

Q26. What is a string? How can we declare string in Python?

String is a Text datatype . This datatype hold the some text value it represent as str(string)

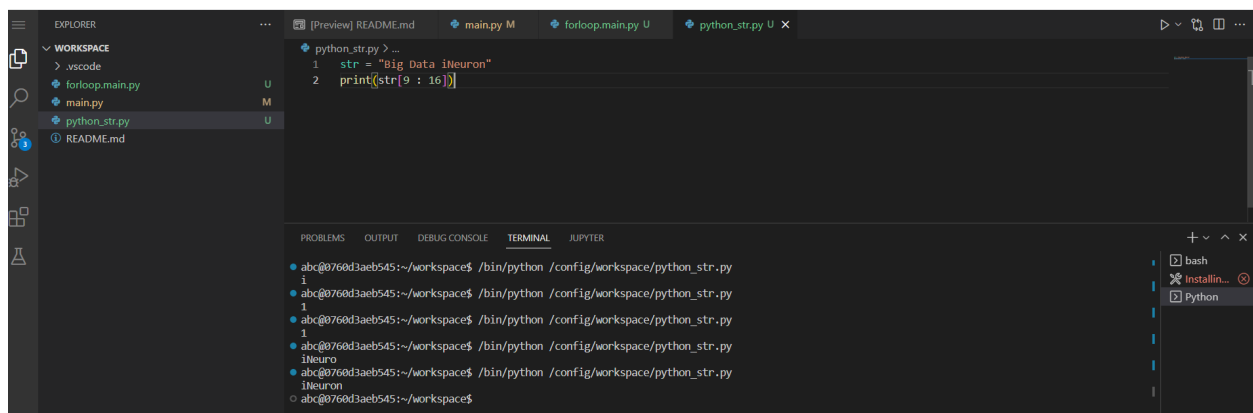
```
Str1 = "STRING"
```

Q27. How can we access the string using its index?

We can access through the square brackets by segregating each character by mentioning the index value . The first character represent the 0 value and last value represent the -1 .

Q28. Write a code to get the desired output of the following

```
string = "Big Data iNeuron"  
desired_output = "iNeuron"
```



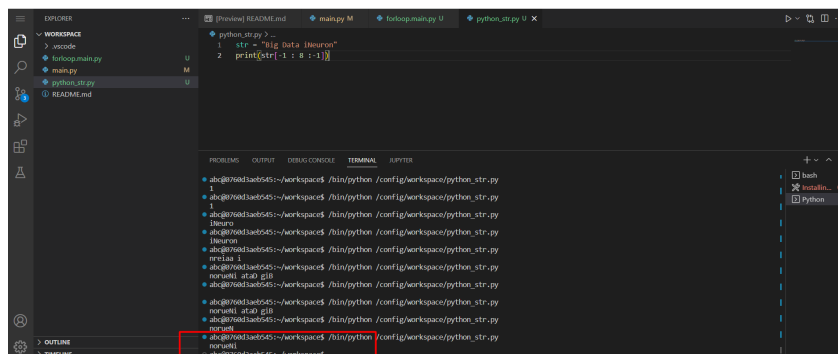
The screenshot shows a VS Code editor with a file named `python_str.py` containing the following code:

```
1 str = "Big Data iNeuron"  
2 print(str[9 : 16])
```

The terminal output shows the execution of the script, resulting in the string `iNeuro` being printed. The output is displayed in the terminal window at the bottom of the editor.

Q29. Write a code to get the desired output of the following

```
string = "Big Data iNeuron"  
desired_output = "norueNi"
```

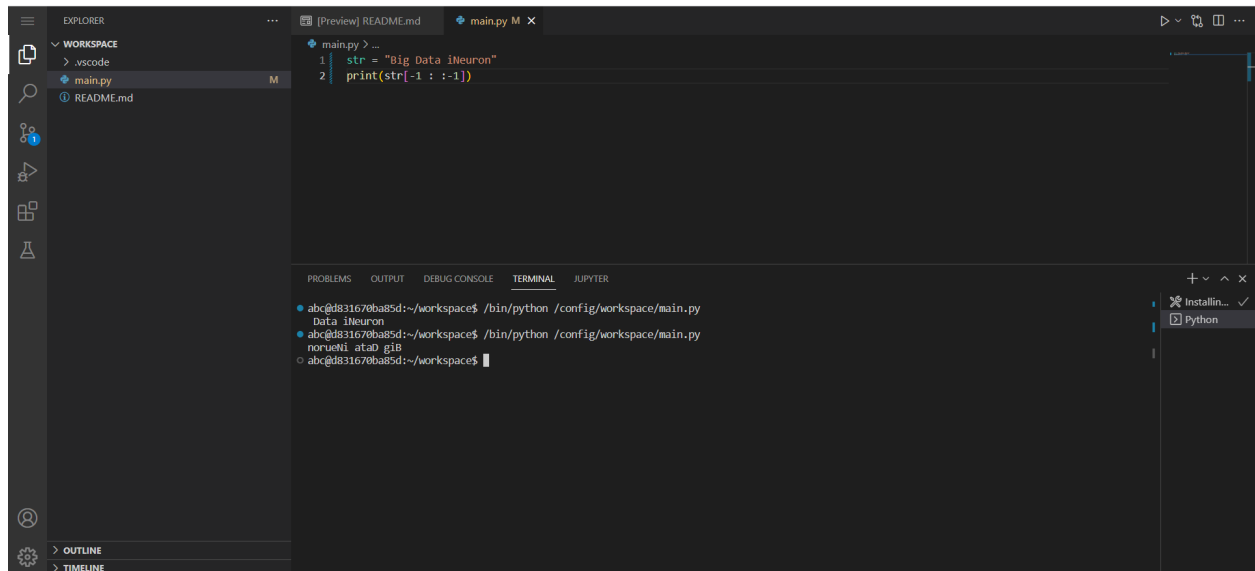


The screenshot shows a VS Code editor with a file named `python_str.py` containing the following code:

```
1 str = "Big Data iNeuron"  
2 print(str[-1 : 8 : -1])
```

The terminal output shows the execution of the script, resulting in the string `norueNi` being printed. The output is displayed in the terminal window at the bottom of the editor.

Q30. Reverse the string given in the above question.



The screenshot shows a VS Code editor with a workspace containing a file named `main.py`. The code in `main.py` is as follows:

```
1 str = "Big Data iNeuron"  
2 print(str[::-1])
```

The terminal output shows the execution of the script:

```
abc@ds31670ba85d:~/workspace$ ./bin/python /config/workspace/main.py  
Data iNeuron  
abc@ds31670ba85d:~/workspace$ ./bin/python /config/workspace/main.py  
nonetli atao giB  
abc@ds31670ba85d:~/workspace$
```

Q31. How can you delete entire string at once?

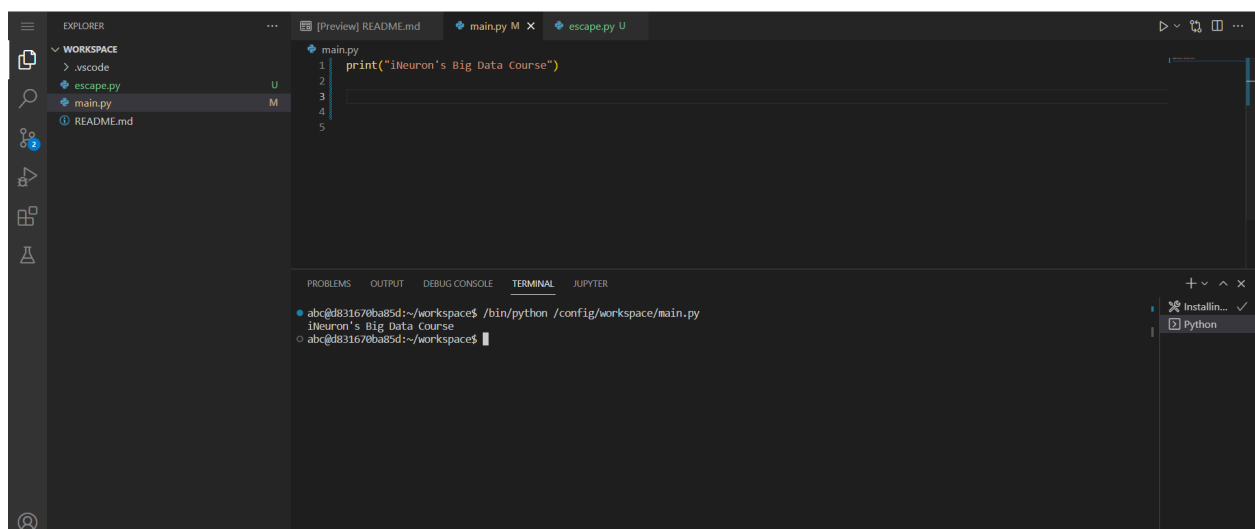
By using `del` command we can delete the entire string

Q32. What is escape sequence?

