

# DAP

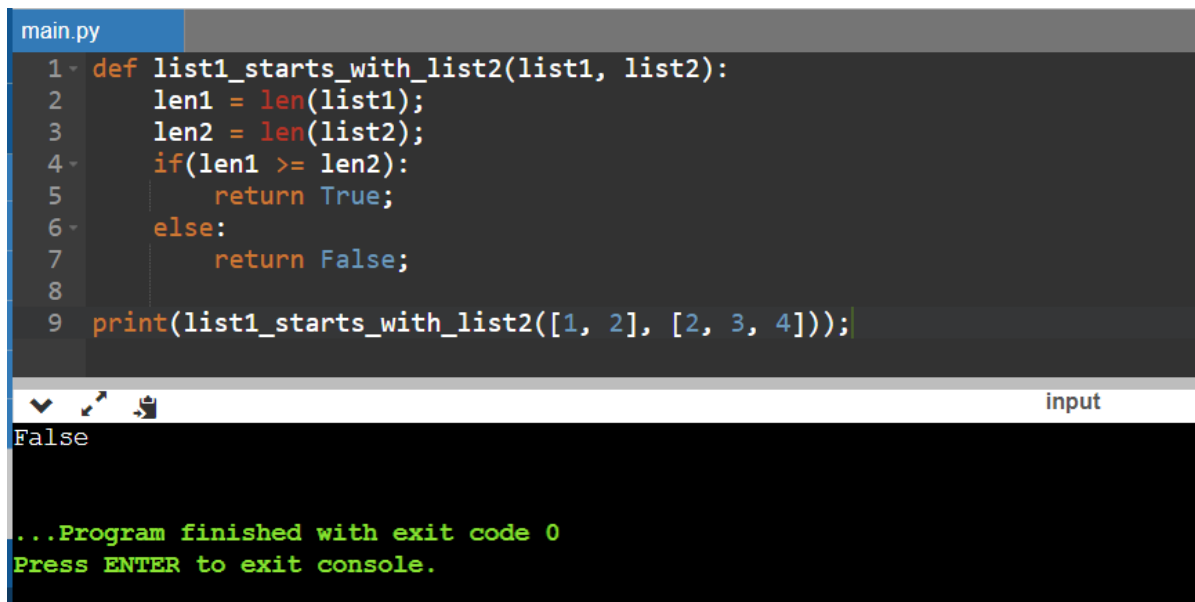
Assignment:01

Name:-**Chetan**

Roll no:-20001011502

Q1> Write a function with the signature `def list1_starts_with_list2(list1, list2)`, which returns True iff list1 is at least as long as list2, and the first `len(list2)` elements of list1 are the same as list2. Note: `len(lis)` is the length of the list `lis`, i.e., the number of elements in `lis`. First write the function without using slicing ("slicing" means doing things like `list1[2:5]`), and using a loop.

