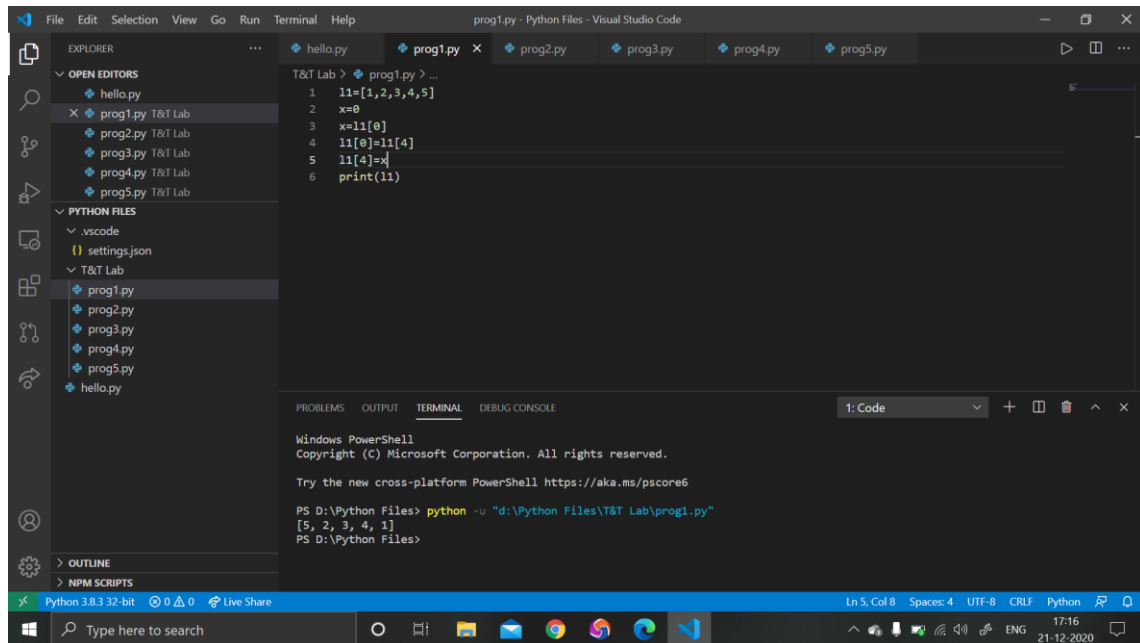


# T&T Lab-1

## BISWARUP MUKHERJEE

### ROLL-1806468

1.WAP to create a list and interchange the 1<sup>st</sup> and last element.



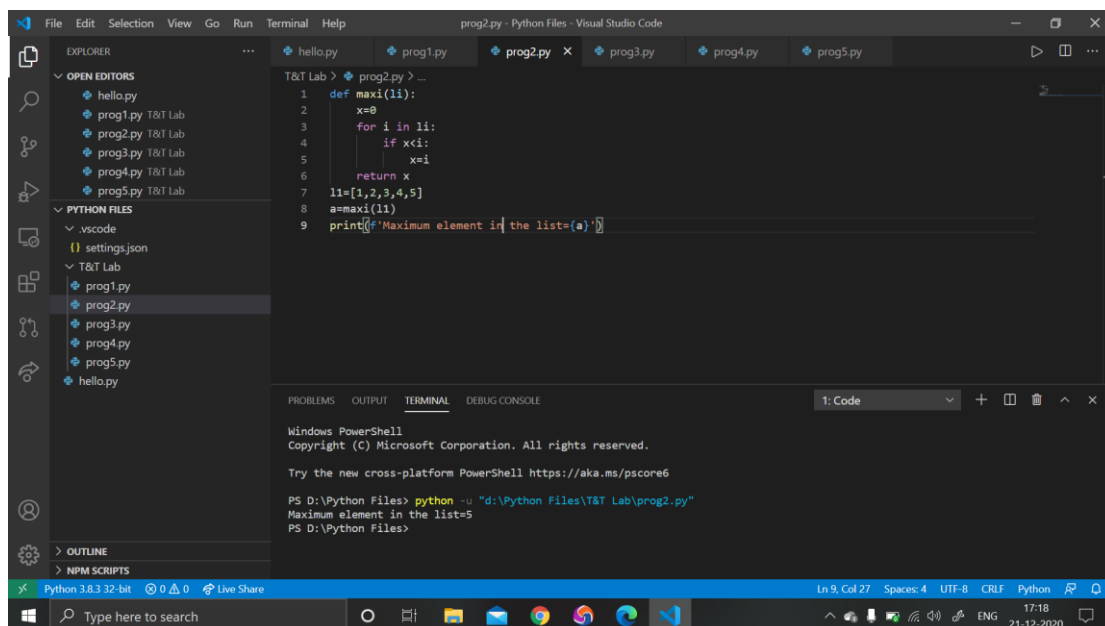
The screenshot shows the Visual Studio Code interface with a Python file named `prog1.py` open. The code in the editor is as follows:

```
1 ll=[1,2,3,4,5]
2 x=0
3 x=ll[0]
4 ll[0]=ll[4]
5 ll[4]=x
6 print(ll)
```

The terminal output shows the execution of the program:

```
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog1.py"
[5, 2, 3, 4, 1]
PS D:\Python Files>
```

2.WAP to find the largest element in the list.