Escape sequence is backlash followed by character you to insert

Q33. How can you print the below string?

`'iNeuron's Big Data Course'`



The screenshot shows a VS Code editor with a workspace containing two files: `main.py` and `escape.py`. The code in `main.py` is as follows:

```
1 print("iNeuron's Big Data Course")  
2  
3  
4  
5
```

The terminal output shows the execution of the script:

```
abc@ds31670ba85d:~/workspace$ ./bin/python /config/workspace/main.py  
iNeuron's Big Data course  
abc@ds31670ba85d:~/workspace$
```

Q34. What is a list in Python?

List stores the number of elements within square brackets , List stores heterogeneous kind of data . Its sequential data.

Q35. How can you create a list in Python?

Lists are created by square brackets separated by commas .

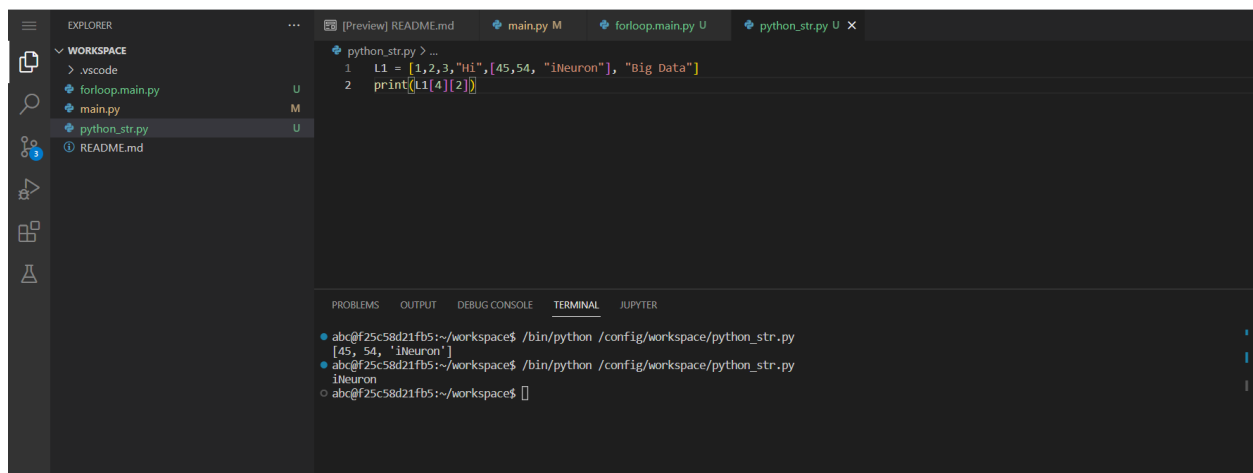
Eg : L1 = [1,2,3,14]

36. How can we access the elements in a list?

List can be access through the index

37. Write a code to access the word "iNeuron" from the given list.

```
lst = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]
```



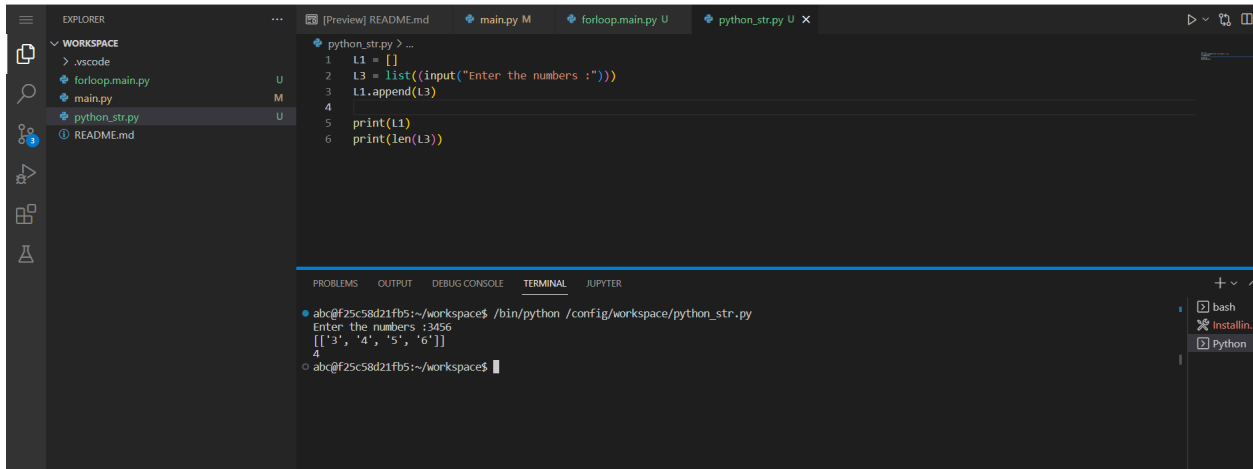
The screenshot shows a VS Code editor with a workspace containing several files: `forloop.main.py`, `main.py`, `python_str.py`, and `README.md`. The `python_str.py` file is open, showing the following code:

```
python_str.py > ...  
1 L1 = [1,2,3,"Hi",[45,54, "iNeuron"], "Big Data"]  
2 print(L1[4][2])
```

The terminal at the bottom shows the execution of the script:

```
abc@f25c58d21fb5:~/workspace$ /bin/python /config/workspace/python_str.py  
[45, 54, 'iNeuron']  
abc@f25c58d21fb5:~/workspace$ /bin/python /config/workspace/python_str.py  
iNeuron  
abc@f25c58d21fb5:~/workspace$
```

38. Take a list as an input from the user and find the length of the list.

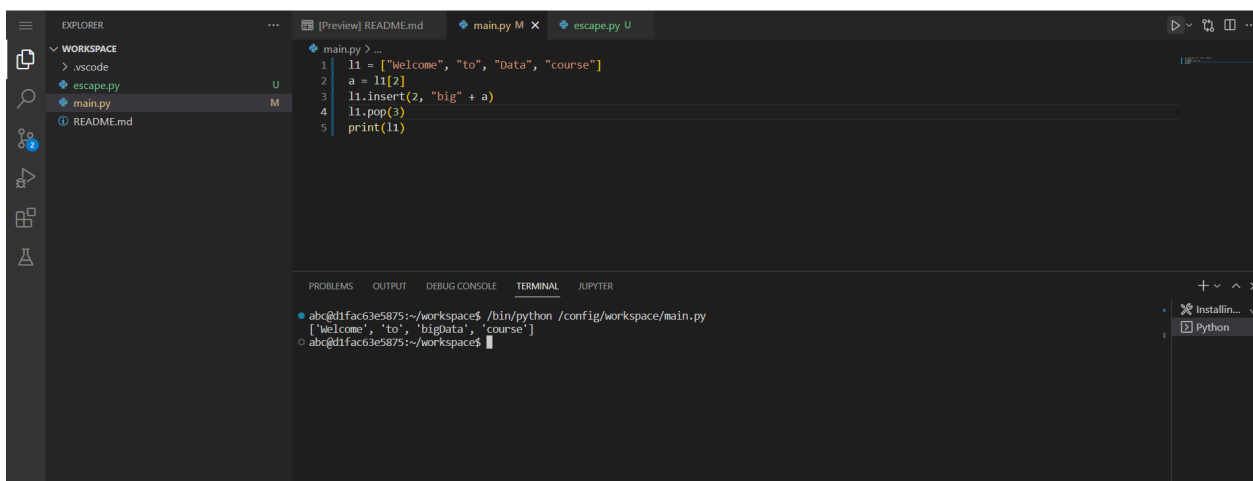


```
python_str.py > ...
1 l1 = []
2 l3 = list(input("Enter the numbers :"))
3 l1.append(l3)
4
5 print(l1)
6 print(len(l3))
```

```
abc@f25c58d21fb5:~/workspace$ /bin/python /config/workspace/python_str.py
Enter the numbers :3456
[['3', '4', '5', '6']]
4
abc@f25c58d21fb5:~/workspace$
```

39. Add the word "Big" in the 3rd index of the given list.

```
lst = ["Welcome", "to", "Data", "course"]
```



```
main.py > ...
1 l1 = ["Welcome", "to", "Data", "course"]
2 a = l1[2]
3 l1.insert(2, "big" + a)
4 l1.pop(3)
5 print(l1)
```

```
abc@1fac63e5875:~/workspace$ /bin/python /config/workspace/main.py
['Welcome', 'to', 'bigData', 'course']
abc@1fac63e5875:~/workspace$
```

40. What is a tuple? How is it different from list?

Tuple stores multiple items in a single variable, tuples are immutable and lists are mutable which means lists can be changed or edited but in tuples cannot be changed any value.

