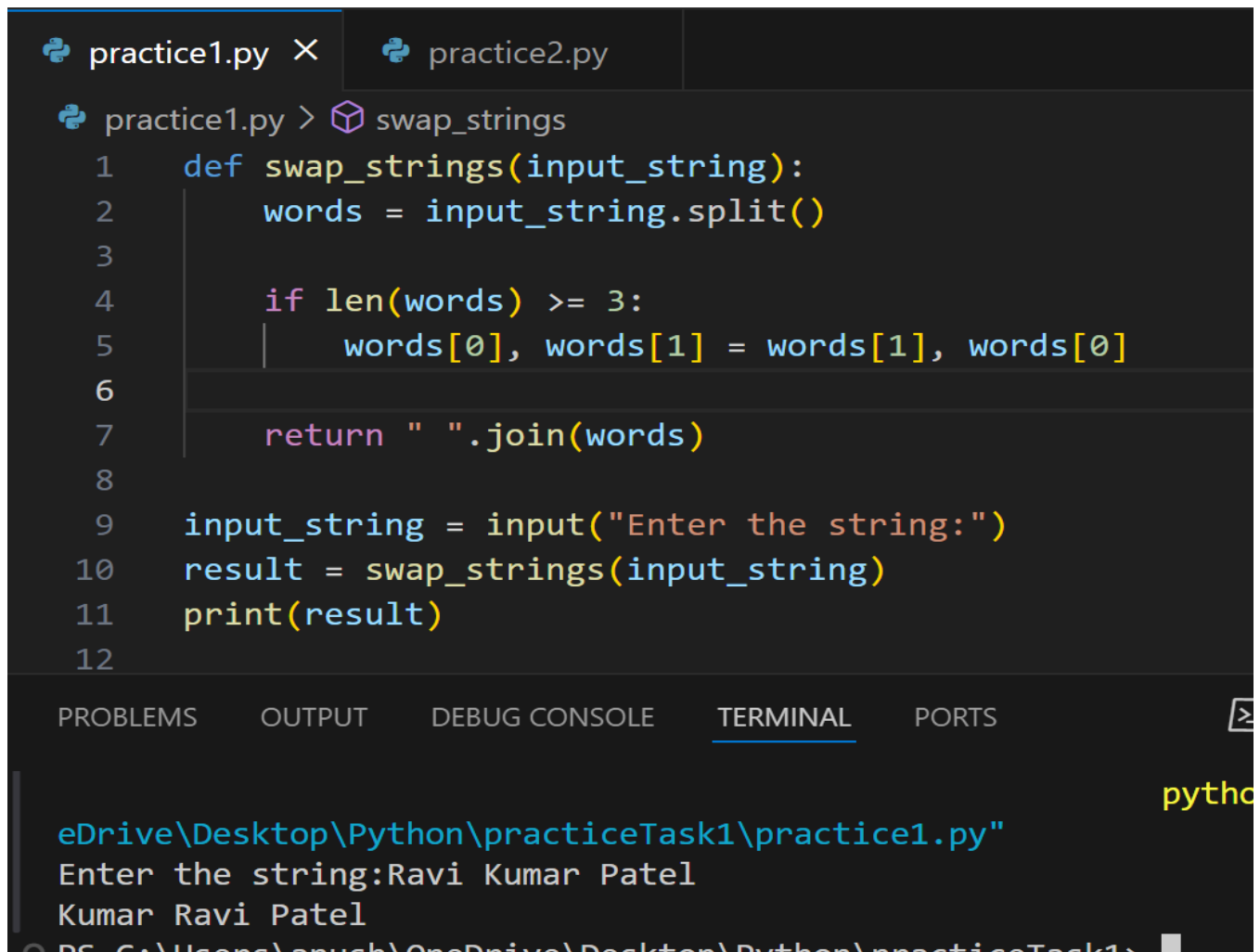


# Practice 1

Q1 Write a python program to swap below string. Input= Ravi Kumar Patel,  
Output= Patel Kumar Ravi.

