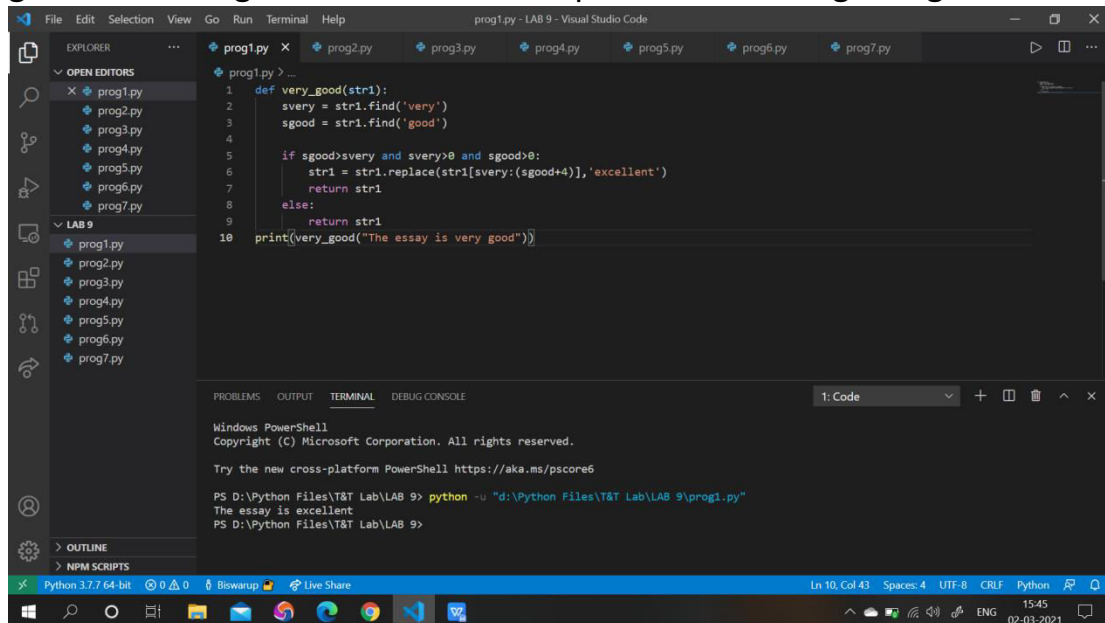


T&T LAB-9

BISWARUP MUKHERJEE

ROLL - 1806468

1. WAPP to find the first appearance of the substring "very" and "good" from a given string "If very follows the good replace the whole "very good" substring with "excellent" and print the resulting string.