41. How can you create a tuple in Python?

Tuples can be created by enclosing all the comma-separated elements inside the parenthesis ()

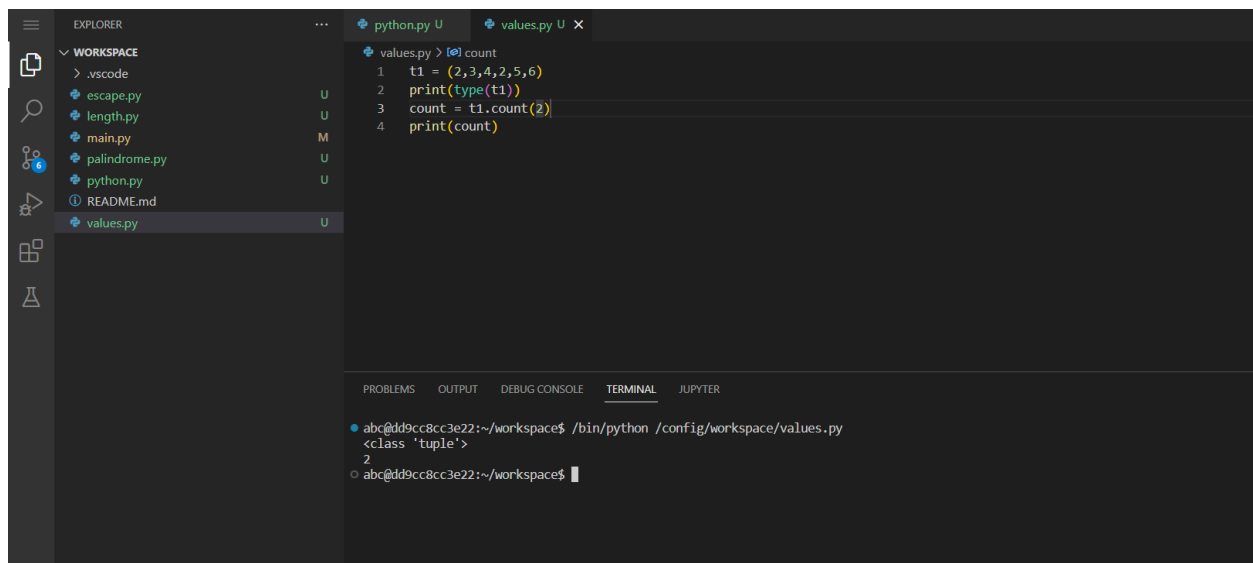
42. Create a tuple and try to add your name in the tuple. Are you able to do it? Support your answer with reason.

We cant able to add name in tuple because tuple is immutable and tuple does not support append function

43. Can two tuple be appended. If yes, write a code for it. If not, why?

No we cant append the tuple because tuple is immutable.

44. Take a tuple as an input and print the count of elements in it.



The screenshot shows a VS Code editor with a workspace containing several files. The file 'values.py' is open and contains the following code:

```
1 t1 = (2,3,4,2,5,6)
2 print(type(t1))
3 count = t1.count(2)
4 print(count)
```

The terminal output shows the execution of the script:

```
abc@dd9cc8cc3e22:~/workspace$ /bin/python /config/workspace/values.py
<class 'tuple'>
2
abc@dd9cc8cc3e22:~/workspace$
```

45. What are sets in Python?

Sets is collection which is unordered, unchangeable*, and unindexed. and also stores the unique or distinct value in it

46. How can you create a set?

Set is created by number of elements within the curly bais {}

47. Create a set and add "iNeuron" in your set.

The screenshot shows the VS Code interface with a workspace containing three files: `.vscode`, `escape.py`, and `main.py`. The `main.py` file is open and contains the following code:

```
1 set1 = set()
2 set1.add("iNeuron")
3 print(set1)
```

The terminal output shows the execution of the script:

```
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{'iNeuron'}
abc@1d5d635f4567:~/workspace$
```

48. Try to add multiple values using add() function.

The screenshot shows the VS Code interface with the same workspace. The `main.py` file is open and contains the following code:

```
1 set1 = set()
2 set1.add("iNeuron")
3 set1.add(3)
4 set1.add(3)
5 set1.add(23.45)
6 print(set1)
```

The terminal output shows the execution of the script:

```
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{'iNeuron'}
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{3, 'iNeuron', 23.45}
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{3, 'iNeuron', 23.45}
abc@1d5d635f4567:~/workspace$
```

49. How is update() different from add()?

In add function we can add only one element but in the update function we can add multiple elements

50. What is clear() in sets?

`clear()` is used to clear all the elements from a list . It does not take any parameter and doesn't return any value

51. What is frozen set?

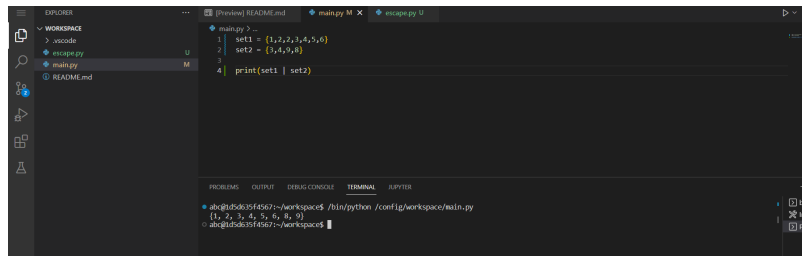
Frozen sets is used in python , frozen sets are immutable

52. How is frozen set different from set?

Set is mutable function and the frozen set is immutable function

53. What is union() in sets? Explain via code.

Union is used to combine two different sets if any duplicate values in set it will produced only distinct values



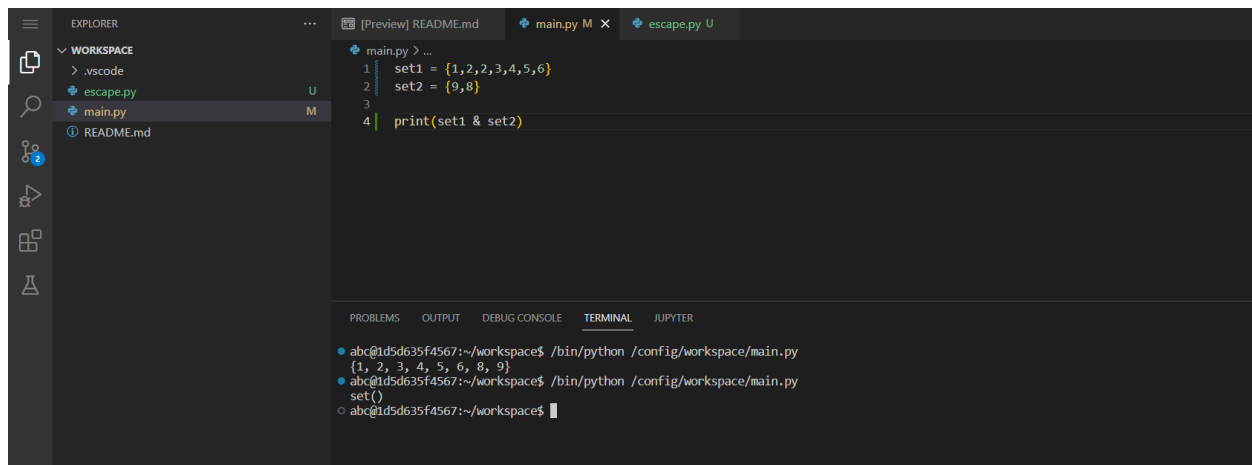
```
1 | set1 = {1,2,2,3,4,5,6}
2 | set2 = {3,4,9,8}
3 |
4 | print(set1 | set2)
```

Terminal output:

```
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{1, 2, 3, 4, 5, 6, 8, 9}
```

54. What is intersection() in sets? Explain via code.

Intersection sets are used to produce common values from the two different set . If no common value in the both set it will produced empty set



```
1 | set1 = {1,2,2,3,4,5,6}
2 | set2 = {9,8}
3 |
4 | print(set1 & set2)
```

Terminal output:

```
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
{1, 2, 3, 4, 5, 6, 8, 9}
abc@1d5d635f4567:~/workspace$ /bin/python /config/workspace/main.py
set()
```

55. What is dictionary in Python?

Dictionary will acts as key of data structure ,A dictionary consist of key value pairs , Each key mapped with associated value

56. How is dictionary different from all other data structures.

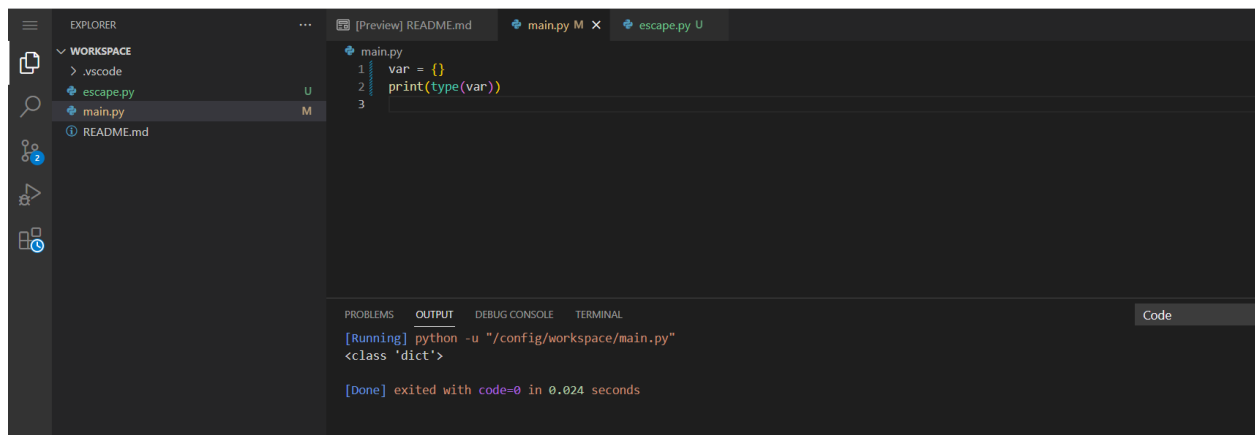
The dictionary stores many keys and each key is associated with a single value that is different from all other data structures.

