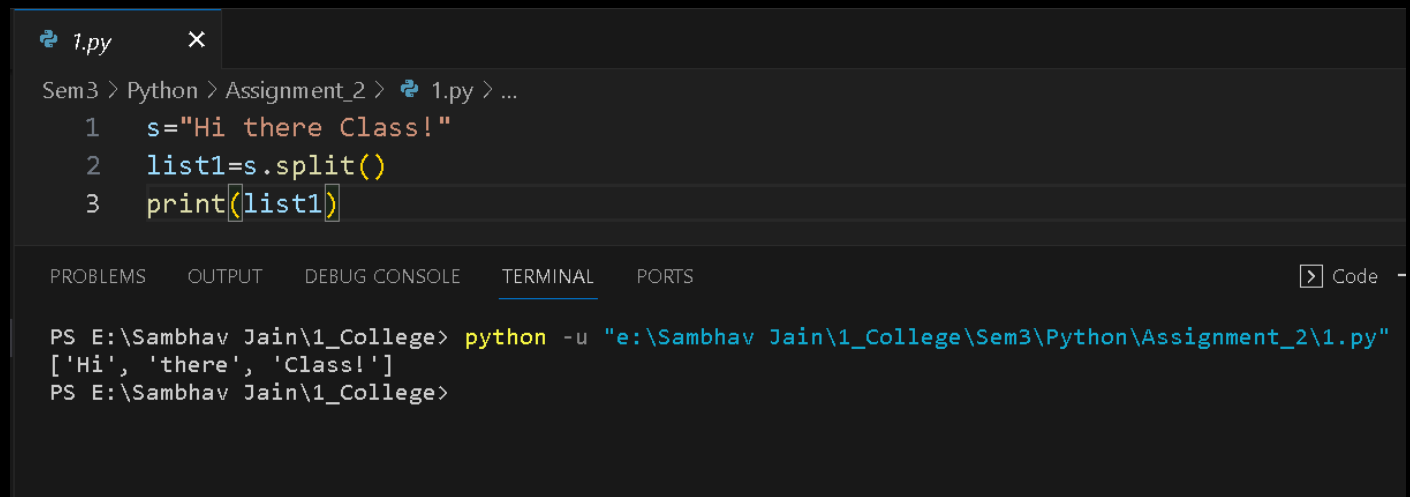


Q1. Split this string
s = "Hi there Class!"
into a list.

Solution Code:

```
s="Hi there Class!"  
list1=s.split()  
print(list1)
```

Output:



The screenshot shows a code editor with a file named `1.py` containing the following Python code:

```
1 s="Hi there Class!"  
2 list1=s.split()  
3 print(list1)
```

Below the code editor is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command `python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\1.py"` being executed in a PowerShell prompt. The output of the command is `['Hi', 'there', 'Class!']`.

Q2. Given the variables

```
planet = "Earth"
```

```
diameter = 12742
```

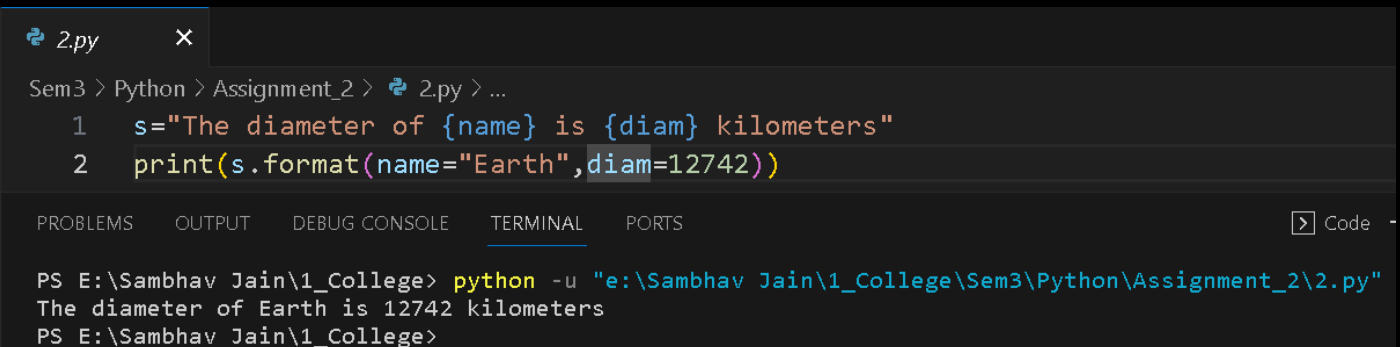
Use `.format()` to print the following string:

The diameter of Earth is 12742 kilometers.