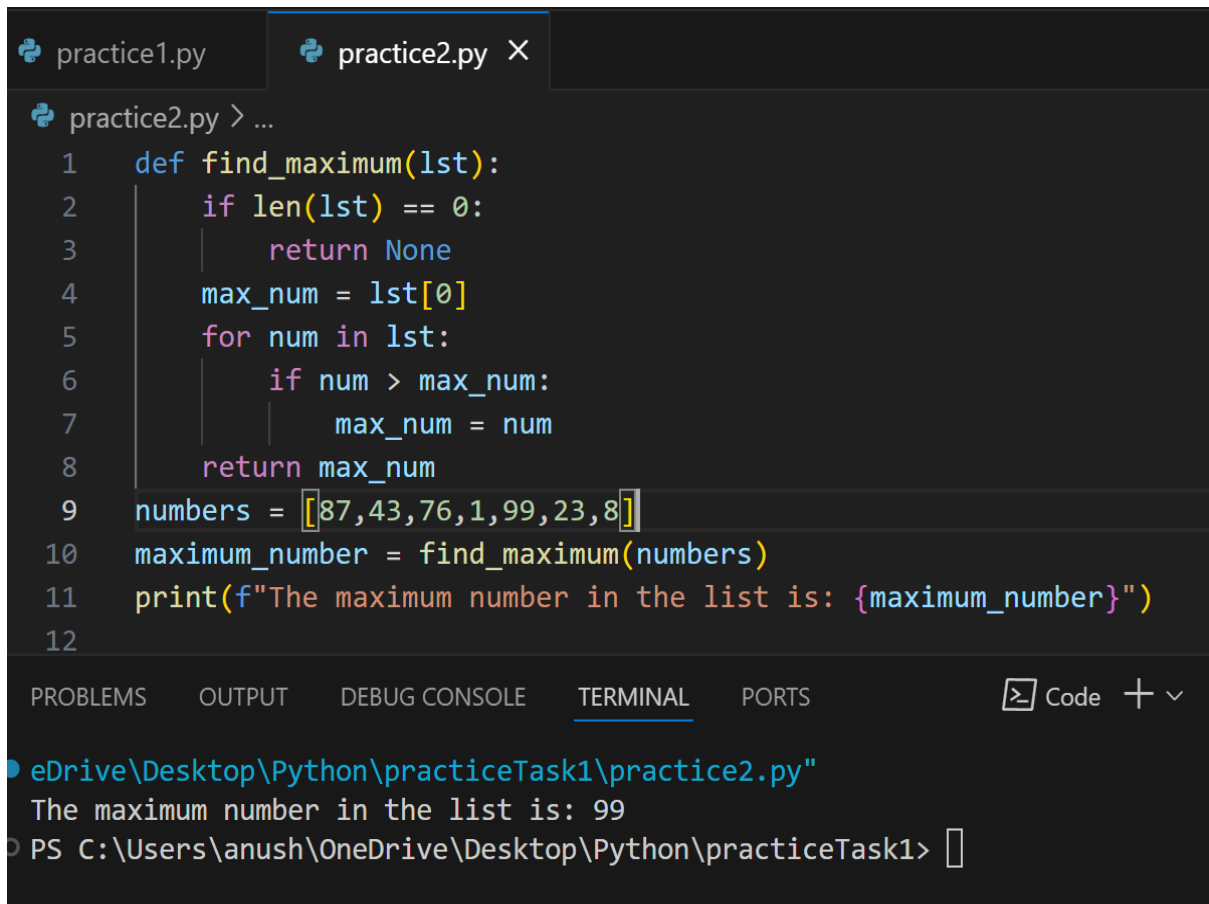


```
practice1.py ×  practice2.py
practice1.py > swap_strings
1  def swap_strings(input_string):
2      words = input_string.split()
3
4      if len(words) >= 3:
5          words[0], words[1] = words[1], words[0]
6
7      return " ".join(words)
8
9  input_string = input("Enter the string:")
10 result = swap_strings(input_string)
11 print(result)
12

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS


eDrive\Desktop\Python\practiceTask1\practice1.py"
Enter the string:Ravi Kumar Patel
Kumar Ravi Patel
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
```

Q2 Write a program to find maximum number from the list without using sorting list.