57. How can we declare a dictionary in Python?

We can declare the dictionary within the curly basis separated by comma

58. What will the output of the following?

```
var = {}  
print(type(var))
```



The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The code editor displays a file named `main.py` with the following content:

```
1 var = {}  
2 print(type(var))  
3
```

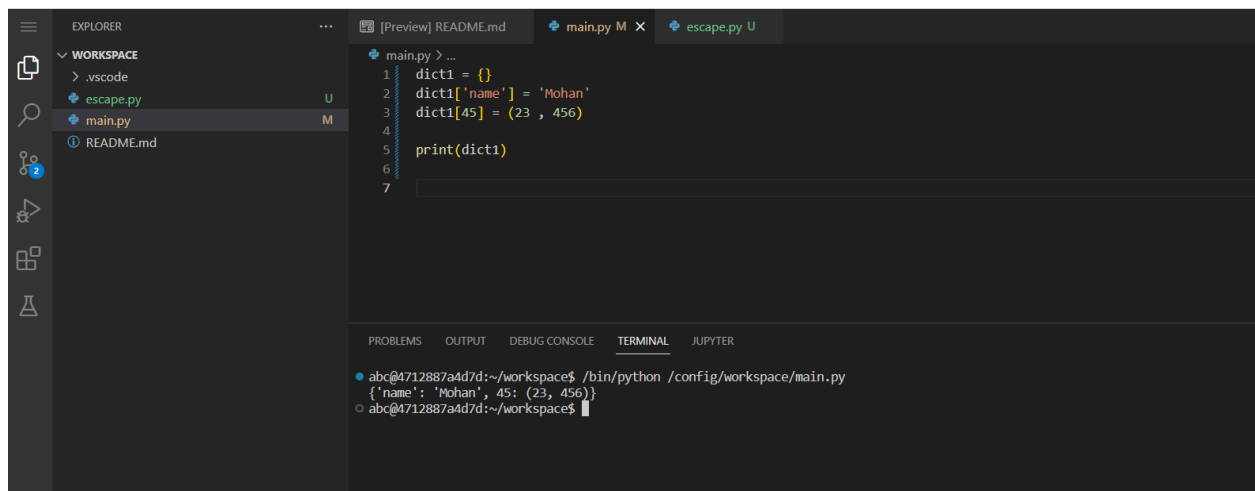
Below the code editor, the `OUTPUT` panel is visible, showing the execution results:

```
[Running] python -u "/config/workspace/main.py"  
<class 'dict'>  
  
[Done] exited with code=0 in 0.024 seconds
```

59. How can we add an element in a dictionary?

In dictionary by inserting new key value in it and assigning particular value to it

60. Create a dictionary and access all the values in that dictionary.



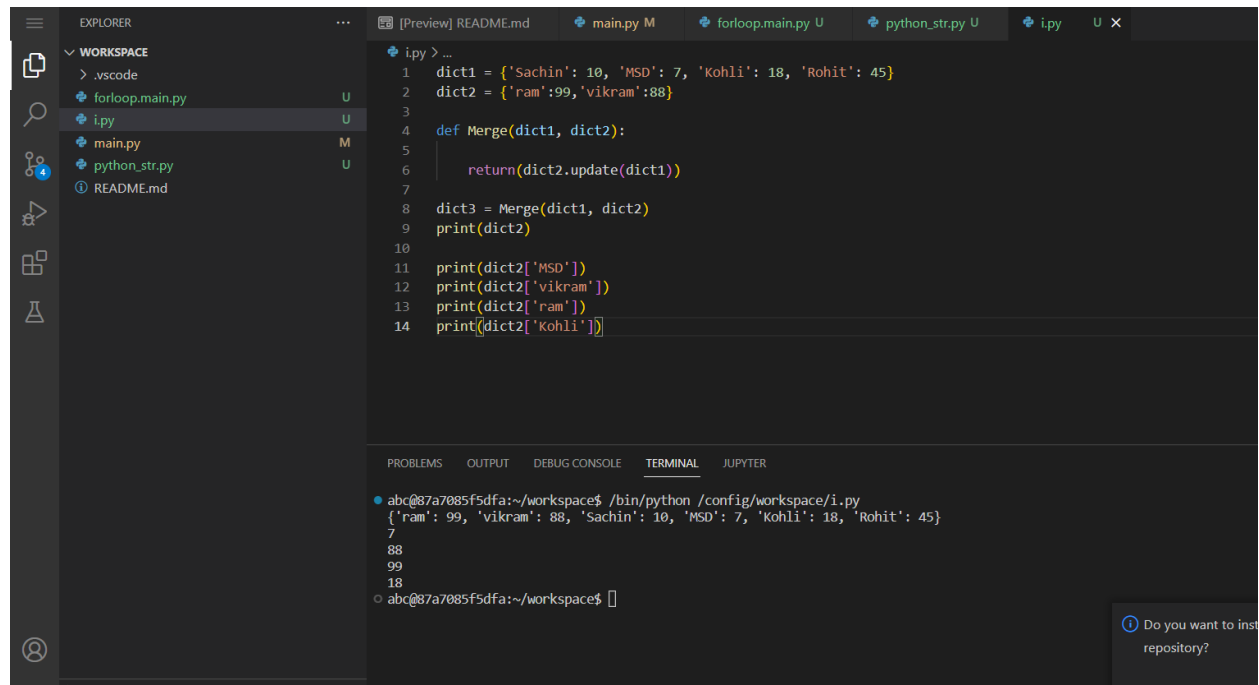
The screenshot shows the VS Code interface with a file explorer on the left and a code editor on the right. The code editor displays a file named `main.py` with the following content:

```
1 dict1 = {}  
2 dict1['name'] = 'Mohan'  
3 dict1[45] = (23, 456)  
4  
5 print(dict1)  
6  
7
```

Below the code editor, the `TERMINAL` panel is visible, showing the execution results:

```
abc@4712887a4d7d:~/workspace$ /bin/python /config/workspace/main.py  
{'name': 'Mohan', 45: (23, 456)}  
abc@4712887a4d7d:~/workspace$
```


61. Create a nested dictionary and access all the elements in the inner dictionary.



The screenshot shows a VS Code editor with a workspace containing several files: `.vscode`, `forloop.main.py`, `i.py`, `main.py`, `python_str.py`, and `README.md`. The `i.py` file is open in the editor, showing the following Python code:

```
1 dict1 = {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}
2 dict2 = {'ram': 99, 'vikram': 88}
3
4 def Merge(dict1, dict2):
5     return(dict2.update(dict1))
6
7
8 dict3 = Merge(dict1, dict2)
9 print(dict2)
10
11 print(dict2['MSD'])
12 print(dict2['vikram'])
13 print(dict2['ram'])
14 print(dict2['Kohli'])
```

The terminal output shows the execution of the script:

```
abc@87a7085f5dfa:~/workspace$ /bin/python /config/workspace/i.py
{'ram': 99, 'vikram': 88, 'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}
7
88
99
18
abc@87a7085f5dfa:~/workspace$
```

62. What is the use of get() function?

Get() function is used to retrieve value from python

63. What is the use of items() function?

Items() function is used to return the list with all the dictionary keys with values

64. What is the use of pop() function?

Pop() is used to remove elements from specified position

65. What is the use of popitems() function?

Popitems() is remove last inserted key value pairs from dictionary and return its as tuple

66. What is the use of keys() function?

Keys() function return the view object , the view object contains all keys in dictionary as list

67. What is the use of values() function?

Values() function return the view object. The view contains all values in dictionary as list

68. What are loops in Python?

Loops are means repeating specific code under the condition are satisfied .There are three loops in python for loop , while loop ,nested loop

69. How many type of loop are there in Python?

For loop , while loop , nested loop

70. What is the difference between for and while loops?

In for loop we already know the number of iteration but in while loop we don't know the iteration

71. What is the use of continue statement?

Continue is used to end the current iteration in for loop / while loop and continues to the next iteration

72. What is the use of break statement?

Break is used to terminate the entire loop iteration

73. What is the use of pass statement?

Pass statement is useful when you don't write the implementation of function but we want to implement in the future code .when pass statement is used in code nothing will happen it pass the current code .

