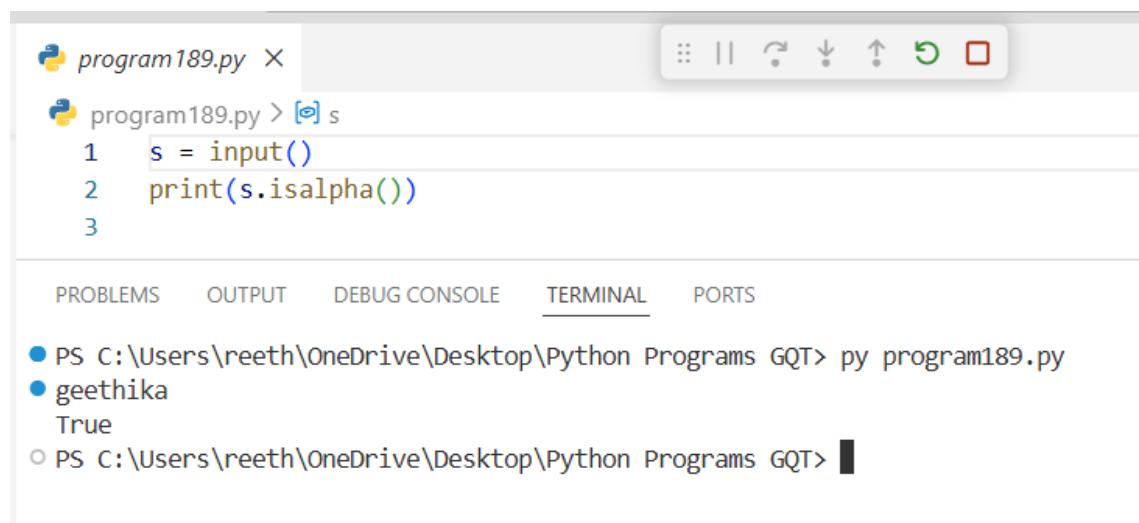


```
program188.py X
program188.py > [s]
1 s = input()
2 print(s.isdigit())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program188.py
- 20  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

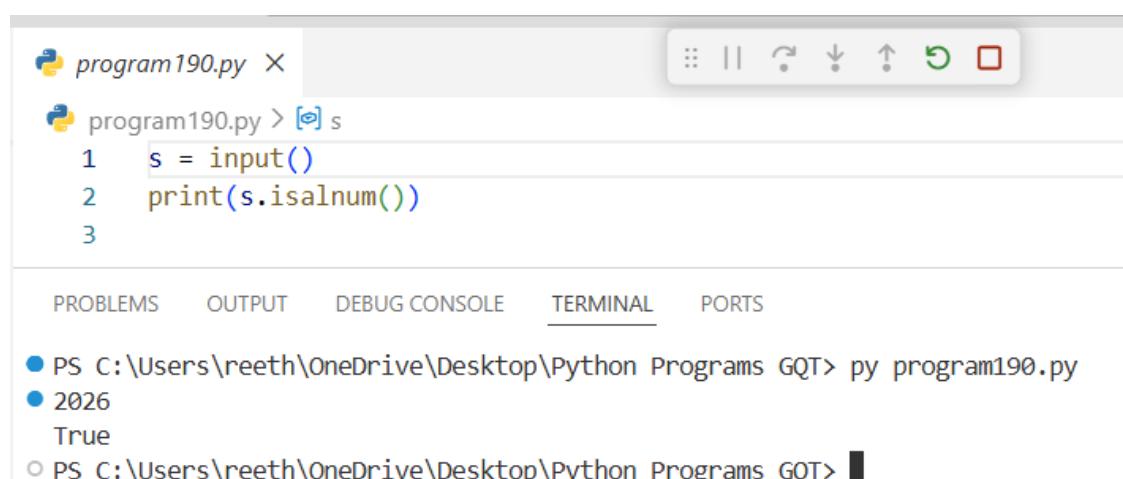


```
program189.py X
program189.py > [s]
1 s = input()
2 print(s.isalpha())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program189.py
- geethika  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