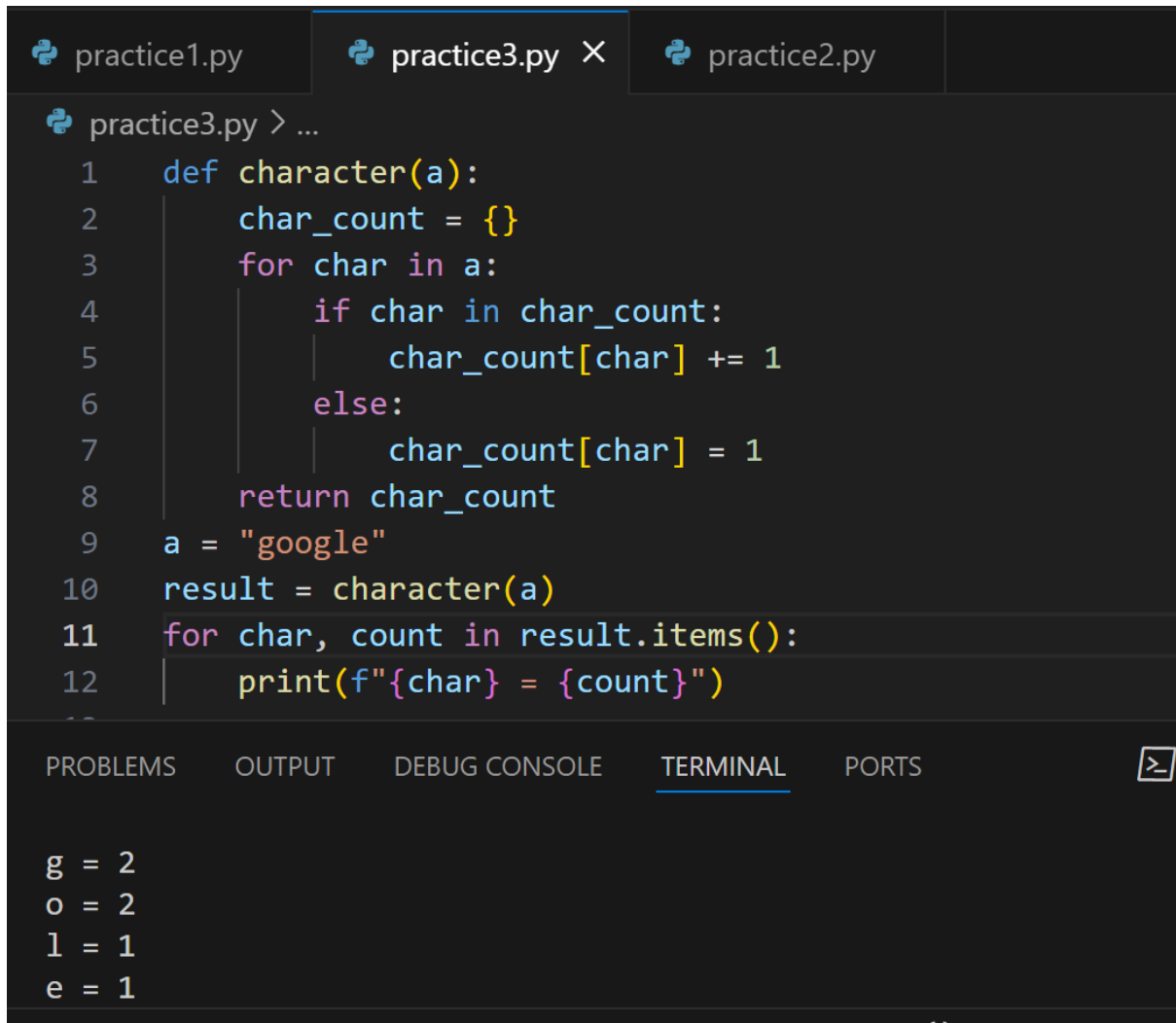
The screenshot shows a Python IDE with two tabs: 'practice1.py' and 'practice2.py'. The 'practice2.py' tab is active, displaying a Python script. The script defines a function 'find\_maximum(lst)' that iterates through a list to find the maximum value. It then uses this function to find the maximum in a list of numbers and prints the result. The terminal output shows the execution of the script, confirming the maximum value is 99.

```
practice2.py > ...
1  def find_maximum(lst):
2      if len(lst) == 0:
3          return None
4      max_num = lst[0]
5      for num in lst:
6          if num > max_num:
7              max_num = num
8      return max_num
9  numbers = [87,43,76,1,99,23,8]
10 maximum_number = find_maximum(numbers)
11 print(f"The maximum number in the list is: {maximum_number}")
12
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  Code + -

- eDrive\Desktop\Python\practiceTask1\practice2.py"
- The maximum number in the list is: 99
- PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>

Q3 Write a python program to find the occurrences of characters in a string  
s1=google ,output=g=2,o=2.....



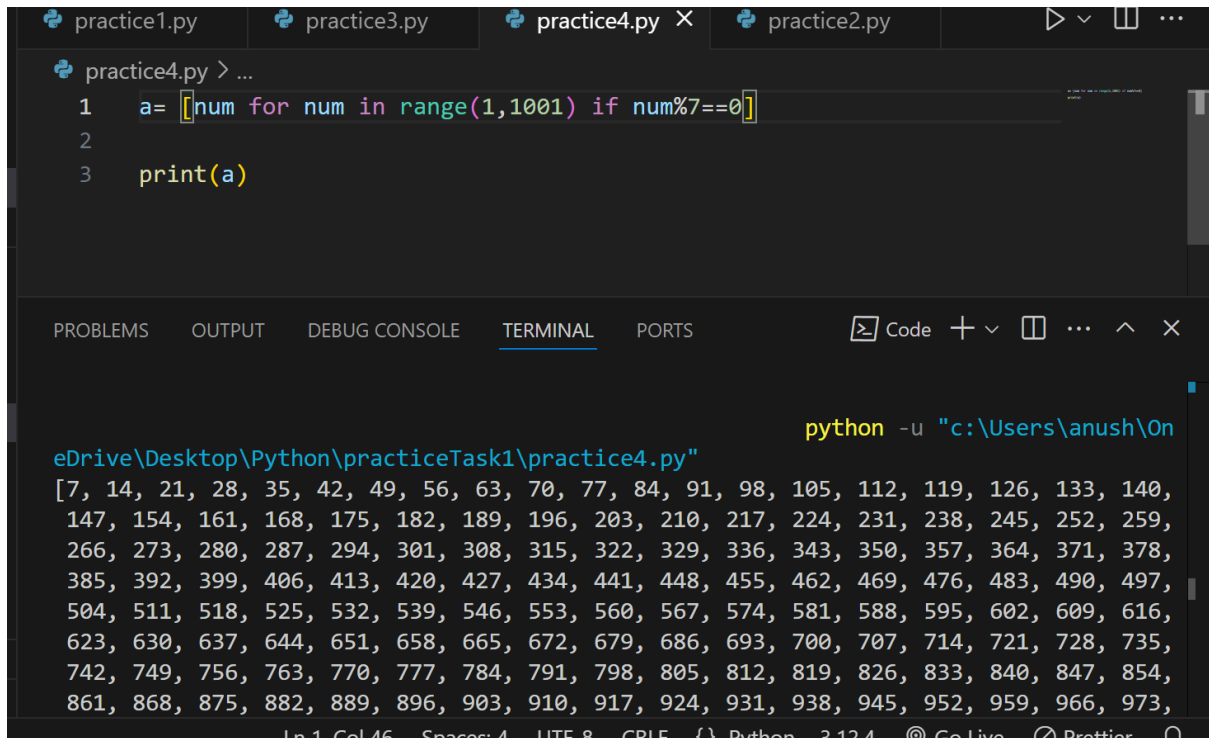
The image shows a screenshot of a Python IDE with three tabs: 'practice1.py', 'practice3.py' (active), and 'practice2.py'. The code in 'practice3.py' defines a function 'character(a)' that counts the occurrences of each character in a string 'a'. It uses a dictionary 'char\_count' to store the counts. The string 'a' is set to 'google', and the function is called to produce the output.