74. What is the use of range() function?

Range() function is to return the sequence of numbers with specified value .It starts from zero and increments by 1 and ends with specified value.

75. How can you loop over a dictionary?

We can loop over a dictionary by using for loop it will return the key of dictionary and values also be return by using index

Coding problems

76. Write a Python program to find the factorial of a given number.

```
main.py > num
1 num = int(input(" Enter the number"))
2 def factorial(a):
3     if a == 0:
4         return 1
5     else:
6         return (a * factorial(a-1))
7
8
9
10
11 result = factorial(num)
12 print("The factorial of a", result)
13
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the numbers
The factorial of a 720
abc@997f1cea6fed:~/workspace$
```

77. Write a Python program to calculate the simple interest. Formula to calculate simple interest is $SI = (PRT)/100$

```
main.py > ...
1 num1 = int(input(" Enter the number"))
2 num2 = int(input(" Enter the number"))
3 num3 = int(input(" Enter the number"))
4 def simple_interest(a,b,c):
5     if a == 0:
6         return 0
7     else:
8         return ((a*b*c)/100)
9
10
11
12
13 result = simple_interest(num1,num2,num3)
14 print("The simple interest is", result)
15
16
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the numbers
The factorial of a 720
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number3
Enter the number4
Enter the number5
The factorial of a 0.6
```

78. Write a Python program to calculate the compound interest. Formula of compound interest is $A = P(1 + R/100)^t$.

```
main.py > simple_interest
1 num1 = int(input(" Enter the number"))
2 num2 = int(input(" Enter the number"))
3 num3 = int(input(" Enter the number"))
4 def simple_interest(a,b,c):
5     if a == 0:
6         return 0
7     else:
8         return a * (1+b//100)^ c
9
10
11
12
13 result = simple_interest(num1,num2,num3)
14 print("The compound interest is", result)
15
16
17
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
return a * (1+b//100)^ c
TypeError: unsupported operand type(s) for ^: 'float' and 'int'
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number2
Enter the number3
Enter the number4
The compound interest is 6
abc@997f1cea6fed:~/workspace$
```

79. Write a Python program to check if a number is prime or not.

```
main.py > ...
1 num = int(input(" Enter the number"))
2
3
4 if num > 1:
5     for i in range(2,num):
6         if (num % i) == 0:
7             print("the number is not prime number")
8             break
9         else:
10            print("the number is prime number")
11 else:
12     print("the number is not prime number")
13
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
ValueError: invalid literal for int() with base 10: '/bin/python /config/workspace/main.py'
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number3
the number is prime number
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number4
the number is not prime number
abc@997f1cea6fed:~/workspace$
```

80. Write a Python program to check Armstrong Number.

The screenshot shows the VS Code editor with a workspace containing `.vscode`, `ast.py`, `main.py`, and `README.md`. The `ast.py` file is open and contains the following Python code:

```
1 num = int(input("Enter a number: "))
2 sum = 0
3
4 temp = num
5 while temp > 0:
6     digit = temp % 10
7     sum += digit ** 3
8     temp //= 10
9
10 if num == sum:
11     print(num, "is an Armstrong number")
12 else:
13     print(num, "is not an Armstrong number")
```

The TERMINAL panel at the bottom shows the execution of the program:

```
abc@cce62760f739:~/workspace$ /bin/python /config/workspace/ast.py
Enter a number: 153
153 is an Armstrong number
abc@cce62760f739:~/workspace$ /bin/python /config/workspace/ast.py
Enter a number: 164
164 is not an Armstrong number
abc@cce62760f739:~/workspace$
```

81. Write a Python program to find the n-th Fibonacci Number.

The screenshot shows the VS Code editor with a workspace containing `.vscode`, `escape.py`, `main.py`, and `README.md`. The `main.py` file is open and contains the following Python code:

```
1 x = int(input("Enter the number"))
2 def Fibonacci_Number(n):
3     if n <= 2:
4         return n-1
5     else:
6         return ((n-1)+(n-2))
7
8
9
10 result = Fibonacci_Number(x)
11 print("The Fibonacci Number is",result)
12
```

The TERMINAL panel at the bottom shows the execution of the program:

```
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number7
The Fibonacci Number is 11
abc@997f1cea6fed:~/workspace$
```

82. Write a Python program to interchange the first and last element in a list.

The screenshot shows the Visual Studio Code interface. The Explorer panel on the left displays the workspace with files: .vscode, escape.py, main.py, and README.md. The main editor window shows the code for main.py, which defines a function swaplist(x) that swaps the first and last elements of a list x. The function uses pop() to remove elements and insert() to add them back in swapped positions. The main program prompts the user to enter a number, which is then used to create a list x. The function swaplist(x) is called, and the resulting list is printed.

```
1 def swaplist(x):
2     first = x.pop(0)
3     last = x.pop(-1)
4     x.insert(0,last)
5     x.append(first)
6     return list(x)
7 x = list((input("Enter the number")))
8 swaplist(x)
9 print(list(x))
10
11
12
13
14
15
16
```

The terminal output shows the execution of the program. The user enters the number 345, and the program outputs the list ['5', '4', '3'].

```
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number345
['5', '4', '3']
abc@997f1cea6fed:~/workspace$
```

83. Write a Python program to swap two elements in a list.

This screenshot is identical to the one above, showing the same Python program for swapping two elements in a list. The code defines a function swaplist(x) that swaps the first and last elements of a list x. The main program prompts the user to enter a number, which is then used to create a list x. The function swaplist(x) is called, and the resulting list is printed.

```
1 def swaplist(x):
2     first = x.pop(0)
3     last = x.pop(-1)
4     x.insert(0,last)
5     x.append(first)
6     return list(x)
7 x = list((input("Enter the number")))
8 swaplist(x)
9 print(list(x))
10
11
12
13
14
15
16
```

The terminal output shows the execution of the program. The user enters the number 345, and the program outputs the list ['5', '4', '3'].

```
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number345
['5', '4', '3']
abc@997f1cea6fed:~/workspace$
```

84. Write a Python program to find N largest element from a list.

The screenshot shows the VS Code interface with a workspace containing `escape.py`, `main.py`, and `README.md`. The `main.py` file is open in the editor, showing a Python function `largest_element(x)` that returns the maximum value of a list `x`. The function is called with user input. The terminal window at the bottom shows the execution of the program, where the user enters the number 4, and the program outputs "largest element 4".

```
def largest_element(x):
    max(x)
    return max(x)

x = list((input("Enter the number")))
largest_element(x)
print("largest element ", max(x))
```

```
abc@997f1cea6fed:~/workspace$ 56234
bash: 56234: command not found
abc@997f1cea6fed:~/workspace$
abc@997f1cea6fed:~/workspace$ /bin/python /config/workspace/main.py
Enter the number:4
largest element 4
abc@997f1cea6fed:~/workspace$
```