Solution Code:

```
s="The diameter of {name} is {diam} kilometers"  
print(s.format(name="Earth",diam=12742))
```

Output:



The screenshot shows a code editor with a file named `2.py`. The code contains two lines: `s="The diameter of {name} is {diam} kilometers"` and `print(s.format(name="Earth",diam=12742))`. Below the code editor is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The terminal shows the command `python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\2.py"` being executed, followed by the output `The diameter of Earth is 12742 kilometers`.

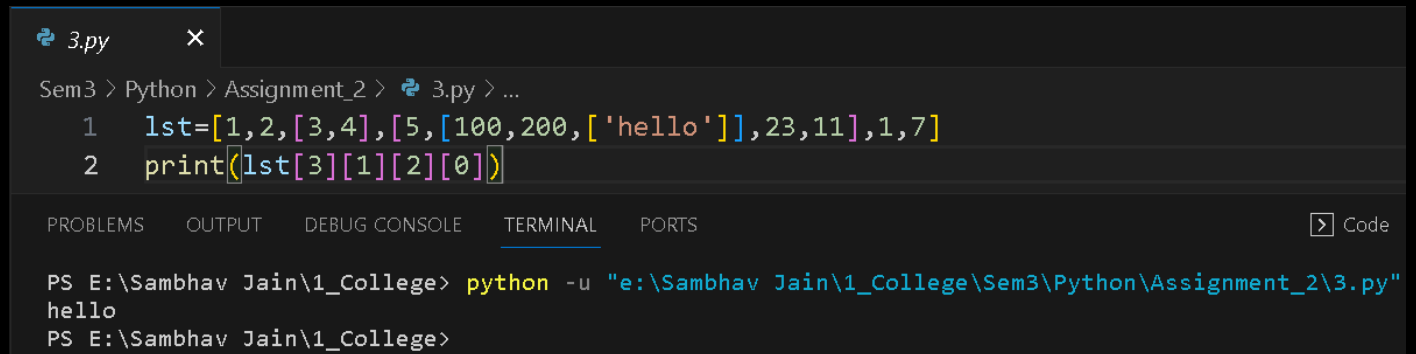
```
2.py x  
Sem3 > Python > Assignment_2 > 2.py > ...  
1 s="The diameter of {name} is {diam} kilometers"  
2 print(s.format(name="Earth",diam=12742))  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS > Code  
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\2.py"  
The diameter of Earth is 12742 kilometers  
PS E:\Sambhav Jain\1_College>
```

Q3. Given this nested list, use indexing to grab the word "hello"
lst = [1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]

Solution Code:

```
lst=[1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
print(lst[3][1][2][0])
```

Output:



The screenshot shows a code editor with a file named `3.py`. The code contains two lines: `lst=[1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]` and `print(lst[3][1][2][0])`. Below the code editor is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The terminal shows the command `python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\3.py"` being executed, resulting in the output `hello`.

```
3.py x
Sem3 > Python > Assignment_2 > 3.py > ...
1  lst=[1,2,[3,4],[5,[100,200,['hello']],23,11],1,7]
2  print(lst[3][1][2][0])

PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  PORTS  Code

PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\3.py"
hello
PS E:\Sambhav Jain\1_College>
```

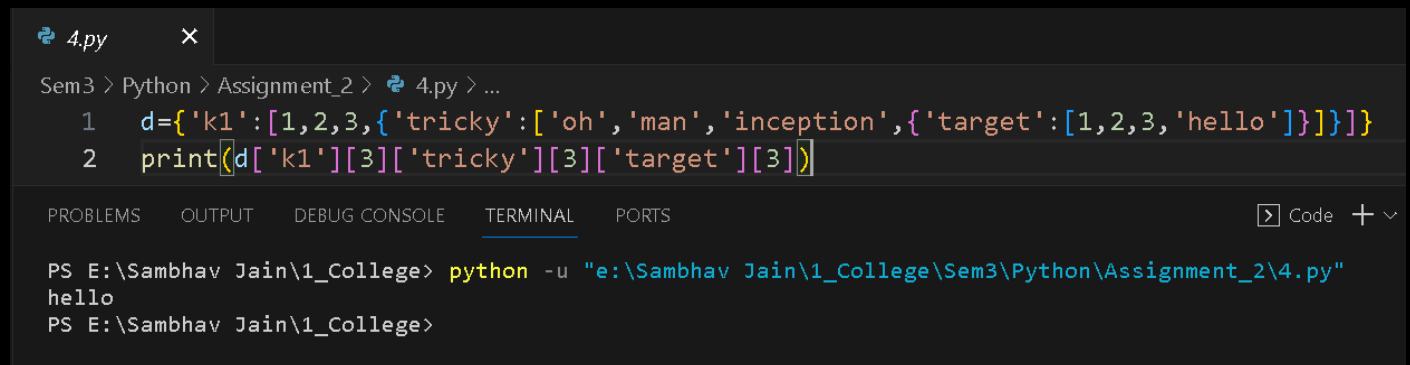
Q4. Given this nested dictionary grab the word "hello".

```
d =  
{'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}
```