```
practice3.py > ...
1  def character(a):
2      char_count = {}
3      for char in a:
4          if char in char_count:
5              char_count[char] += 1
6          else:
7              char_count[char] = 1
8      return char_count
9  a = "google"
10 result = character(a)
11 for char, count in result.items():
12     print(f"{char} = {count}")
```

The output in the terminal is:

```
g = 2
o = 2
l = 1
e = 1
```

Q4 Use list comprehensions to solve below problem Find all numbers from 1-1000 divisible by 7.



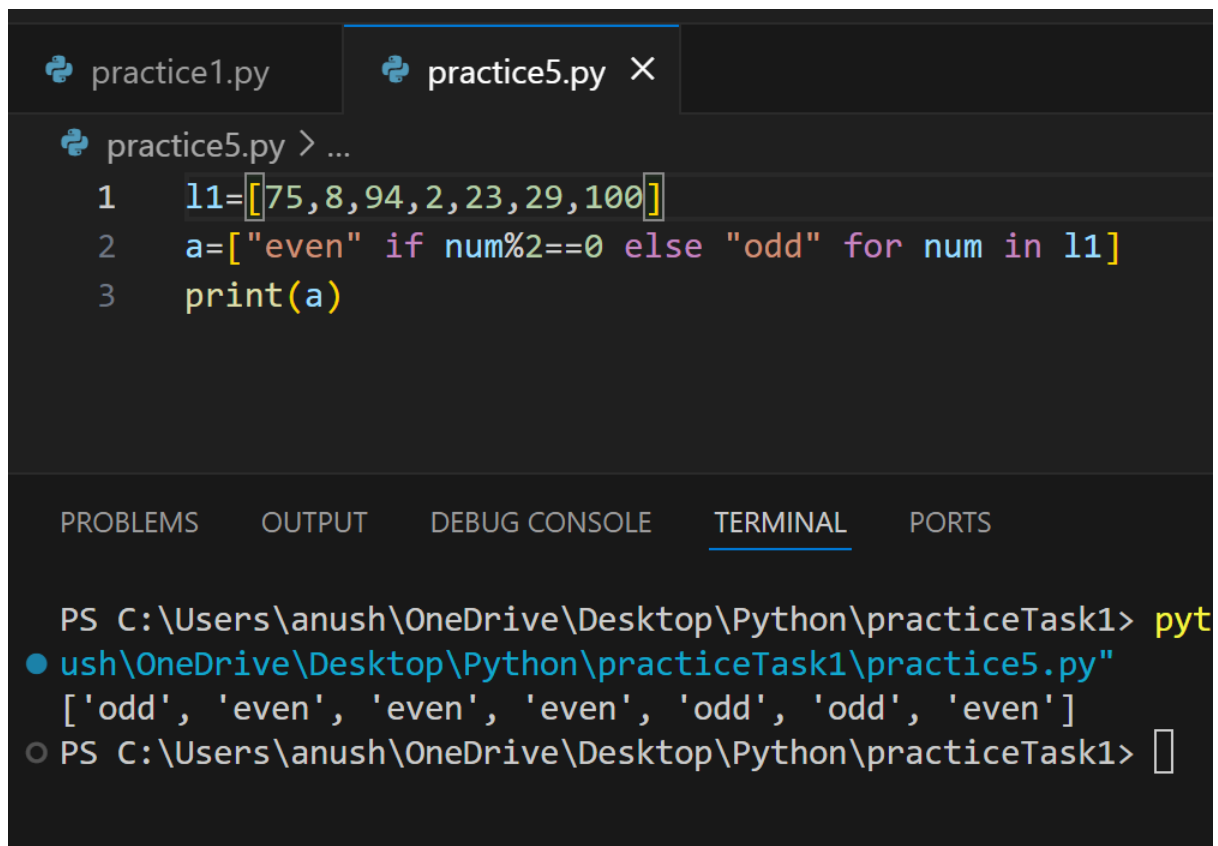
The image shows a screenshot of a Python IDE with a dark theme. At the top, there are tabs for 'practice1.py', 'practice3.py', 'practice4.py' (which is active and has a close button), and 'practice2.py'. The editor area shows the following code in 'practice4.py':

```
1 a= [num for num in range(1,1001) if num%7==0]
2
3 print(a)
```

Below the editor is a panel with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL' (which is active), and 'PORTS'. The terminal shows the command 'python -u "c:\Users\anush\On' and the output of the program, which is a long list of numbers from 7 to 994, all divisible by 7. The status bar at the bottom indicates 'Ln 1, Col 46', 'Spaces: 4', 'UTF-8', 'CRLF', '{ }', 'Python', '3.12.4', '@ Go Live', and 'Prettier'.