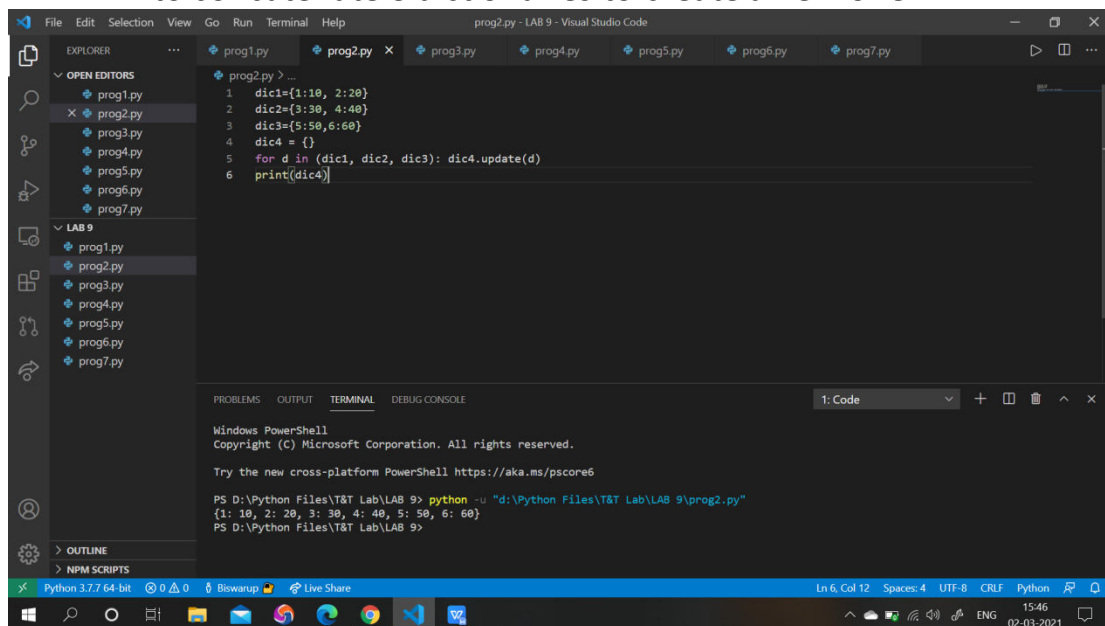
```
1 def very_good(str1):
2     svery = str1.find('very')
3     sgood = str1.find('good')
4
5     if sgood > svery and svery > 0 and sgood > 0:
6         str1 = str1.replace(str1[svery:(sgood+4)], 'excellent')
7         return str1
8     else:
9         return str1
10 print(very_good("The essay is very good"))
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog1.py"
The essay is excellent
PS D:\Python Files\T&T Lab\LAB 9>

2. WAPP to concatenate 3 dictionaries to create a new one.



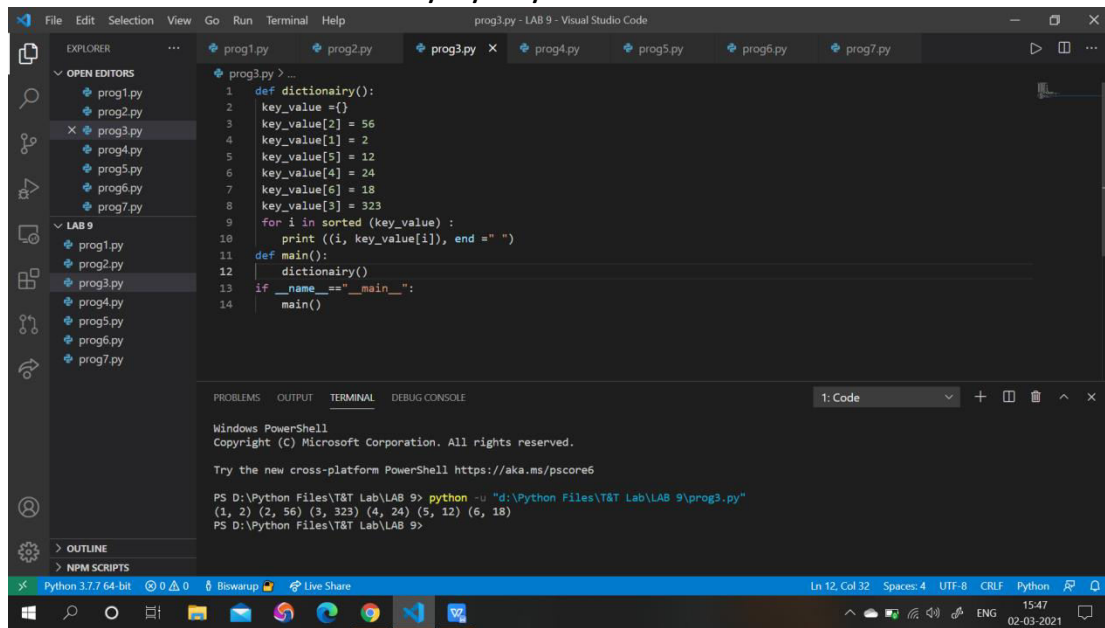
```
1 dic1={1:10, 2:20}
2 dic2={3:30, 4:40}
3 dic3={5:50, 6:60}
4 dic4 = {}
5 for d in (dic1, dic2, dic3): dic4.update(d)
6 print(dic4)
```

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell <https://aka.ms/pscore6>

PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog2.py"
{1: 10, 2: 20, 3: 30, 4: 40, 5: 50, 6: 60}
PS D:\Python Files\T&T Lab\LAB 9>

3. WAPP to sort a dictionary by key value.



The screenshot shows the Visual Studio Code interface with a Python file named `prog3.py` open. The code defines a `dictionary()` function that creates a dictionary `key_value` with the following key-value pairs: (2, 56), (1, 2), (5, 12), (4, 24), (6, 18), (3, 323). It then sorts the dictionary by key and prints the sorted items. The `main()` function calls `dictionary()`. The terminal output shows the execution of the program, resulting in the sorted dictionary: (1, 2) (2, 56) (3, 323) (4, 24) (5, 12) (6, 18).