Solution Code:

```
d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}  
print(d['k1'][3]['tricky'][3]['target'][3])
```

Output:



The screenshot shows a Python IDE window titled '4.py'. The code editor contains the following code:

```
1 d={'k1':[1,2,3,{'tricky':['oh','man','inception',{'target':[1,2,3,'hello']}]}]}  
2 print(d['k1'][3]['tricky'][3]['target'][3])
```

The IDE has tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active, showing the command prompt output:

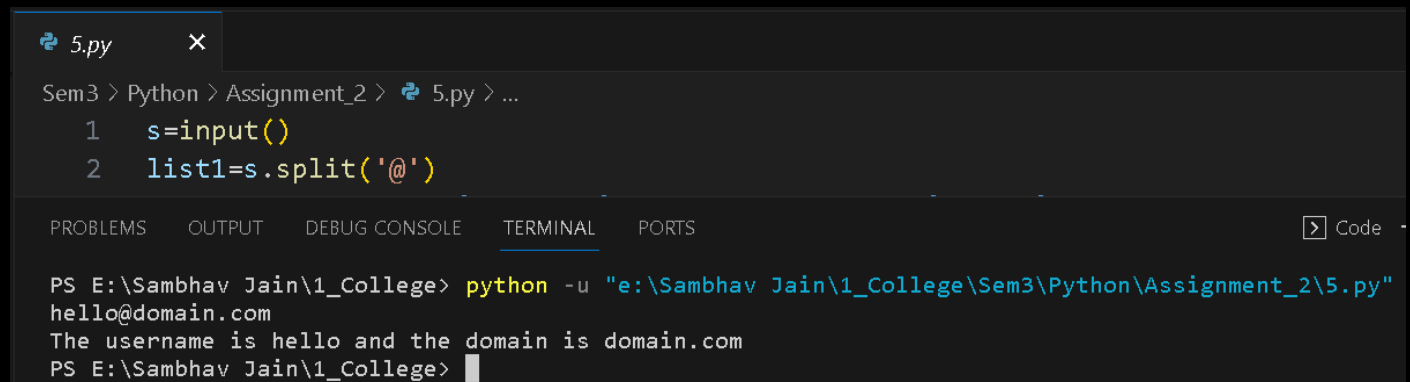
```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\4.py"  
hello  
PS E:\Sambhav Jain\1_College>
```

Q5. Create a function that grabs the email website domain from a string in the form:
user@domain.com

Solution Code:

```
s=input()
list1=s.split('@')
s="The username is {username} and the domain is {domain}"
print(s.format(username=list1[0],domain=list1[1]))
```

Output:



The screenshot shows a code editor with a file named `5.py` containing the following Python code:

```
1 s=input()
2 list1=s.split('@')
```

Below the editor is a terminal window with tabs for PROBLEMS, OUTPUT, DEBUG CONSOLE, TERMINAL, and PORTS. The TERMINAL tab is active, showing the command `python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\5.py"` being executed. The output of the script is displayed in the terminal:

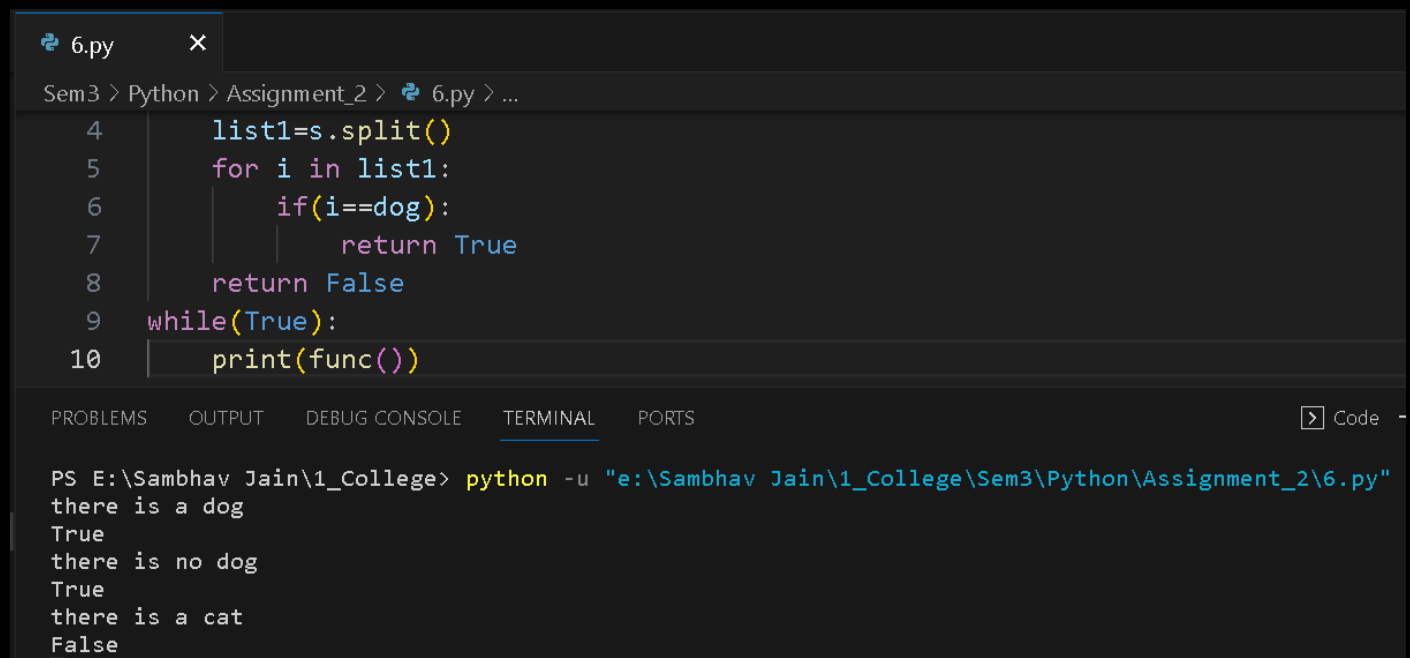
```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\5.py"
hello@domain.com
The username is hello and the domain is domain.com
PS E:\Sambhav Jain\1_College>
```

Q6. Create a basic function that returns True if the word 'dog' is contained in the input string.

Solution Code:

```
def func():  
    s=input()  
    dog='dog'  
    list1=s.split()  
    for i in list1:  
        if(i==dog):  
            return True  
    return False  
while(True):  
    print(func())
```

Output:



The screenshot shows a code editor with a file named '6.py' open. The code in the editor is as follows:

```
4 list1=s.split()  
5 for i in list1:  
6     if(i==dog):  
7         return True  
8 return False  
9 while(True):  
10    print(func())
```

Below the code editor is a terminal window with the following output:

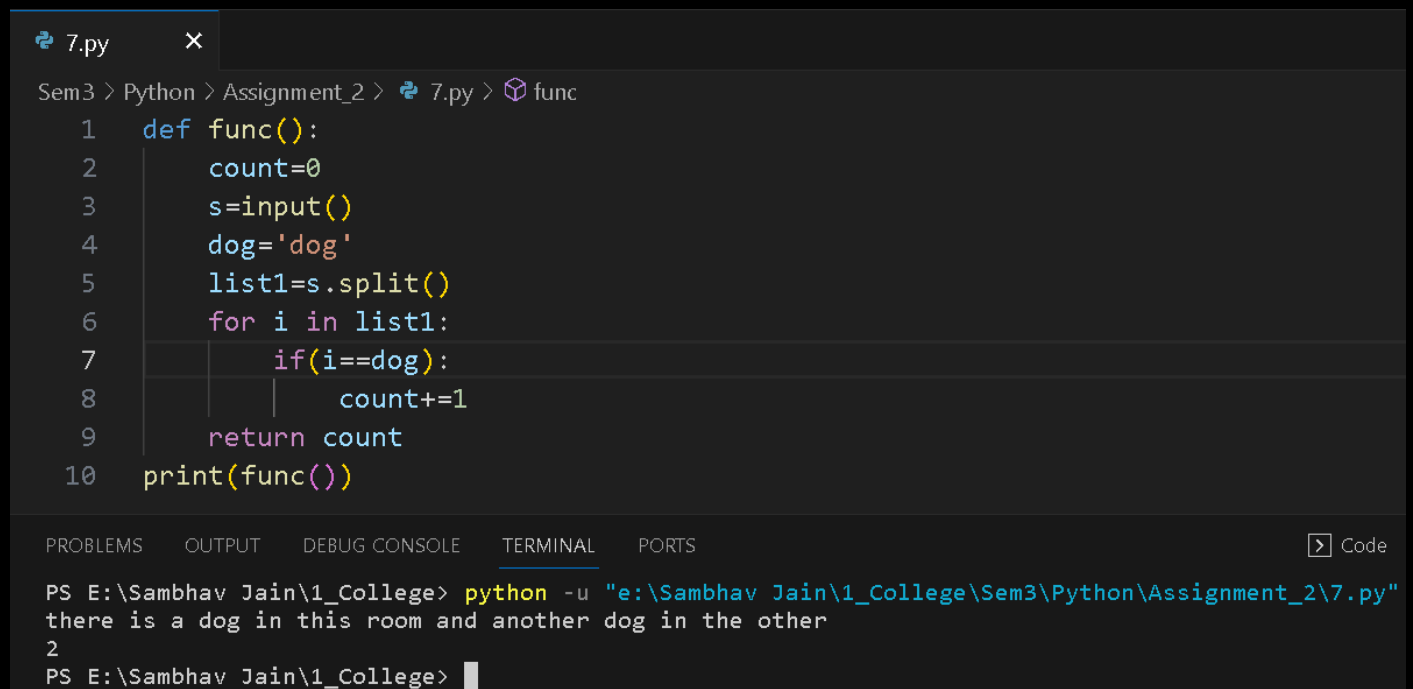
```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\6.py"  
there is a dog  
True  
there is no dog  
True  
there is a cat  
False
```

Q7. Create a function that counts the number of times the word "dog" occurs in a string.

Solution Code:

```
def func():  
    count=0  
    s=input()  
    dog='dog'  
    list1=s.split()  
    for i in list1:  
        if(i==dog):  
            count+=1  
    return count  
print(func())
```

Output:



The screenshot shows a code editor with a file named 7.py. The code defines a function func() that counts the occurrences of the word 'dog' in a string. The function takes user input, splits it into a list, and iterates through the list to count the word 'dog'. The output of the program is displayed in the terminal window below the code editor.

```
7.py x  
Sem3 > Python > Assignment_2 > 7.py > func  
1 def func():  
2     count=0  
3     s=input()  
4     dog='dog'  
5     list1=s.split()  
6     for i in list1:  
7         if(i==dog):  
8             count+=1  
9     return count  
10 print(func())  
  
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS  
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\7.py"  
there is a dog in this room and another dog in the other  
2  
PS E:\Sambhav Jain\1_College>
```

Q8. Use lambda expressions and the filter() function to filter out words from a list that don't start with the letter 's'.
For example: seq = ['soup','dog','salad','cat','great']

Solution Code:

```
seq = ['soup','dog','salad','cat','great']  
ans=list(filter(lambda it:it[0]=="s",seq))  
print(ans)
```

Output:

The screenshot shows a code editor with two tabs: '7.py' and '8.py'. The '8.py' tab is active and contains the following code:

```
1 seq = ['soup','dog','salad','cat','great']  
2 ans=list(filter(lambda it:it[0]=="s",seq))  
3 print(ans)
```

Below the code editor is a terminal window with tabs for 'PROBLEMS', 'OUTPUT', 'DEBUG CONSOLE', 'TERMINAL', and 'PORTS'. The 'TERMINAL' tab is active and shows the command to run the script and its output:

```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\8.py"  
['soup', 'salad']  
PS E:\Sambhav Jain\1_College>
```


Q9. You are driving a little too fast, and a police officer stops you. Write a function to return one of 3 possible results: "No Challan", "Small Challan", or "Heavy Challan". If your speed is 60 or less, the result is "No Challan". If speed is between 61 and 80 inclusive, the result is "Small Challan". If speed is 81 or more, the result is "Heavy Challan". Unless it is your birthday (encoded as a Boolean value in the parameters of the function) -- on your birthday, your speed can be 5 higher in all cases.

caught_speeding(81,True)

caught_speeding(81,False)

Solution Code:

```
def caught_speeding(n,t):
    add=[0,5][t]
    if(n<=60+add):
        return "No Challan"
    if(n<=80+add):
        return "Small Challan"
    return "Heavy Challan"
while(True):
    n=int(input("Enter speed\n"))
    t=int(input("Is it your birthday(0/1)\n"))
    print(caught_speeding(n,t)+"\n")
```