```
python -u "c:\Users\anush\On
eDrive\Desktop\Python\practiceTask1\practice4.py"
[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119, 126, 133, 140,
147, 154, 161, 168, 175, 182, 189, 196, 203, 210, 217, 224, 231, 238, 245, 252, 259,
266, 273, 280, 287, 294, 301, 308, 315, 322, 329, 336, 343, 350, 357, 364, 371, 378,
385, 392, 399, 406, 413, 420, 427, 434, 441, 448, 455, 462, 469, 476, 483, 490, 497,
504, 511, 518, 525, 532, 539, 546, 553, 560, 567, 574, 581, 588, 595, 602, 609, 616,
623, 630, 637, 644, 651, 658, 665, 672, 679, 686, 693, 700, 707, 714, 721, 728, 735,
742, 749, 756, 763, 770, 777, 784, 791, 798, 805, 812, 819, 826, 833, 840, 847, 854,
861, 868, 875, 882, 889, 896, 903, 910, 917, 924, 931, 938, 945, 952, 959, 966, 973,
```

Q5 Produce list containing words even if numbers is even else print odd.



The image shows a code editor with two tabs: 'practice1.py' and 'practice5.py'. The 'practice5.py' tab is active, displaying the following Python code:

```
1 l1=[75,8,94,2,23,29,100]
2 a=["even" if num%2==0 else "odd" for num in l1]
3 print(a)
```

Below the code editor is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is selected, showing the command prompt output:

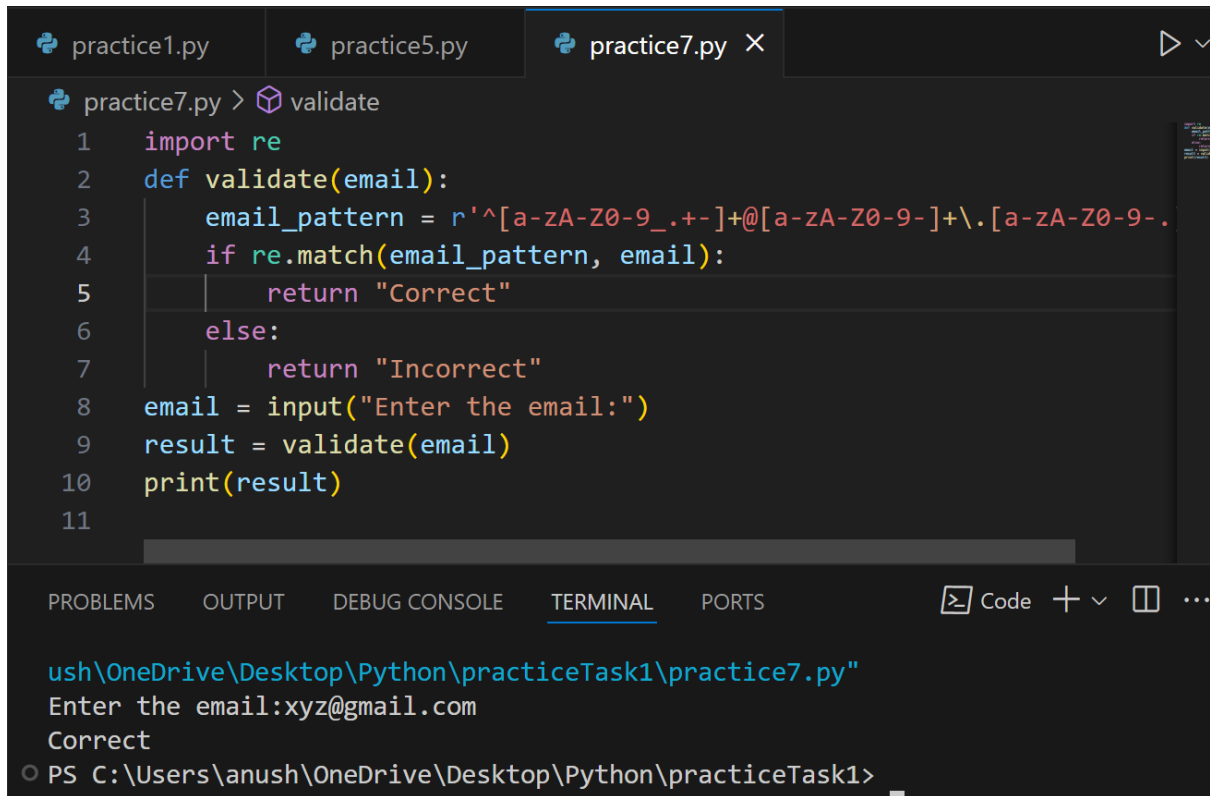
```
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> pyth
● ush\OneDrive\Desktop\Python\practiceTask1\practice5.py"
['odd', 'even', 'even', 'even', 'odd', 'odd', 'even']
○ PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> 
```

Q6 Write a program to extract numbers from the string. Input='hello 1 hi 9 . How are 10' output=[1,9,10]