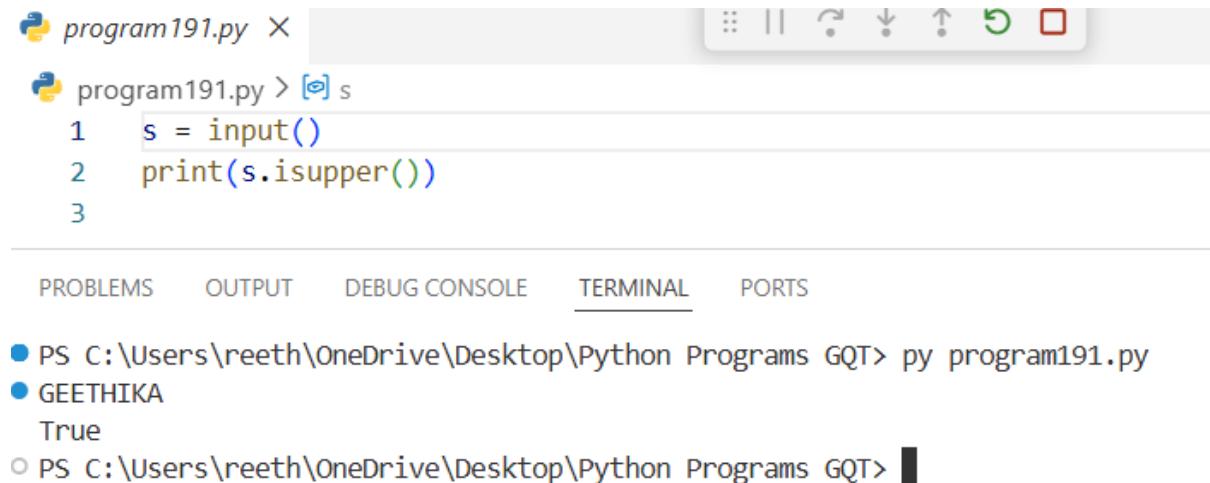


```
program190.py X
program190.py > [s]
1 s = input()
2 print(s.isalnum())
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program190.py
- 2026  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

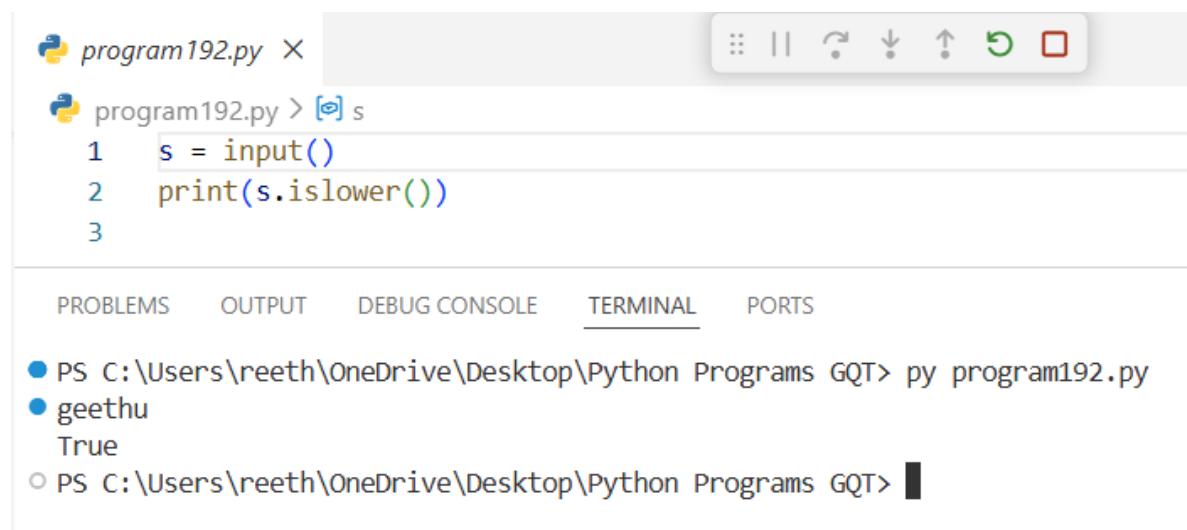


```
program191.py > s
1 s = input()
2 print(s.isupper())
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program191.py
- GEETHIKA  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