85. Write a Python program to find cumulative sum of a list.

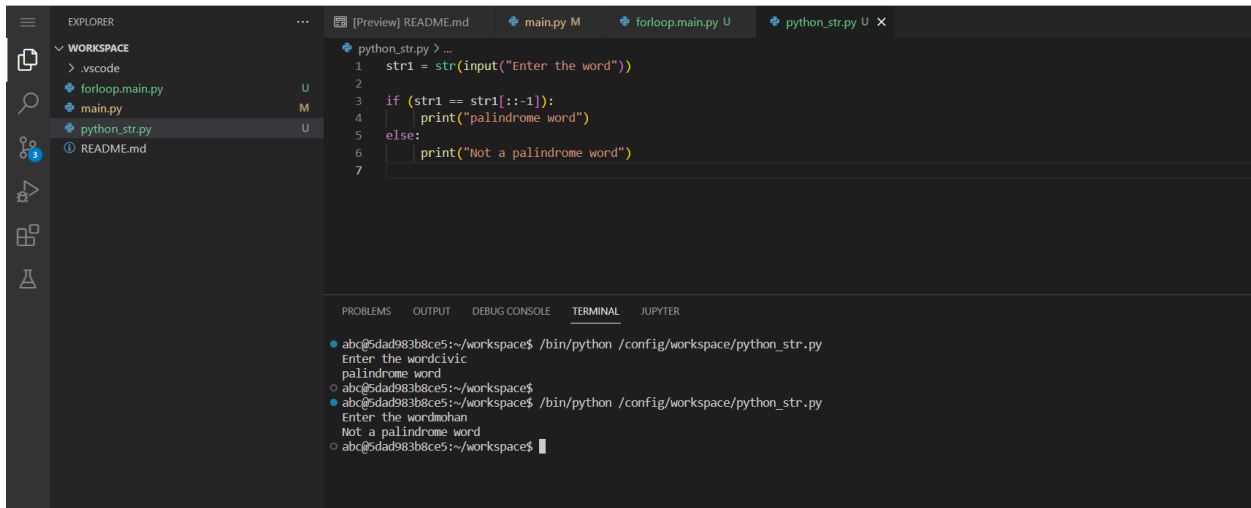
The screenshot shows the VS Code interface with a workspace containing `ast.py` and `main.py`. The `ast.py` file is open in the editor, showing a Python program that calculates the cumulative sum of a list `[10, 20, 30]` and stores the result in a new list `l1`. The terminal window at the bottom shows the execution of the program, which results in an `AttributeError: 'list' object has no attribute 'update'` because the `list` object does not have an `update` method. The error message also shows the state of the list `l1` as `[10, 30, 60]`.

```
list = [10,20,30]
print(type(list))
l1 = []
sum = 0
for i in range(0,len(list)):
    sum = sum + list[i]
    l1.append(sum)

print(l1)
```

```
l1.update(sum)
AttributeError: 'list' object has no attribute 'update'
abc@cce62760f739:~/workspace$ /bin/python /config/workspace/ast.py
<class 'list'>
[60]
abc@cce62760f739:~/workspace$ /bin/python /config/workspace/ast.py
<class 'list'>
[10, 30, 60]
abc@cce62760f739:~/workspace$
```

86. Write a Python program to check if a string is palindrome or not.



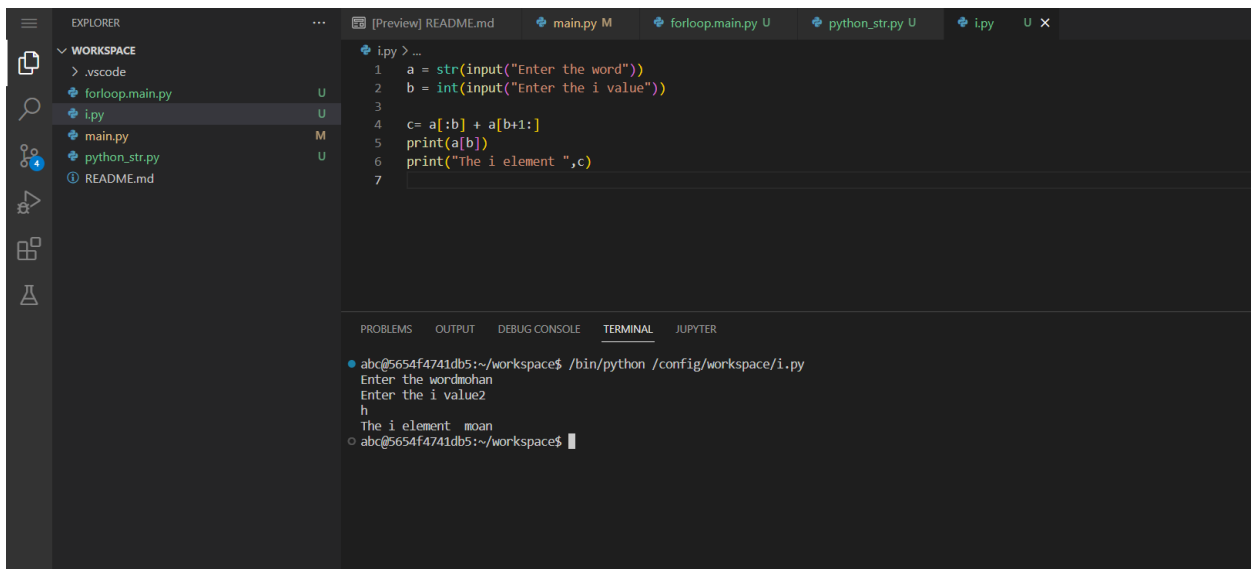
The screenshot shows the VS Code interface with a workspace containing files: `.vscode`, `forloop.main.py`, `main.py`, `python_str.py`, and `README.md`. The `python_str.py` file is open, showing the following code:

```
python_str.py > ...
1  str1 = str(input("Enter the word"))
2
3  if (str1 == str1[::-1]):
4      print("palindrome word")
5  else:
6      print("Not a palindrome word")
7
```

The terminal output shows the execution of the program:

```
abc@5dad983b8ce5:~/workspace$ /bin/python /config/workspace/python_str.py
Enter the wordcivic
palindrome word
abc@5dad983b8ce5:~/workspace$ /bin/python /config/workspace/python_str.py
Enter the wordmohan
Not a palindrome word
abc@5dad983b8ce5:~/workspace$
```

87. Write a Python program to remove i'th element from a string.



The screenshot shows the VS Code interface with a workspace containing files: `.vscode`, `forloop.main.py`, `ipy`, `main.py`, `python_str.py`, and `README.md`. The `ipy` file is open, showing the following code:

```
ipy > ...
1  a = str(input("Enter the word"))
2  b = int(input("Enter the i value"))
3
4  c= a[:b] + a[b+1:]
5  print(a[b])
6  print("The i element ",c)
7
```

The terminal output shows the execution of the program:

```
abc@5654f4741db5:~/workspace$ /bin/python /config/workspace/i.py
Enter the wordmohan
Enter the i value2
h
The i element moan
abc@5654f4741db5:~/workspace$
```

88. Write a Python program to check if a substring is present in a given string.

The screenshot shows the VS Code interface with a workspace containing files: `.vscode`, `forloop.main.py`, `i.py`, `main.py`, `python_str.py`, and `README.md`. The `i.py` file is open in the editor, showing the following code:

```
1 a = str(input("Enter the string"))
2 b = str(input("Enter the substring"))
3
4 if b in a :
5     print("true")
6 else:
7     print("false")
8
```

The terminal at the bottom shows the execution of the program:

```
abc@5654f4741db5:~/workspace$ /bin/python /config/workspace/i.py
Enter the stringmohan
Enter the substringhre
false
abc@5654f4741db5:~/workspace$
abc@5654f4741db5:~/workspace$ /bin/python /config/workspace/i.py
Enter the stringmohan
Enter the substringhan
true
abc@5654f4741db5:~/workspace$
```

Q89. Write a Python program to find words which are greater than given length k.

The screenshot shows the VS Code interface with a workspace containing files: `.vscode`, `ast.py`, `main.py`, `new.py`, `README.md`, and `star.py`. The `ast.py` file is open in the editor, showing the following code:

```
1 text1 = input("Enter the string")
2 k = int(input("Enter the number"))
3 L1= []
4 text2 = text1 . split(" ")
5 for i in text2:
6     if len(i) > k:
7         L1.append(i)
8 print(L1)
```

The terminal at the bottom shows the execution of the program:

```
abc@7ccddf24aeb4:~/workspace$ /bin/python /config/workspace/ast.py
Enter the stringthe world is beautifual
Enter the number3
['world', 'beautifual']
abc@7ccddf24aeb4:~/workspace$
```

Q90. Write a Python program to extract unquire dictionary values.

The screenshot shows the VS Code interface with a workspace containing several files. The file 'values.py' is selected and open in the editor. The code in 'values.py' is as follows:

```
1 dict1 = {'Sachin': 10, 'MSD': 7, 'Kohli': 45, 'Rohit': 45}
2
3 values = set(dict1.values())
4 print(values)
```

The terminal at the bottom shows the execution of the script using the command `/bin/python /config/workspace/values.py`. The output is `{10, 45, 7}`.

Q91. Write a Python program to merge two dictionary.