```
practice1.py  practice5.py  practice6.py X
practice6.py > extract_numbers
1  import re
2  def extract_numbers(s):
3      numbers = re.findall(r'\d+', s)
4      numbers = [int(num) for num in numbers]
5      return numbers
6  input_string = input("Enter the string:")
7  numbers = extract_numbers(input_string)
8  print(numbers)
9

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS
● PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> py
  ush\OneDrive\Desktop\Python\practiceTask1\practice6.py"
  Enter the string:hello 1 hi 9 . How are 10
○ [1, 9, 10]
  PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
○ PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
```

Q7 Write a program to find given string of email is in correct format or not.



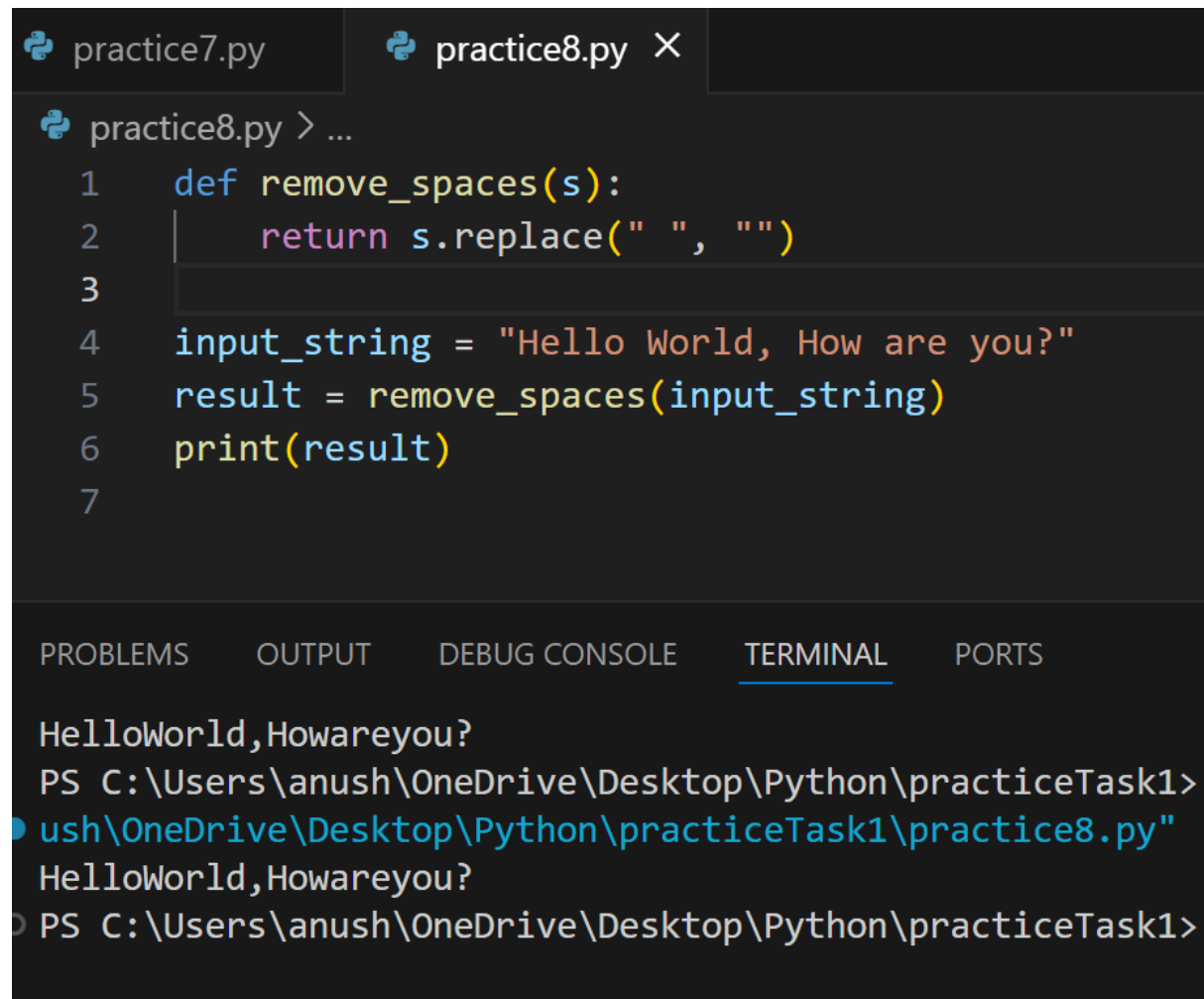
The screenshot shows a Python IDE with three tabs: practice1.py, practice5.py, and practice7.py. The active tab is practice7.py, which contains a Python script to validate an email format. The script defines a function `validate(email)` that uses a regular expression to check if the email is in the correct format. The regular expression is `r'^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+'`. The script prompts the user to enter an email address and prints the result.

```
practice7.py > validate
1  import re
2  def validate(email):
3      email_pattern = r'^[a-zA-Z0-9_+-.]+@[a-zA-Z0-9-]+\.[a-zA-Z0-9-]+'
4      if re.match(email_pattern, email):
5          return "Correct"
6      else:
7          return "Incorrect"
8  email = input("Enter the email:")
9  result = validate(email)
10 print(result)
11
```

The terminal output shows the execution of the program. The user enters the email address `xyz@gmail.com`, and the program outputs `Correct`.