Output:

```
7.py 9.py x
Sem3 > Python > Assignment_2 > 9.py > ...
1 def caught_speeding(n,t):
  ...

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code +

PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\9.py"
Enter speed
81
Is it your birthday(0/1)
1
Small Challan

Enter speed
81
Is it your birthday(0/1)
0
Heavy Challan

Enter speed
86
Is it your birthday(0/1)
1
Heavy Challan

Enter speed
86
Is it your birthday(0/1)
0
Heavy Challan

Enter speed
60
Is it your birthday(0/1)
0
No Challan
```

Q10. Concatenate two lists index-wise

```
list1 = ["M", "na", "i", "She"]
```

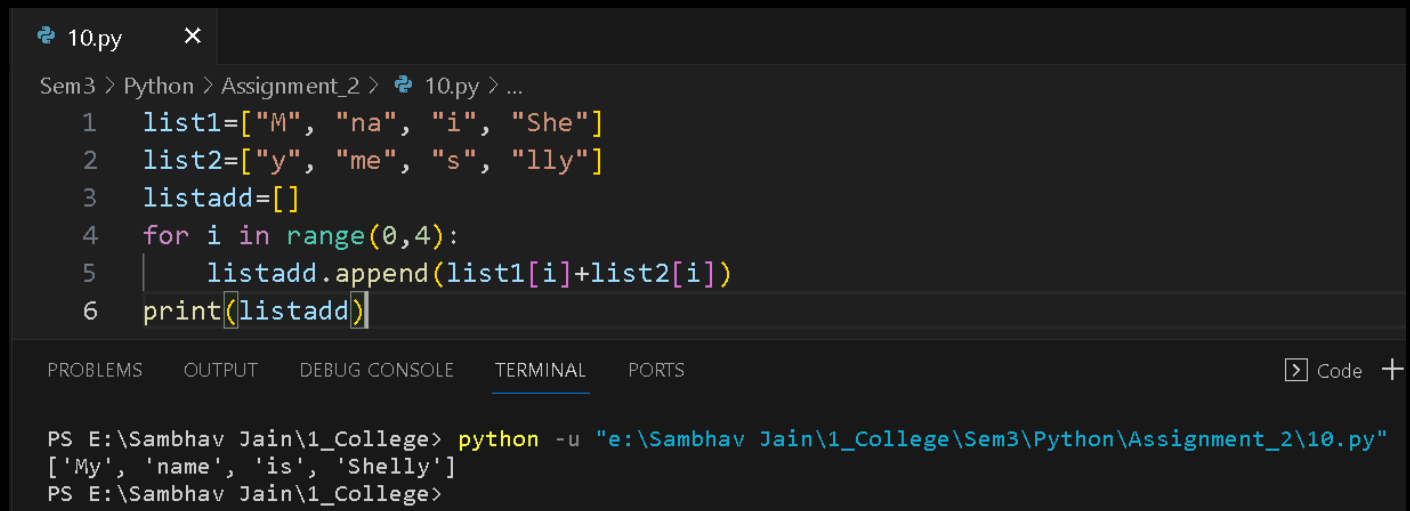
```
list2 = ["y", "me", "s", "lly"]
```

Expected Outcome: ['My', 'name', 'is', 'Shelly']

Solution Code:

```
list1=["M", "na", "i", "She"]
list2=["y", "me", "s", "lly"]
listadd=[]
for i in range(0,4):
    listadd.append(list1[i]+list2[i])
print(listadd)
```

Output:



The screenshot shows a code editor with a file named 10.py. The code is as follows:

```
1 list1=["M", "na", "i", "She"]
2 list2=["y", "me", "s", "lly"]
3 listadd=[]
4 for i in range(0,4):
5     listadd.append(list1[i]+list2[i])
6 print(listadd)
```

Below the code editor, there is a terminal window showing the execution of the code. The terminal output is:

```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\10.py"
['My', 'name', 'is', 'Shelly']
PS E:\Sambhav Jain\1_College>
```

Q11. Concatenate two lists in the following order

```
list1 = ["Hello ", "take "]
```

```
list2 = ["Dear", "Sir"]
```

Expected Output: ['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']

Solution Code:

```
list1 = ["Hello ", "take "]
list2 = ["Dear", "Sir"]
listcomb=[]
for i in list1:
    for j in list2:
        listcomb.append(i+j)
print(listcomb)
```

Output:



The screenshot shows a code editor with a file named 11.py. The code is as follows:

```
1 list1 = ["Hello ", "take "]
2 list2 = ["Dear", "Sir"]
3 listcomb=[]
4 for i in list1:
5     for j in list2:
6         listcomb.append(i+j)
7 print(listcomb)
```