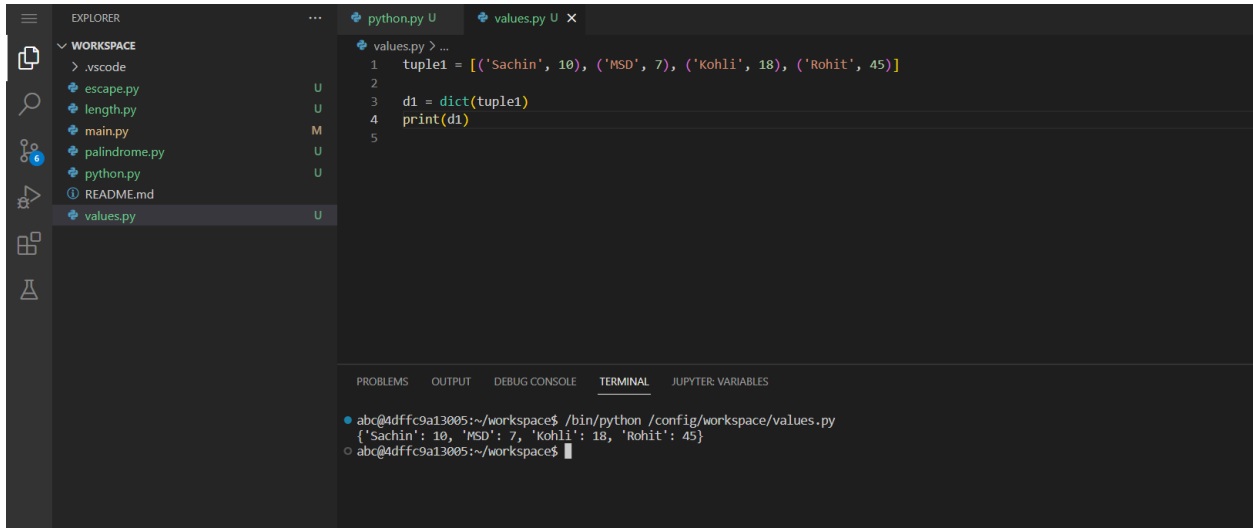
The screenshot shows the VS Code interface with a workspace containing several files. The file 'i.py' is selected and open in the editor. The code in 'i.py' is as follows:

```
1 dict1 = {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}
2 dict2 = {'ram': 99, 'vikram': 88}
3
4 def Merge(dict1, dict2):
5     return(dict2.update(dict1))
6
7 dict3 = Merge(dict1, dict2)
8 print(dict2)
```

The terminal at the bottom shows the execution of the script using the command `/bin/python /config/workspace/i.py`. The output is `{'ram': 99, 'vikram': 88, 'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}`.

92. Write a Python program to convert a list of tuples into dictionary.

Input : [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
Output : {'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}



The screenshot shows the VS Code interface with a file explorer on the left containing files like .vscode, escape.py, length.py, main.py, palindrome.py, python.py, README.md, and values.py. The main editor displays a Python script in values.py:

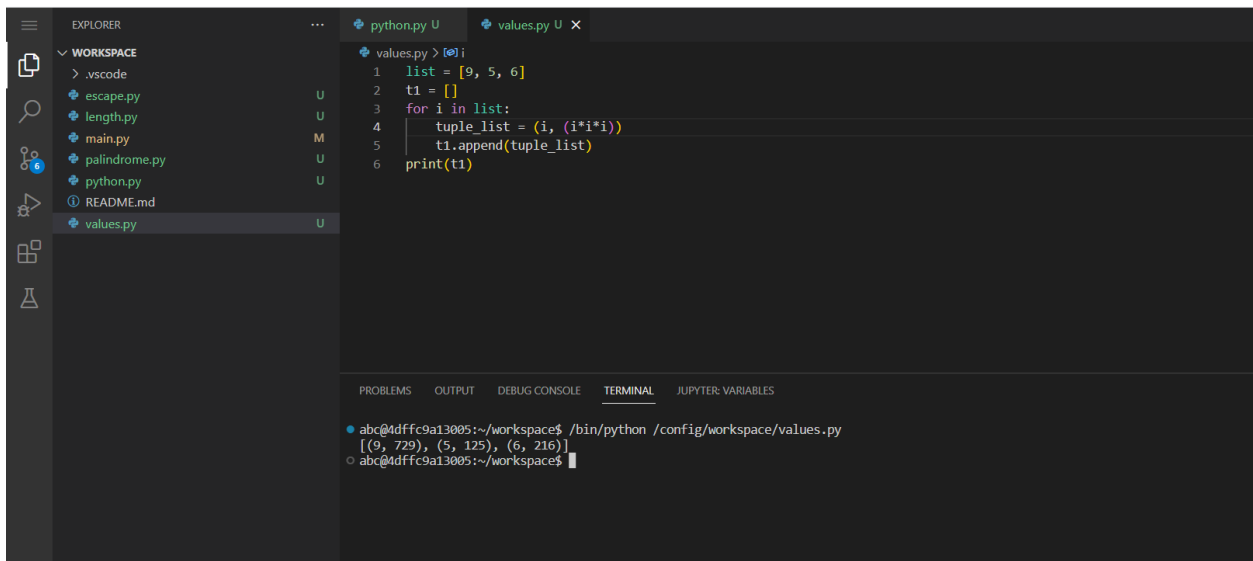
```
1 tuple1 = [('Sachin', 10), ('MSD', 7), ('Kohli', 18), ('Rohit', 45)]
2
3 d1 = dict(tuple1)
4 print(d1)
5
```

The terminal at the bottom shows the command `python /config/workspace/values.py` being executed, resulting in the output: `{'Sachin': 10, 'MSD': 7, 'Kohli': 18, 'Rohit': 45}`.

93. Write a Python program to create a list of tuples from given list having number and its cube in each tuple.

Input: `list = [9, 5, 6]`

Output: `[(9, 729), (5, 125), (6, 216)]`



The screenshot shows the VS Code interface with the same file explorer. The main editor displays a Python script in values.py:

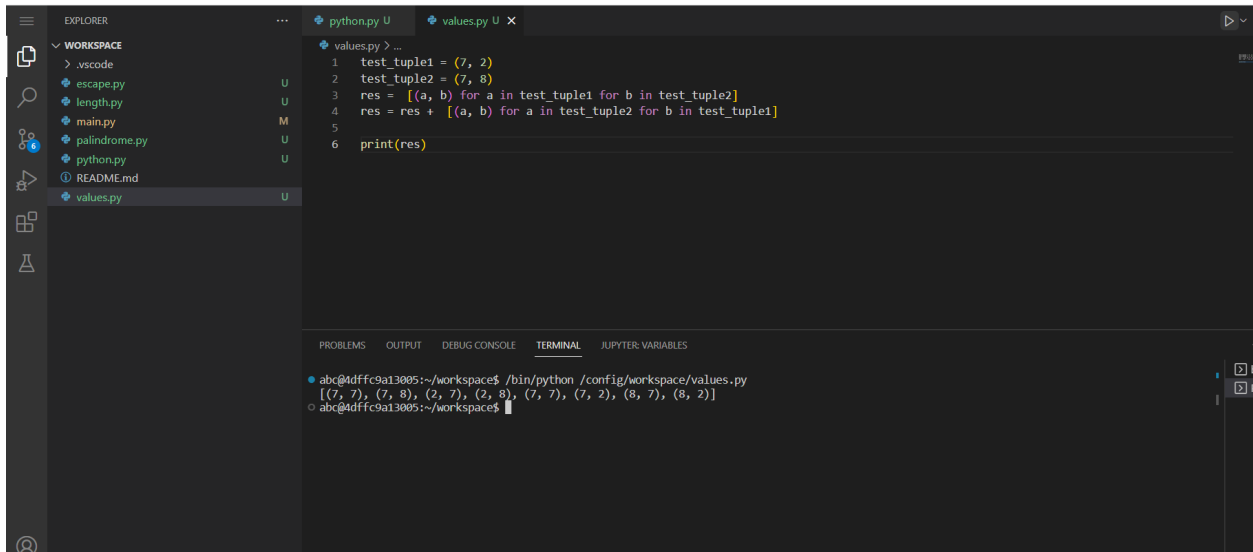
```
1 list = [9, 5, 6]
2 t1 = []
3 for i in list:
4     tuple_list = (i, (i*i*i))
5     t1.append(tuple_list)
6 print(t1)
```

The terminal at the bottom shows the command `python /config/workspace/values.py` being executed, resulting in the output: `[(9, 729), (5, 125), (6, 216)]`.