```
ush\OneDrive\Desktop\Python\practiceTask1\practice7.py"
Enter the email:xyz@gmail.com
Correct
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
```

Q8. Write a program to remove all white spaces from the string.



The image shows a screenshot of a Python IDE with two tabs: 'practice7.py' and 'practice8.py'. The 'practice8.py' tab is active, showing a Python script. The script defines a function 'remove\_spaces(s)' that returns 's.replace(" ", "")'. It then sets 'input\_string' to 'Hello World, How are you?', calls 'remove\_spaces(input\_string)', and prints the result. Below the code editor, there are tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is selected, showing the output 'HelloWorld,Howareyou?' and the command prompt path 'PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>'. The command 'python C:\Users\anush\OneDrive\Desktop\Python\practiceTask1\practice8.py' has been executed, resulting in the output 'HelloWorld,Howareyou?'.

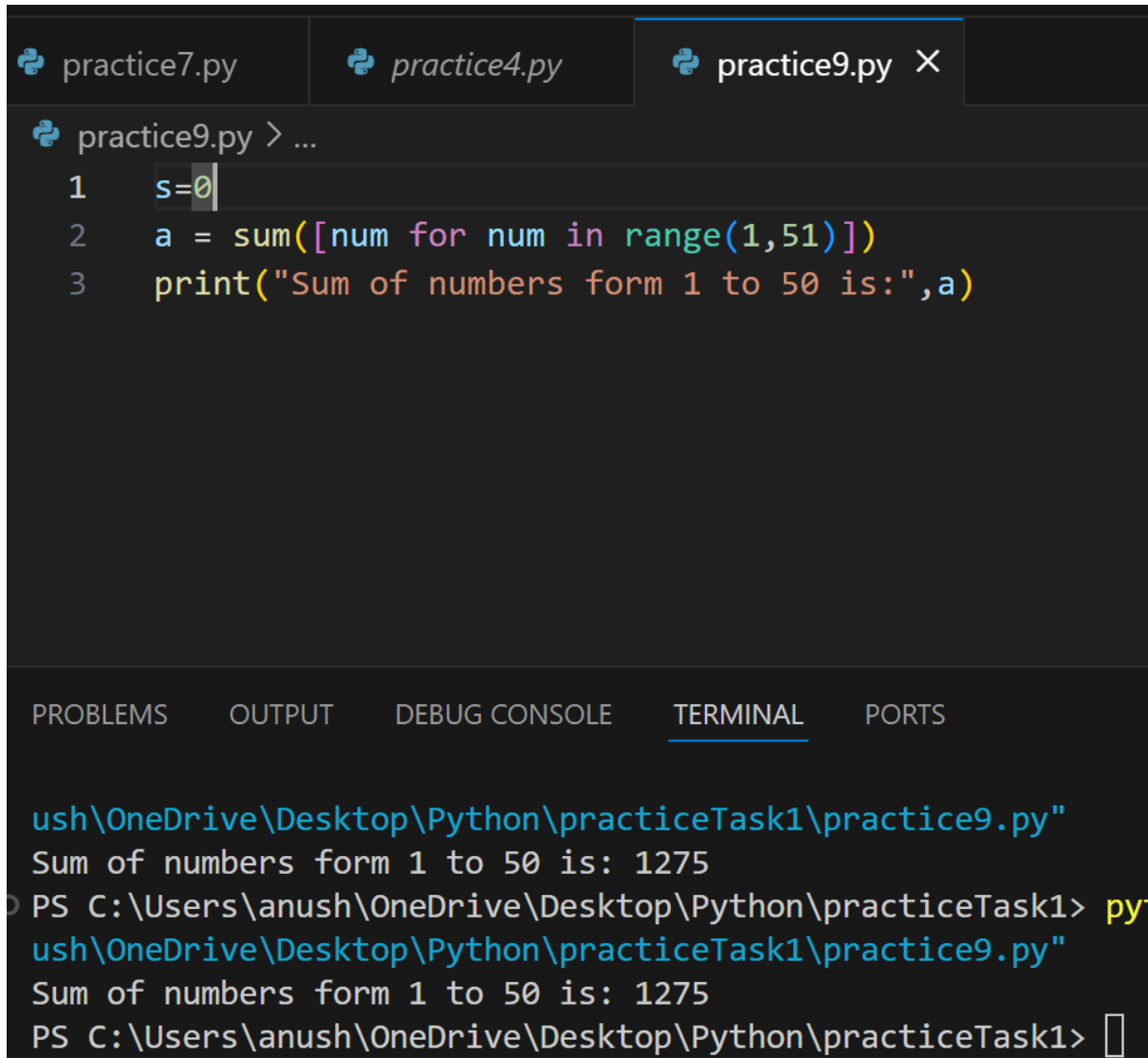
```
practice7.py  practice8.py X
practice8.py > ...
1  def remove_spaces(s):
2      return s.replace(" ", "")
3
4  input_string = "Hello World, How are you?"
5  result = remove_spaces(input_string)
6  print(result)
7