The terminal output shows the command to run the script and the resulting list:

```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\11.py"
['Hello Dear', 'Hello Sir', 'take Dear', 'take Sir']
PS E:\Sambhav Jain\1_College>
```

Q12. Add item 7000 after 6000 in the following Python List
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]

Solution Code:

```
list1 = [10, 20, [300, 400, [5000, 6000], 500], 30, 40]
list1[2][2].append(7000)
print(list1)
```

Output:



The screenshot shows a Python IDE with a file named 8.py. The code in the editor is:

```
1 seq = ['soup', 'dog', 'salad', 'cat', 'great']
2 ans=list(filter(lambda it:it[0]=="s",seq))
3 print(ans)
```

The terminal output shows the execution of a Python command to run a script named 12.py, which produces the following output:

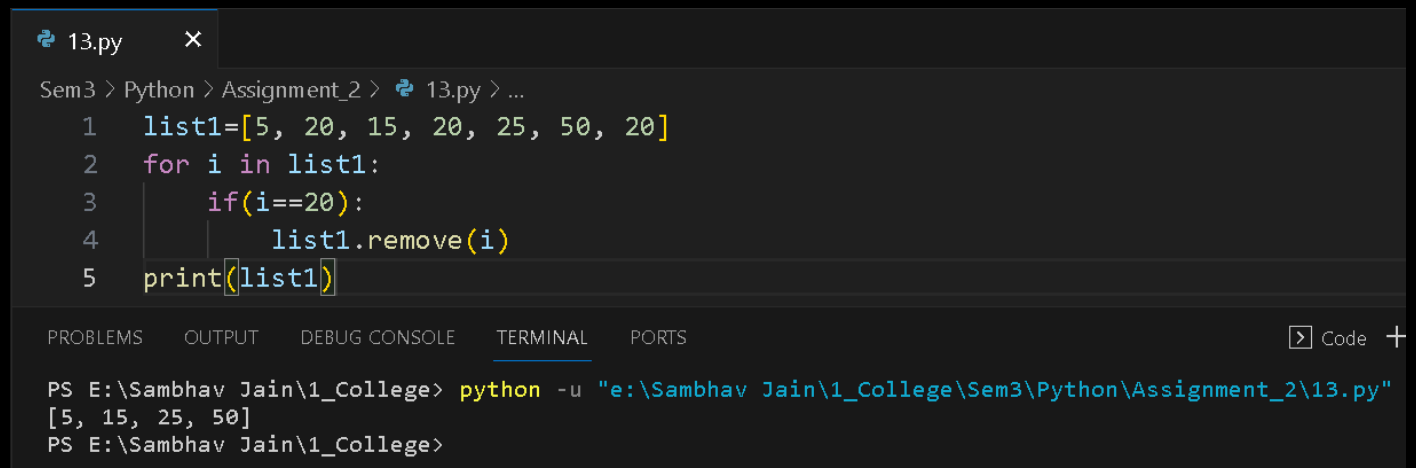
```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\12.py"
[10, 20, [300, 400, [5000, 6000, 7000], 500], 30, 40]
PS E:\Sambhav Jain\1_College>
```

Q13. Given a Python list, remove all occurrence of 20 from the list
list1 = [5, 20, 15, 20, 25, 50, 20]

Solution Code:

```
list1=[5, 20, 15, 20, 25, 50, 20]
for i in list1:
    if(i==20):
        list1.remove(i)
print(list1)
```

Output:



The screenshot shows a code editor with a file named 13.py. The code in the editor is as follows:

```
1 list1=[5, 20, 15, 20, 25, 50, 20]
2 for i in list1:
3     if(i==20):
4         list1.remove(i)
5 print(list1)
```

Below the code editor is a terminal window. The terminal shows the command to run the script and its output:

```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\13.py"
[5, 15, 25, 50]
PS E:\Sambhav Jain\1_College>
```