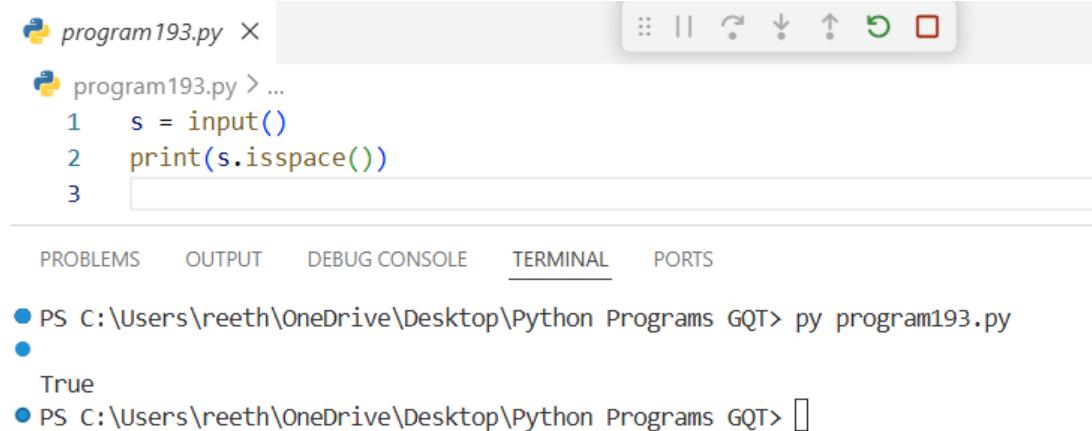


```
program192.py > s
1 s = input()
2 print(s.islower())
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program192.py
- geethu  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

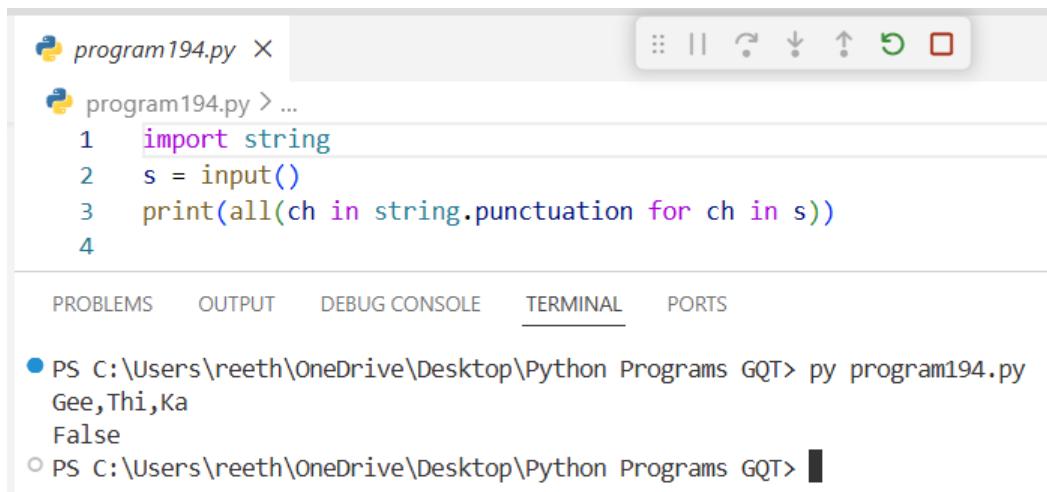


```
program193.py > ...
1 s = input()
2 print(s.isspace())
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program193.py
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