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS

HelloWorld,Howareyou?
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
python C:\Users\anush\OneDrive\Desktop\Python\practiceTask1\practice8.py
HelloWorld,Howareyou?
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
```



Q9 Write a program to find sum of 1 to 50 numbers using list comprehension.



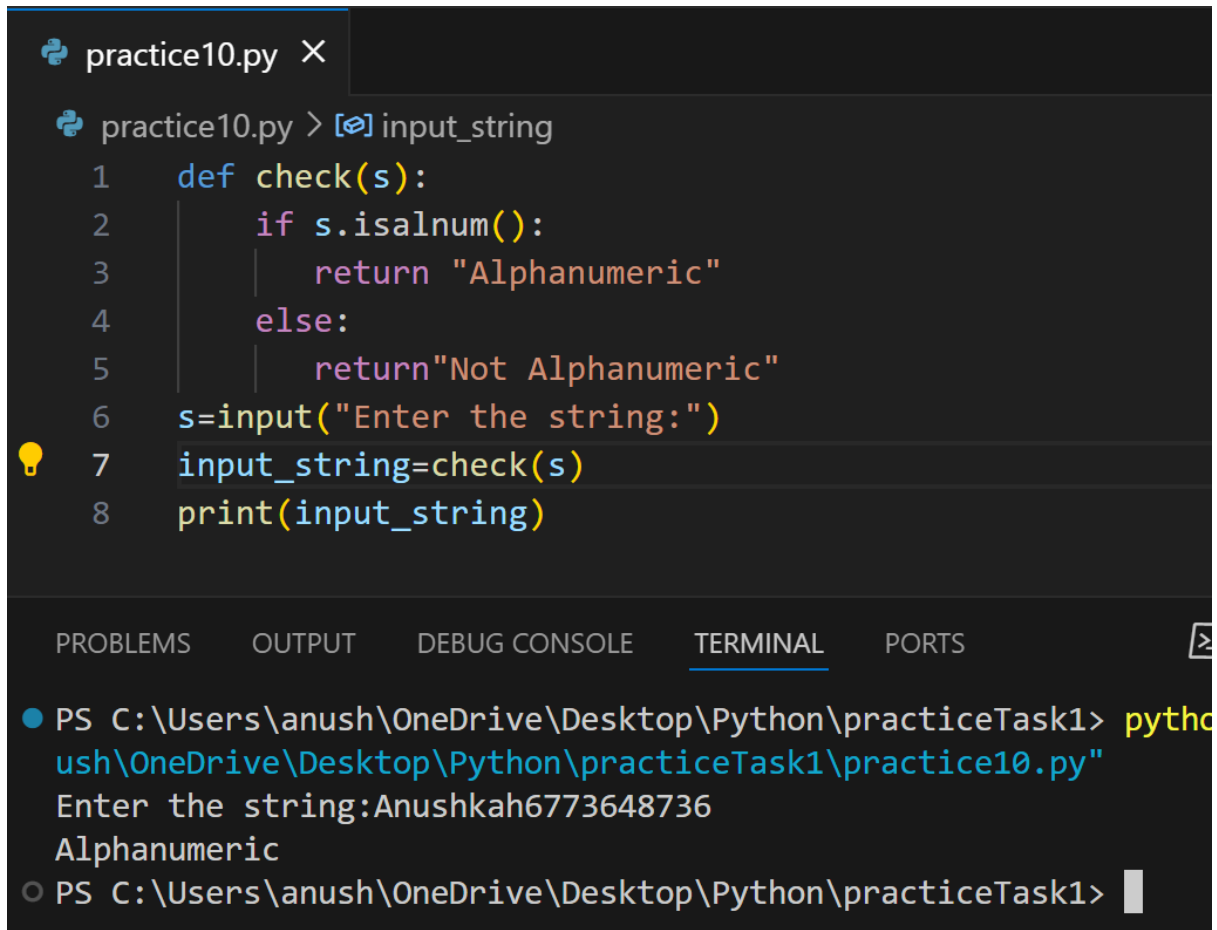
The image shows a code editor with three tabs: `practice7.py`, `practice4.py`, and `practice9.py`. The `practice9.py` tab is active, showing the following code:

```
1 s=0
2 a = sum([num for num in range(1,51)])
3 print("Sum of numbers form 1 to 50 is:",a)
```

Below the code editor is a terminal window with tabs: `PROBLEMS`, `OUTPUT`, `DEBUG CONSOLE`, `TERMINAL` (selected), and `PORTS`. The terminal shows the execution of the program:

```
ush\OneDrive\Desktop\Python\practiceTask1\practice9.py"
Sum of numbers form 1 to 50 is: 1275
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> py
ush\OneDrive\Desktop\Python\practiceTask1\practice9.py"
Sum of numbers form 1 to 50 is: 1275
PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> 
```

Q10. Write a program to check string is alphanumeric or not.



```
practice10.py ×  
practice10.py > [?] input_string  
1 def check(s):  
2     if s.isalnum():  
3         return "Alphanumeric"  
4     else:  
5         return "Not Alphanumeric"  
6 s=input("Enter the string:")  
7 input_string=check(s)  
8 print(input_string)  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
● PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1> python practice10.py  
Enter the string:Anushkah6773648736  
Alphanumeric  
○ PS C:\Users\anush\OneDrive\Desktop\Python\practiceTask1>
```