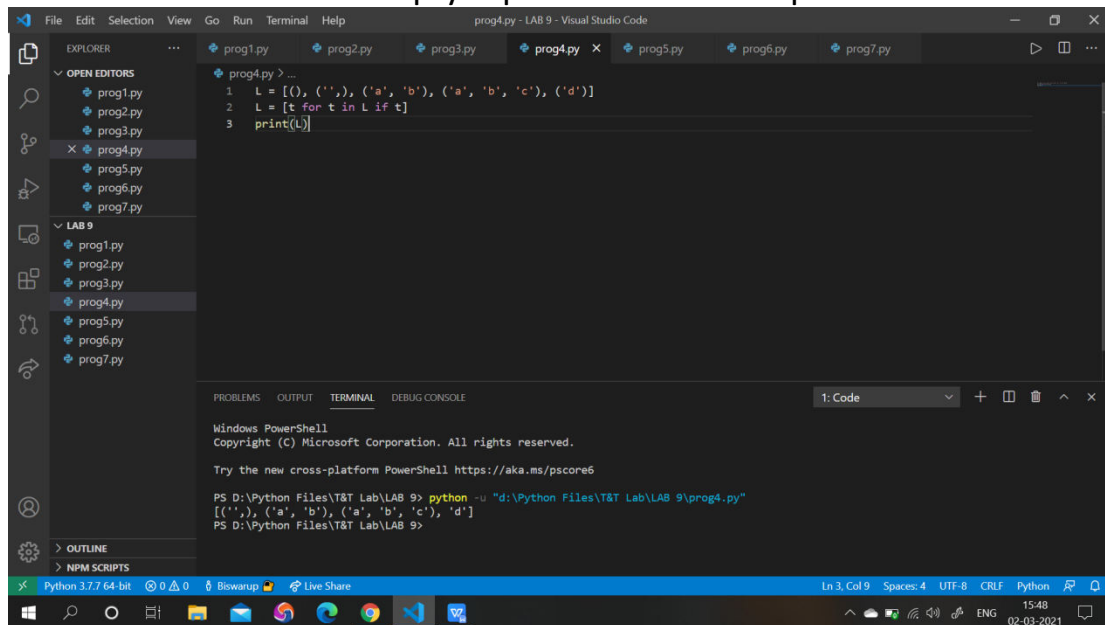
```
1 def dictionary():
2     key_value = {}
3     key_value[2] = 56
4     key_value[1] = 2
5     key_value[5] = 12
6     key_value[4] = 24
7     key_value[6] = 18
8     key_value[3] = 323
9     for i in sorted(key_value):
10        print ((i, key_value[i]), end = " ")
11
12 def main():
13     dictionary()
14     if __name__ == "__main__":
15         main()
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog3.py"
(1, 2) (2, 56) (3, 323) (4, 24) (5, 12) (6, 18)
PS D:\Python Files\T&T Lab\LAB 9>
```

4. WAPP to remove an empty tuple from a list of tuples.



The screenshot shows the Visual Studio Code interface with a Python file named `prog4.py` open. The code defines a list `L` containing tuples: `[(), ('a', 'b'), ('a', 'b', 'c'), ('d')]`. It then creates a new list `L` containing only the non-empty tuples using a list comprehension: `L = [t for t in L if t]`. The `print(L)` statement outputs the filtered list: `[('a', 'b'), ('a', 'b', 'c'), ('d')]`. The terminal output shows the execution of the program, resulting in the filtered list: `[('a', 'b'), ('a', 'b', 'c'), ('d')]`.

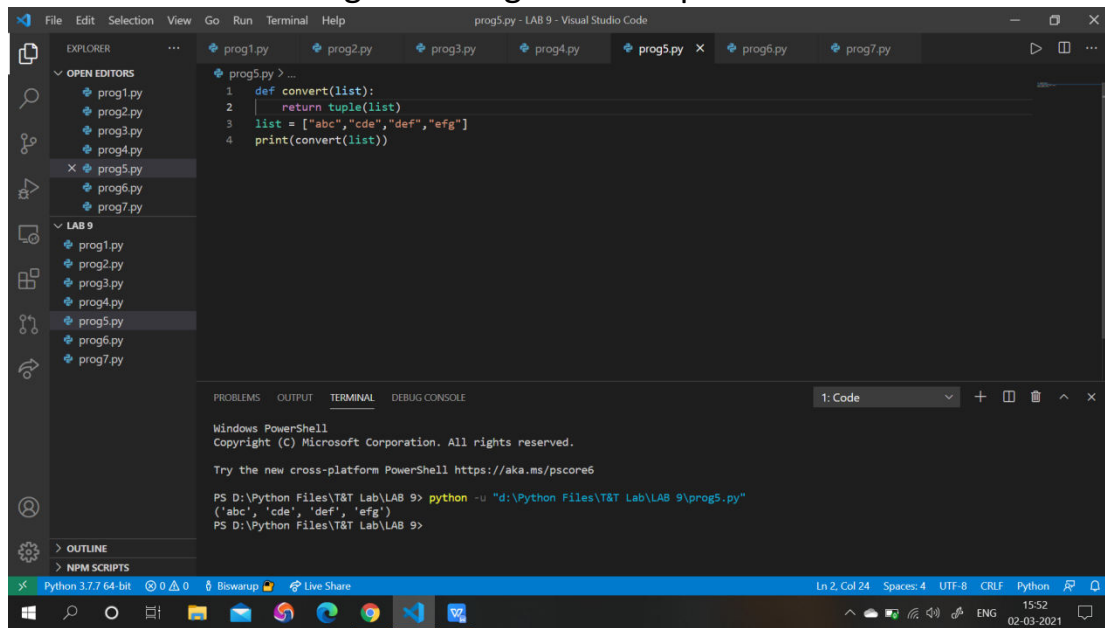
```
1 L = [(), ('a', 'b'), ('a', 'b', 'c'), ('d')]
2 L = [t for t in L if t]
3 print(L)
```