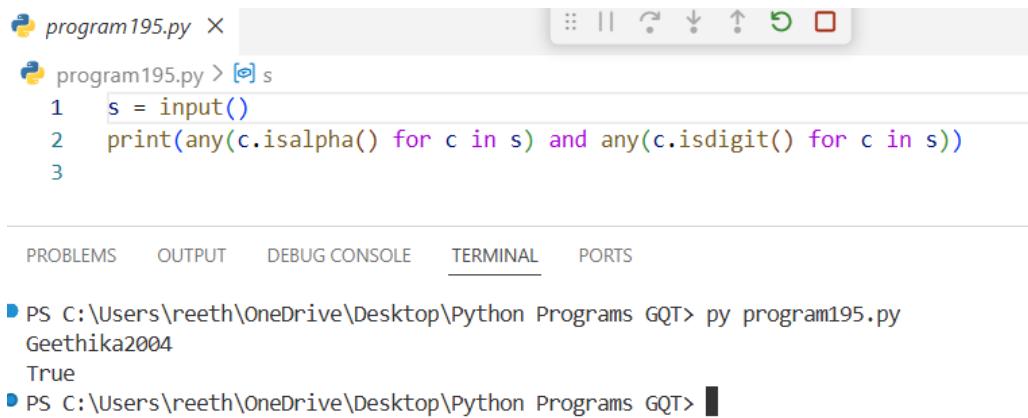


```
program194.py > ...
1 import string
2 s = input()
3 print(all(ch in string.punctuation for ch in s))
4
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program194.py  
Gee,Thi,Ka  
False
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

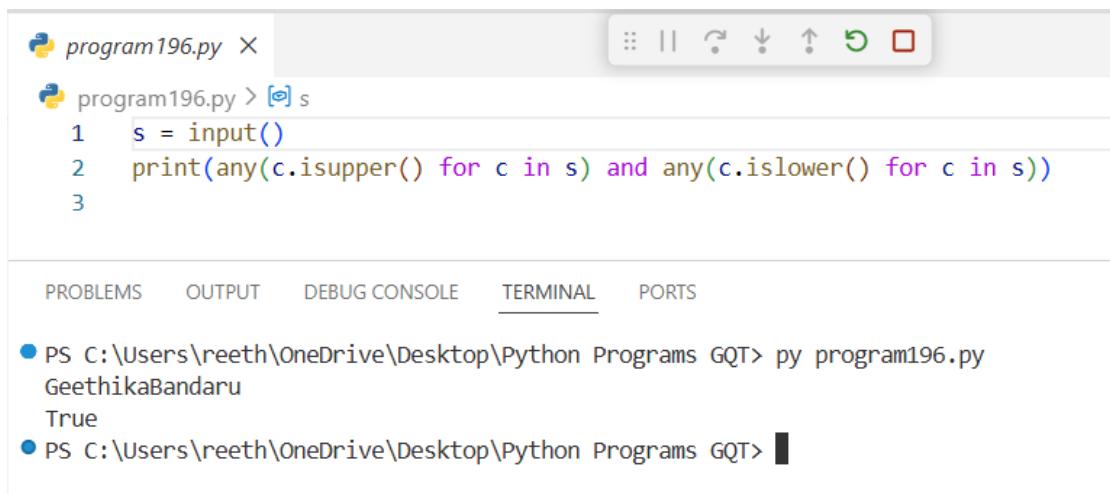


```
program195.py > ...
1 s = input()
2 print(any(c.isalpha() for c in s) and any(c.isdigit() for c in s))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program195.py  
Geethika2004  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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



```
program196.py > ...
1 s = input()
2 print(any(c.isupper() for c in s) and any(c.islower() for c in s))
3
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program196.py  
GeethikaBandaru  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

A screenshot of a code editor window titled "program197.py". The code defines a variable "vowels" as "aeiou" and prints "True" if the input string contains both vowels and consonants. The terminal shows the program running and outputting "True" for the input "GeethikaBandaru".

```
program197.py > [ɔ] s
1 s = input().lower()
2 vowels = "aeiou"
3 print(any(c in vowels for c in s) and any(c.isalpha() and c not in vowels for c in s))
4

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program197.py
GeethikaBandaru
True
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
```

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

A screenshot of a code editor window titled "program198.py". The code checks if the length of a set of the input string is equal to the length of the string itself, indicating if there are any repeated characters. The terminal shows the program running and outputting "True" for the input "Geethika".