94. Write a Python program to get all combinations of 2 tuples.

Input : `test_tuple1 = (7, 2), test_tuple2 = (7, 8)`

Output : `[(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]`



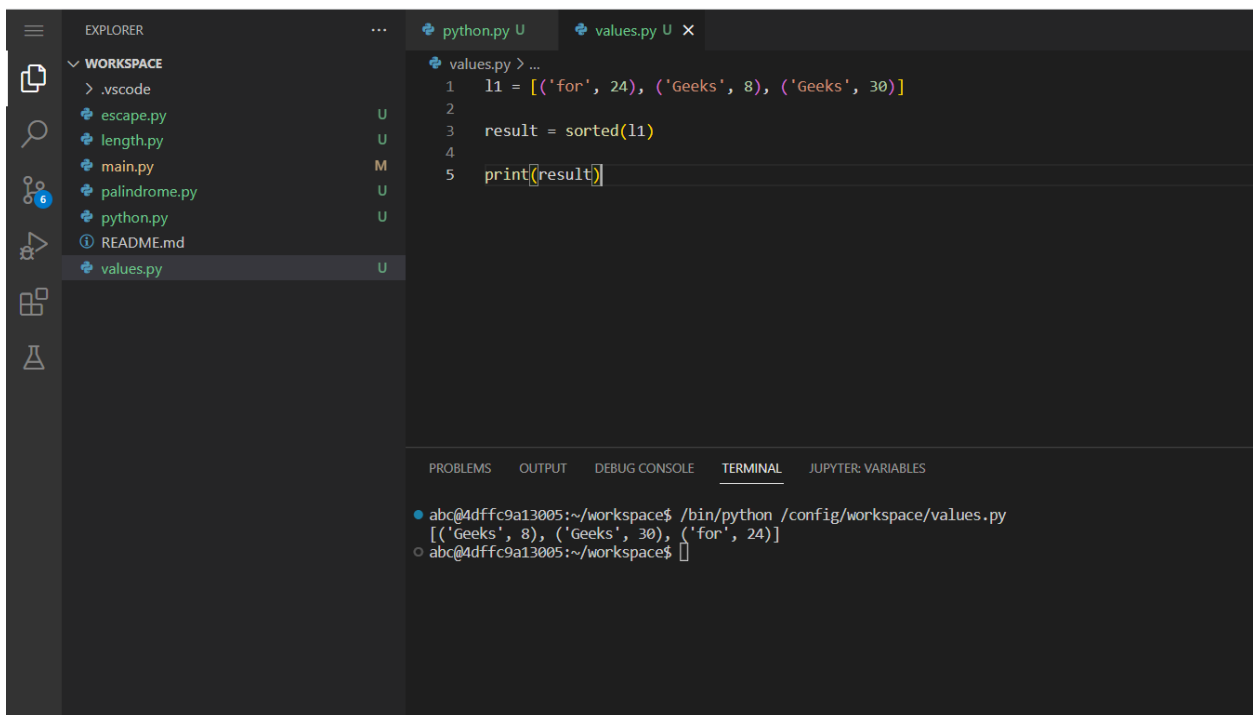
```
python.py U  values.py U X
values.py > ...
1 test_tuple1 = (7, 2)
2 test_tuple2 = (7, 8)
3 res = [(a, b) for a in test_tuple1 for b in test_tuple2]
4 res = res + [(a, b) for a in test_tuple2 for b in test_tuple1]
5
6 print(res)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES
abc@4dffc9a13005:~/workspaces$ /bin/python /config/workspace/values.py
[(7, 7), (7, 8), (2, 7), (2, 8), (7, 7), (7, 2), (8, 7), (8, 2)]
abc@4dffc9a13005:~/workspaces$
```

95. Write a Python program to sort a list of tuples by second item.

Input : [('for', 24), ('Geeks', 8), ('Geeks', 30)]

Output : [('Geeks', 8), ('for', 24), ('Geeks', 30)]



```
python.py U  values.py U X
values.py > ...
1 l1 = [('for', 24), ('Geeks', 8), ('Geeks', 30)]
2
3 result = sorted(l1)
4
5 print(result)

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER VARIABLES
abc@4dffc9a13005:~/workspace$ /bin/python /config/workspace/values.py
[('Geeks', 8), ('Geeks', 30), ('for', 24)]
abc@4dffc9a13005:~/workspace$
```

96. Write a python program to print below pattern.

```
*
* *
* * *
```

```

* * * *
* * * * *

```

The screenshot shows the VS Code interface with a workspace containing several files. The file `values.py` is open and contains the following Python code:

```

1 n = int(input("Enter the number"))
2 for i in range(n):
3     for j in range(i):
4         print("=", end="")
5     print("\n")

```

The terminal window shows the execution of the program:

```

abc@dd9cc8cc3e22:~/workspace$ /bin/python /config/workspace/values.py
Enter the number6

*
**
***
****
*****

```

97. Write a python program to print below pattern.

```

*
* *
* * *
* * * *
* * * * *

```

The screenshot shows the VS Code interface with a workspace containing several files. The file `sr.py` is open and contains the following Python code:

```

1 n=5;i=0
2 while(i<=n):
3     print(" " * (n - i) + "*" * i)
4     i+=1

```

The terminal window shows the execution of the program:

```

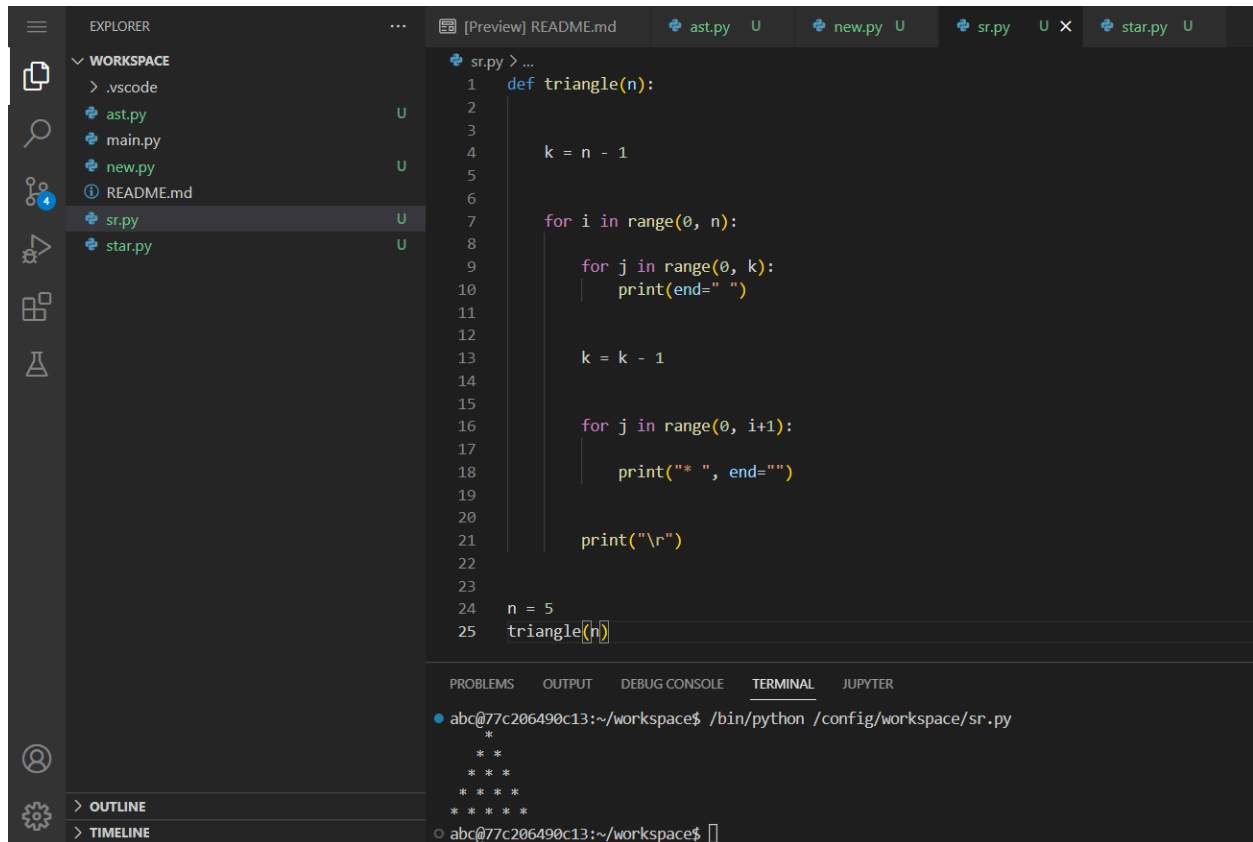
abc@77c206490c13:~/workspace$ /bin/python /config/workspace/sr.py

*
**
***
****
*****

```

Q98. Write a python program to print below pattern.

```
  *
 * *
* * *
* * * *
* * * * *
```



```
sr.py > ...
1  def triangle(n):
2
3
4      k = n - 1
5
6
7      for i in range(0, n):
8
9          for j in range(0, k):
10             print(end=" ")
11
12
13         k = k - 1
14
15
16         for j in range(0, i+1):
17
18             print("* ", end="")
19
20
21         print("\n")
22
23
24     n = 5
25     triangle(n)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER

```
abc@77c206490c13:~/workspace$ /bin/python /config/workspace/sr.py
*
* *
* * *
* * * *
* * * * *
abc@77c206490c13:~/workspace$
```

Q99. Write a python program to print below pattern.

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

The screenshot shows a VS Code editor with a workspace containing several files: `.vscode`, `ast.py`, `main.py`, `new.py`, `README.md`, `sr.py`, and `star.py`. The `sr.py` file is selected and open in the editor. The code in `sr.py` is as follows:

```
1
2
3 def numpat(n):
4
5     num = 1
6
7     for i in range(0, n):
8
9         num = 1
10
11         for j in range(0, i+1):
12
13             print(num, end=" ")
14
15             num = num + 1
16
17         print("\r")
18
19
20 n = 5
21 numpat(n)
```

The terminal at the bottom shows the command `/bin/python /config/workspace/sr.py` being executed, resulting in the following output:

```
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
```

Q100. Write a python program to print below pattern.

```
A
B B
C C C
D D D D
E E E E E
```