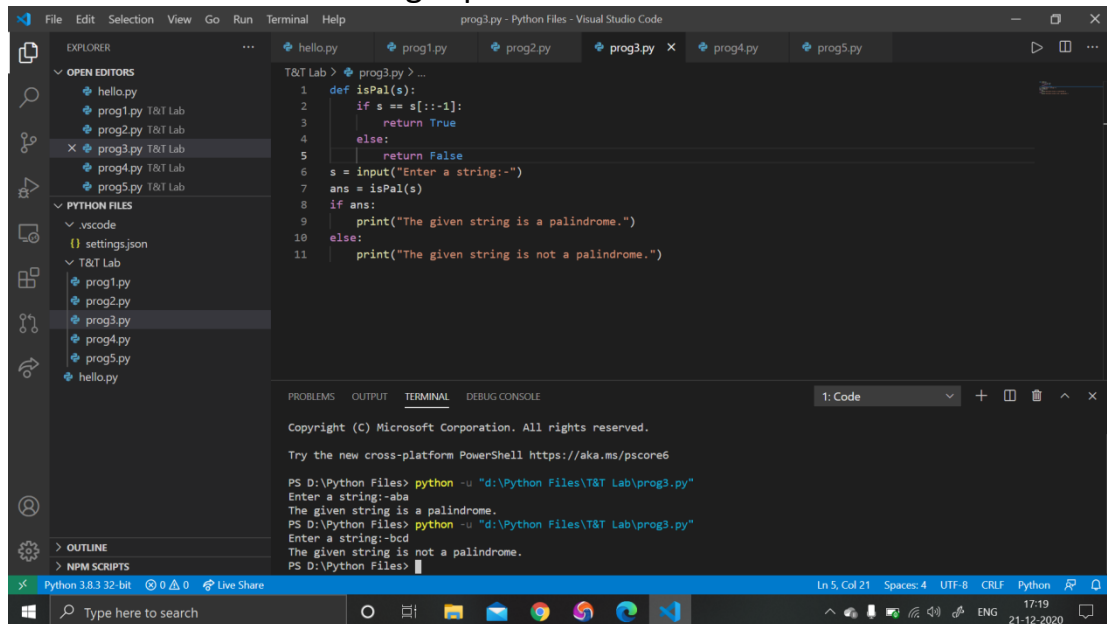
The screenshot shows the Visual Studio Code interface with a Python file named `prog2.py` open. The code in the editor is as follows:

```
1 def maxi(ll):
2     x=0
3     for i in ll:
4         if x<i:
5             x=i
6     return x
7 ll=[1,2,3,4,5]
8 a=maxi(ll)
9 print(f'Maximum element in the list={a}')
```

The terminal output shows the execution of the program:

```
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog2.py"
Maximum element in the list=5
PS D:\Python Files>
```