```
program198.py > [ɔ] s
1 s = input()
2 print(len(set(s)) != len(s))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program198.py
Geethika
True
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
○
```

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

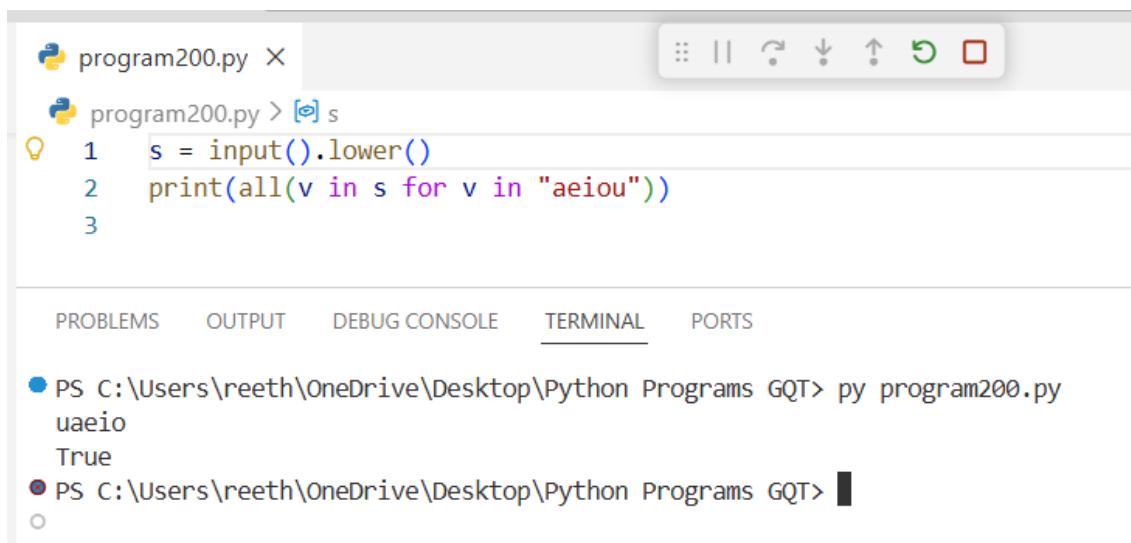
A screenshot of a code editor window titled "program199.py". The code checks if the length of a set of the input string is equal to the length of the string itself, indicating if all characters are unique. The terminal shows the program running and outputting "True" for the input "orange".

```
program199.py > [ɔ] s
1 s = input()
2 print(len(set(s)) == len(s))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program199.py
orange
True
● PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █
○
```

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

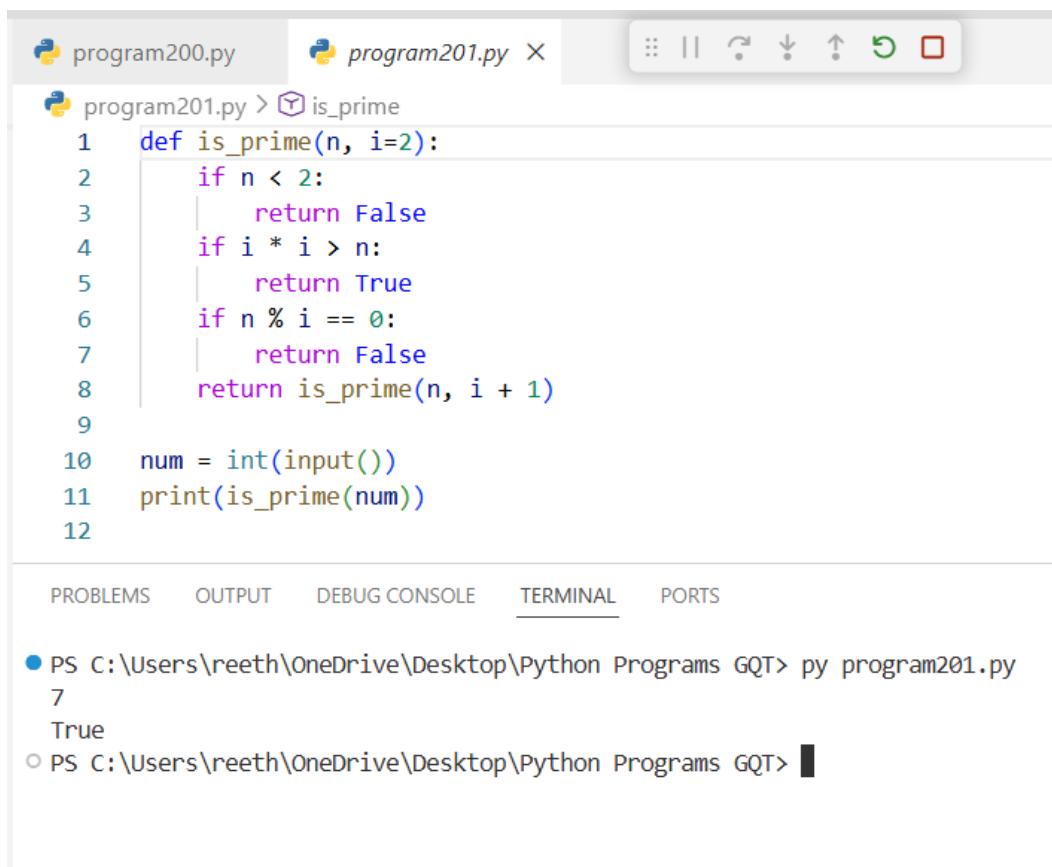