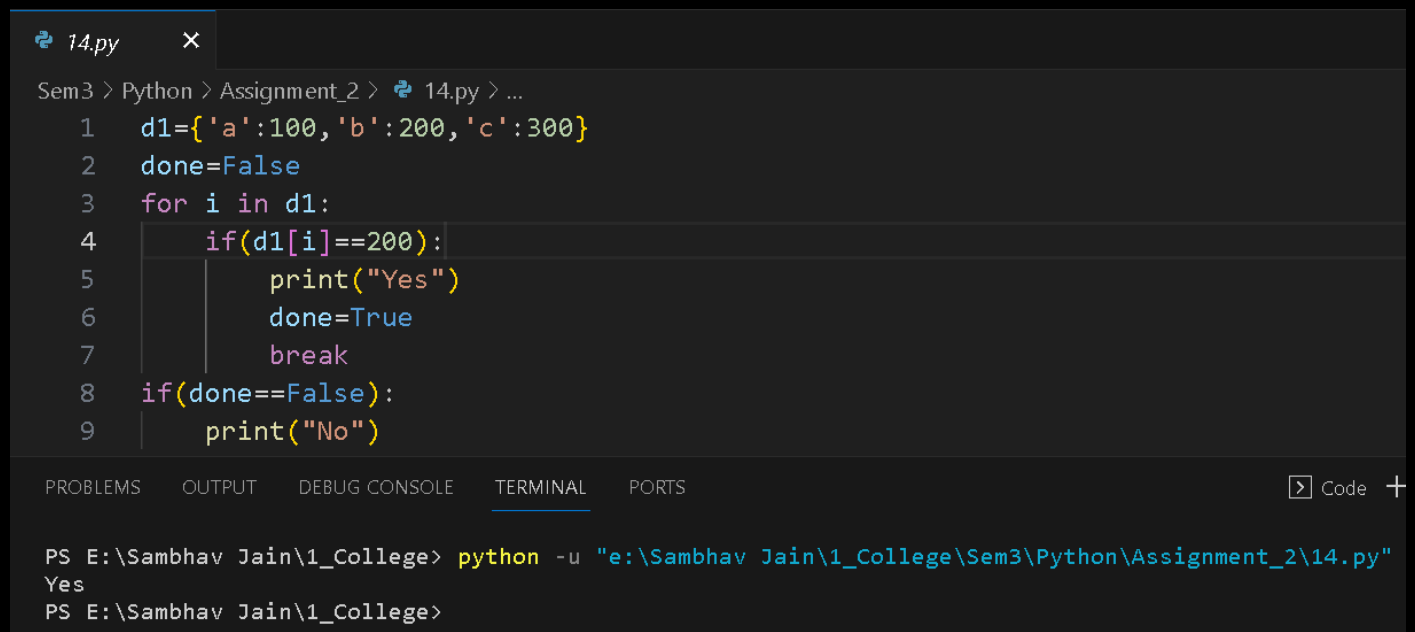
Q14. Check if a value 200 exists in a dictionary

```
d1 = {'a': 100, 'b': 200, 'c': 300}
```

Solution Code:

```
d1={'a':100,'b':200,'c':300}
done=False
for i in d1:
    if(d1[i]==200):
        print("Yes")
        done=True
        break
if(done==False):
    print("No")
```

Output:



The screenshot shows a code editor window titled '14.py' with the following Python code:

```
1 d1={'a':100,'b':200,'c':300}
2 done=False
3 for i in d1:
4     if(d1[i]==200):
5         print("Yes")
6         done=True
7         break
8 if(done==False):
9     print("No")
```

Below the code editor is a terminal window with the following output:

```
PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\14.py"
Yes
PS E:\Sambhav Jain\1_College>
```

Q15. Find the sum of the series $2 + 22 + 222 + 2222 + \dots$ n terms

Solution Code:

```
def func(n):
    print(int(2*(-n+10*(10**n-1)/9)/9))
    n-=1
    start=2
    sum=0
    while(n):
        sum+=start
        start=start*10+2
        n-=1
    sum+=start
    print(sum)
n=int(input("Enter number upto which you want to calculate sum of series"))
for i in range(1,n+1):
    func(i)
# 2+22+222...=2/9(10-1 +100-1+1000-i)
#2/9(-n+)
# s*10=s-10+pow(10,n+1)
# 9*s=10**(n+1)-10
#2*(-n+(10**(n+1)-10)/9)/9
```


Output:

```
15.py x
Sem3 > Python > Assignment_2 > 15.py > func
1 def func(n):
2     print(int(2*(-n+10*(10**n-1)/9)/9))
3     n-=1
4     start=2
5     sum=0
6     while(n):
-

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS Code +

PS E:\Sambhav Jain\1_College> python -u "e:\Sambhav Jain\1_College\Sem3\Python\Assignment_2\15.py"
Enter number upto which you want to calculate sum of series10
2
2
24
24
246
246
2468
2468
24690
24690
246912
246912
2469134
2469134
24691356
24691356
246913578
246913578
2469135800
2469135800
PS E:\Sambhav Jain\1_College>
```