### 3. WAP to check if a string is palindrome or not.

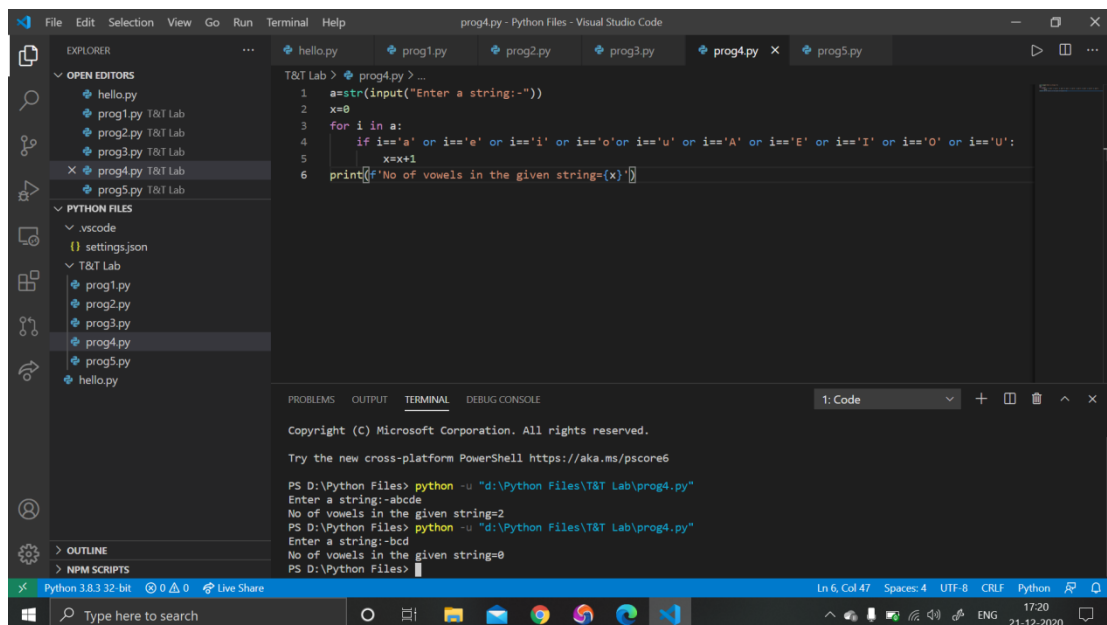


The screenshot shows the Visual Studio Code interface with a Python file named `prog3.py` open. The code defines a function `isPal(s)` that checks if a string is a palindrome. It takes user input and prints the result. The terminal shows the execution of the program with test cases 'aba' and 'bcd'.

```
1 def isPal(s):
2     if s == s[::-1]:
3         return True
4     else:
5         return False
6 s = input("Enter a string:-")
7 ans = isPal(s)
8 if ans:
9     print("The given string is a palindrome.")
10 else:
11     print("The given string is not a palindrome.")
```

```
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog3.py"
Enter a string:-aba
The given string is a palindrome.
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog3.py"
Enter a string:-bcd
The given string is not a palindrome.
PS D:\Python Files>
```

### 4.WAP to check whether string contains vowel or not also count no of vowels if present else print 0.

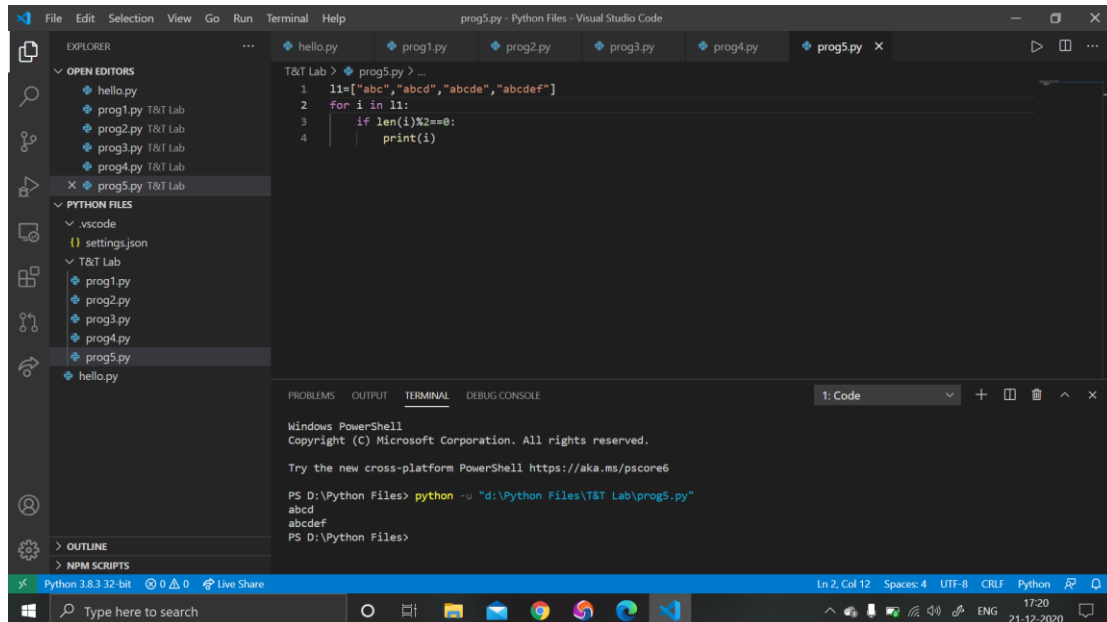


The screenshot shows the Visual Studio Code interface with a Python file named `prog4.py` open. The code takes user input and checks if it contains any vowels. If it does, it prints the count of vowels; otherwise, it prints 0. The terminal shows the execution of the program with test cases 'abcde' and 'bcd'.

```
1 a=string(input("Enter a string:-"))
2 x=0
3 for i in a:
4     if i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or i=='O' or i=='U':
5         x+=1
6 print(f'No of vowels in the given string={x}')
```

```
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog4.py"
Enter a string:-abcde
No of vowels in the given strings=2
PS D:\Python Files> python -u "d:\Python Files\T&T Lab\prog4.py"
Enter a string:-bcd
No of vowels in the given string=0
PS D:\Python Files>
```

5.WAP to create a list and find the even length strings from that list.



The screenshot displays the Visual Studio Code interface with a Python file named `prog5.py` open. The Explorer sidebar on the left shows the file structure, including `hello.py`, `prog1.py`, `prog2.py`, `prog3.py`, `prog4.py`, and `prog5.py`. The main editor area shows the following Python code:

```
1 l1=["abc","abcd","abcde","abcdef"]
2 for i in l1:
3     if len(i)%2==0:
4         print(i)
```

The bottom panel shows the Terminal output, which includes the Windows PowerShell prompt and 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> python -u "d:\Python Files\T&T Lab\prog5.py"
abcd
abcdef
PS D:\Python Files>
```

The status bar at the bottom indicates the Python version (3.8.3 32-bit) and the current file position (Ln 2, Col 12).