```
program200.py X
program200.py > [s]
1 s = input().lower()
2 print(all(v in s for v in "aeiou"))
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program200.py  
uaeio  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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



```
program200.py program201.py X
program201.py > [is_prime
1 def is_prime(n, i=2):
2     if n < 2:
3         return False
4     if i * i > n:
5         return True
6     if n % i == 0:
7         return False
8     return is_prime(n, i + 1)
9
10 num = int(input())
11 print(is_prime(num))
12

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
```

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program201.py  
7  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program200.py program202.py <input>
program202.py > is_palindrome
1 def is_palindrome(s):
2     if len(s) <= 1:
3         return True
4     if s[0] != s[-1]:
5         return False
6     return is_palindrome(s[1:-1])
7
8 s = input()
9 print(is_palindrome(s))
10
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program202.py  
katak  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program200.py program203.py <input>
program203.py > armstrong
1 def armstrong(n, order):
2     if n == 0:
3         return 0
4     return (n % 10) ** order + armstrong(n // 10, order)
5
6 num = int(input())
7 order = len(str(num))
8 print(armstrong(num, order) == num)
9
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program203.py  
153  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program204.py". The "program204.py" tab is active, displaying the following Python code:

```
1 def perfect(n, i=1, total=0):
2     if i == n:
3         return total
4     if n % i == 0:
5         total += i
6     return perfect(n, i + 1, total)
7
8 num = int(input())
9 print(perfect(num) == num)
10
```

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

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program204.py
- 6
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program205.py". The "program205.py" tab is active, displaying the following Python code:

```
1 def reverse(n, rev=0):
2     if n == 0:
3         return rev
4     return reverse(n // 10, rev * 10 + n % 10)
5
6 num = int(input())
7 print(reverse(num) == num)
8
```

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

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program205.py
- 525
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program206.py". The "program206.py" tab is active, displaying the following Python code:

```
program206.py > anagram
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 s1 = input()
9 s2 = input()
10 print(anagram(s1, s2))
11
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal output section shows the following command and its execution:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program206.py
- listen
- silent
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program207.py". The "program207.py" tab is active, displaying the following Python code:

```
program207.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 s = input()
11 print(pangram(s))
12
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal output section shows the following command and its execution:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program207.py
- The quick brown fox jumps over the lazy dog
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor with two tabs: "program200.py" and "program208.py". The "program208.py" tab is active, displaying the following Python code:

```
program208.py > only_digits
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 s = input()
9 print(only_digits(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. Under the TERMINAL tab, there are two entries:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program208.py  
20260402  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor with two tabs: "program200.py" and "program209.py". The "program209.py" tab is active, displaying the following Python code:

```
program209.py > only_alpha
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 s = input()
9 print(only_alpha(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS. Under the TERMINAL tab, there are two entries:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program209.py  
Padmavathi  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program210.py X". The "program210.py" tab is active, displaying the following Python code:

```
program210.py > only_alnum
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 s = input()
9 print(only_alnum(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following command-line session:

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program210.py
Geethika2004
True
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program211.py X". The "program211.py" tab is active, displaying the following Python code:

```
program211.py > only_upper
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 s = input()
9 print(only_upper(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is selected, showing the following command-line session:

```
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program211.py
GEETHIKA
True
PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>
```

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program212.py X". The "program212.py" tab is active, displaying the following Python code:

```
program212.py > only_lower
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 s = input()
9 print(only_lower(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS. Under the TERMINAL tab, the terminal output is shown:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program212.py  
geethika  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program213.py X". The "program213.py" tab is active, displaying the following Python code:

```
program213.py > only_space
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 s = input()
9 print(only_space(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (underlined), and PORTS. Under the TERMINAL tab, the terminal output is shown:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program213.py  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor with two tabs: "program200.py" and "program214.py". The "program214.py" tab is active, displaying the following Python code:

```
1 import string
2
3 def only_special(s):
4     if s == "":
5         return True
6     if s[0] not in string.punctuation:
7         return False
8     return only_special(s[1:])
9
10 s = input()
11 print(only_special(s))
12
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal window shows the following output:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program214.py  
@#\$  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor with two tabs: "program200.py" and "program215.py". The "program215.py" tab is active, displaying the following Python code:

```
1 def check(s, has_alpha=False, has_digit=False):
2     if s == "":
3         return has_alpha and has_digit
4     if s[0].isalpha():
5         has_alpha = True
6     if s[0].isdigit():
7         has_digit = True
8     return check(s[1:], has_alpha, has_digit)
9
10 s = input()
11 print(check(s))
12
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal window shows the following output:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program215.py  
Geethika2004  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

```
program216.py > check_case
1 def check_case(s, has_upper=False, has_lower=False):
2     if s == "":
3         return has_upper and has_lower
4     if s[0].isupper():
5         has_upper = True
6     if s[0].islower():
7         has_lower = True
8     return check_case(s[1:], has_upper, has_lower)
9
10 s = input()
11 print(check_case(s))
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program216.py  
GeethikaBandaru  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

```
program217.py > check_vowel_consonant
1 def check_vowel_consonant(s, has_vowel=False, has_consonant=False):
2     if s == "":
3         return has_vowel and has_consonant
4     ch = s[0].lower()
5     if ch.isalpha():
6         if ch in "aeiou":
7             has_vowel = True
8         else:
9             has_consonant = True
10    return check_vowel_consonant(s[1:], has_vowel, has_consonant)
11
12 s = input()
13 print(check_vowel_consonant(s))
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program217.py  
Geethika  
True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> █

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program218.py". The "program218.py" tab is active, displaying the following Python code:

```
1 def has_repeated(s):
2     if len(s) <= 1:
3         return False
4     if s[0] in s[1:]:
5         return True
6     return has_repeated(s[1:])
7
8 s = input()
9 print(has_repeated(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal output shows the execution of the program:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program218.py
- Geethika
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a code editor window with two tabs: "program200.py" and "program219.py". The "program219.py" tab is active, displaying the following Python code:

```
1 def is_unique(s):
2     if len(s) <= 1:
3         return True
4     if s[0] in s[1:]:
5         return False
6     return is_unique(s[1:])
7
8 s = input()
9 print(is_unique(s))
10
```

Below the code editor, there is a navigation bar with tabs: PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL (which is underlined), and PORTS.

The terminal output shows the execution of the program:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program219.py
- Orange
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>

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

The screenshot shows a Python code editor interface. At the top, there are two tabs: "program200.py" and "program220.py X". Below the tabs is a toolbar with icons for file operations. The main area displays the following Python code:

```
program220.py > contains_all_vowels
1 def contains_all_vowels(s, vowels=set("aeiou")):
2     if not vowels:
3         return True
4     if s == "":
5         return False
6     return contains_all_vowels(s[1:], vowels - {s[0].lower()})
7
8 s = input()
9 print(contains_all_vowels(s))
10
```

Below the code, there are tabs for "PROBLEMS", "OUTPUT", "DEBUG CONSOLE", "TERMINAL", and "PORTS". The "TERMINAL" tab is selected, showing the following command-line session:

- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT> py program220.py
- uaeio
- True
- PS C:\Users\reeth\OneDrive\Desktop\Python Programs GQT>



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 | www.gqtech.in