```
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog4.py"
[('a', 'b'), ('a', 'b', 'c'), ('d')]
PS D:\Python Files\T&T Lab\LAB 9>
```

5. WAPP to convert a given string list to a tuple.



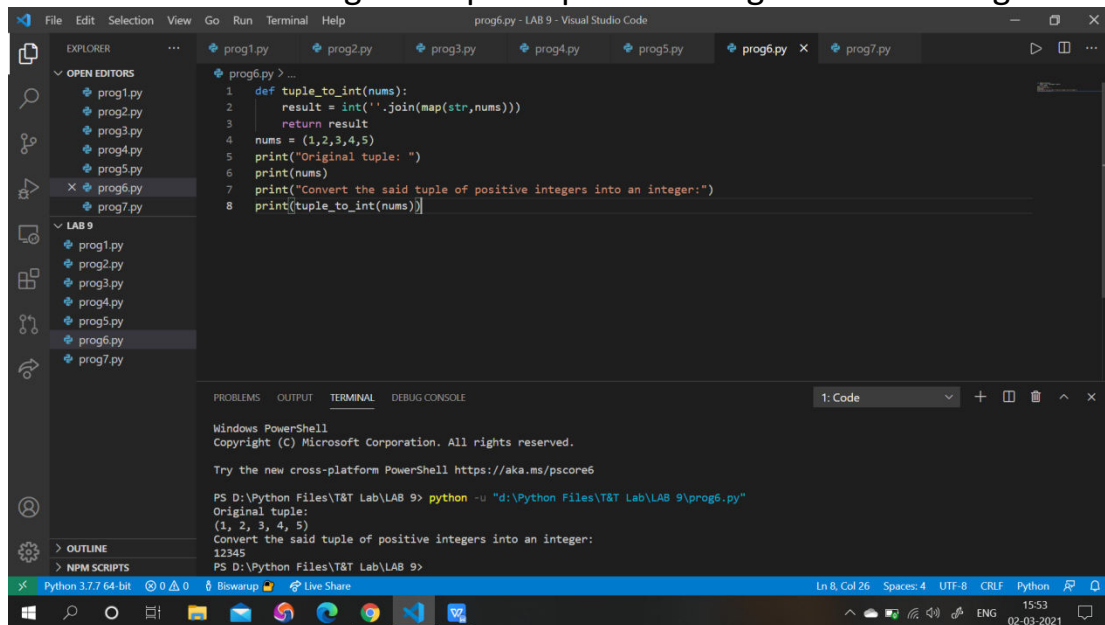
The screenshot shows the Visual Studio Code interface with a file explorer on the left containing files prog1.py through prog7.py. The main editor displays prog5.py with the following code:

```
1 def convert(list):  
2     | return tuple(list)  
3 list = ["abc", "cde", "def", "efg"]  
4 print(convert(list))
```

The bottom terminal window shows the execution of the script:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog5.py"  
( 'abc', 'cde', 'def', 'efg' )  
PS D:\Python Files\T&T Lab\LAB 9>
```

6. WAPP to convert a given tuple of positive integers into an integer.



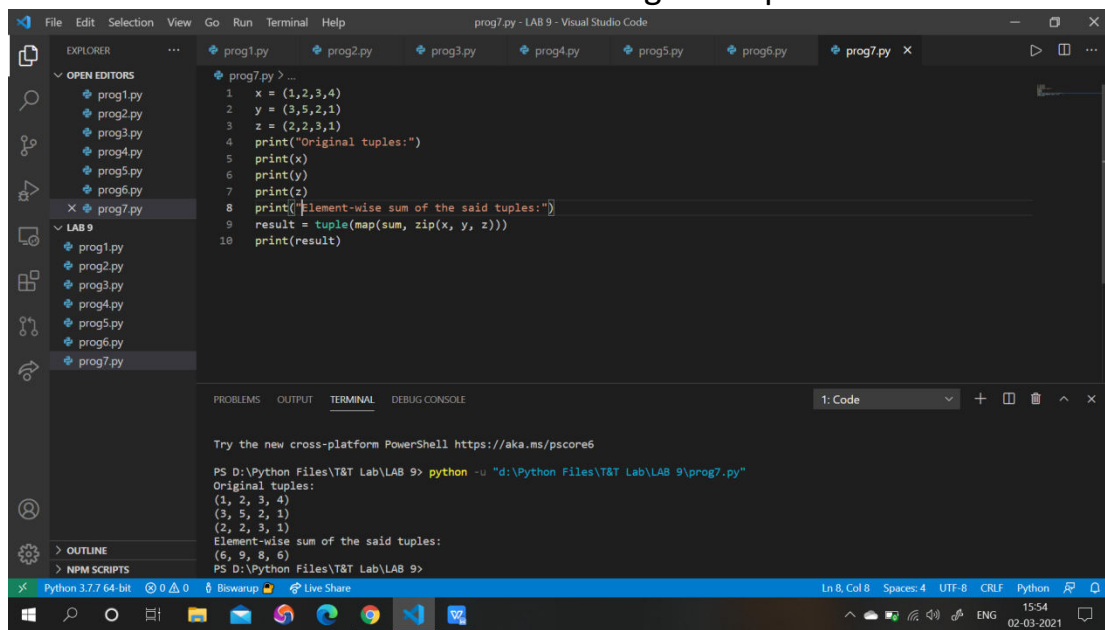
The screenshot shows the Visual Studio Code interface with a file explorer on the left containing files prog1.py through prog7.py. The main editor displays prog6.py with the following code:

```
1 def tuple_to_int(nums):  
2     result = int(''.join(map(str,nums)))  
3     return result  
4 nums = (1,2,3,4,5)  
5 print("Original tuple: ")  
6 print(nums)  
7 print("Convert the said tuple of positive integers into an integer:")  
8 print(tuple_to_int(nums))
```

The bottom terminal window shows the execution of the script:

```
Windows PowerShell  
Copyright (C) Microsoft Corporation. All rights reserved.  
  
Try the new cross-platform PowerShell https://aka.ms/pscore6  
  
PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog6.py"  
Original tuple:  
(1, 2, 3, 4, 5)  
Convert the said tuple of positive integers into an integer:  
12345  
PS D:\Python Files\T&T Lab\LAB 9>
```

7. WAPP to find element wise sum of the given tuples.



The screenshot shows the Visual Studio Code interface with a Python file named `prog7.py` open. The code defines three tuples: `x = (1, 2, 3, 4)`, `y = (3, 5, 2, 1)`, and `z = (2, 2, 3, 1)`. It prints the original tuples and then calculates the element-wise sum using the `map` and `zip` functions. The result is a tuple `(6, 9, 8, 6)`.

```
1 x = (1,2,3,4)
2 y = (3,5,2,1)
3 z = (2,2,3,1)
4 print("Original tuples:")
5 print(x)
6 print(y)
7 print(z)
8 print("Element-wise sum of the said tuples:")
9 result = tuple(map(sum, zip(x, y, z)))
10 print(result)
```

The terminal output shows the execution of the program, displaying the original tuples and the element-wise sum.

```
PS D:\Python Files\T&T Lab\LAB 9> python -u "d:\Python Files\T&T Lab\LAB 9\prog7.py"
Original tuples:
(1, 2, 3, 4)
(3, 5, 2, 1)
(2, 2, 3, 1)
Element-wise sum of the said tuples:
(6, 9, 8, 6)
PS D:\Python Files\T&T Lab\LAB 9>
```