Ans>



```
main.py
1 def list1_starts_with_list2(list1, list2):
2     len1 = len(list1);
3     len2 = len(list2);
4     if(len1 >= len2):
5         return True;
6     else:
7         return False;
8
9 print(list1_starts_with_list2([1, 2], [2, 3, 4]));

input
False

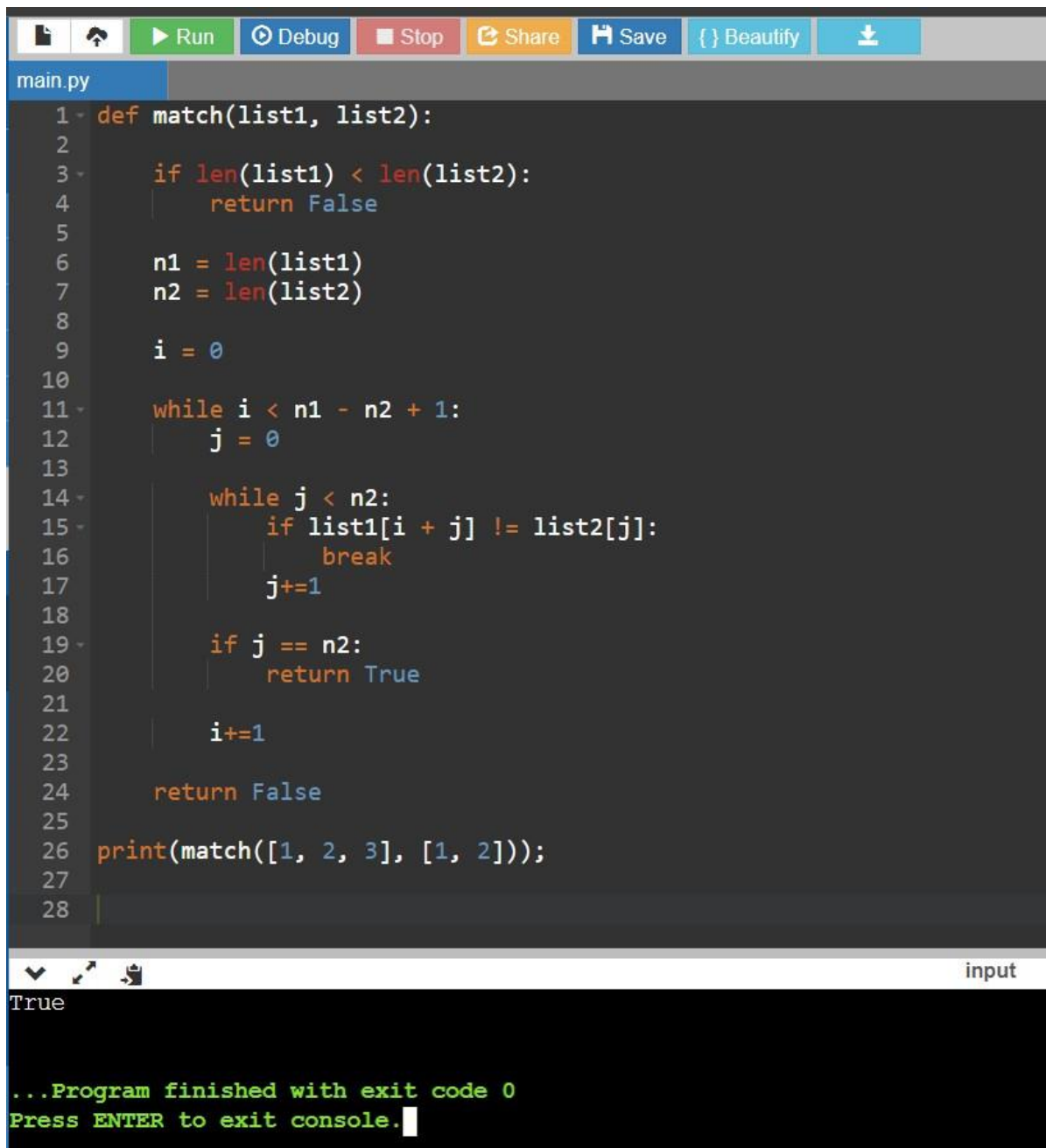
...Program finished with exit code 0
Press ENTER to exit console.
```

Q2> Write a function with the signature `def match_pattern(list1, list2)` which returns True iff the pattern list2 appears in list1. In other words, we return True iff there is an  $i$  such that  $0 \leq i \leq \text{len}(\text{list1}) - \text{len}(\text{list2})$  and  $\text{list1}[i] = \text{list2}[0]$   $\text{list1}[i + 1] = \text{list2}[1]$

.  
.  
.

$\text{list1}[i + \text{len}(\text{list2}) - 1] = \text{list2}[-1]$  For example, if list1 is [4, 10, 2, 3, 50, 100] and list2 is [2, 3, 50], `match_pattern(list1, list2)` returns True since the pattern [2, 3, 50] appears in list1

Ans



```
main.py
1 def match(list1, list2):
2
3     if len(list1) < len(list2):
4         return False
5
6     n1 = len(list1)
7     n2 = len(list2)
8
9     i = 0
10
11     while i < n1 - n2 + 1:
12         j = 0
13
14         while j < n2:
15             if list1[i + j] != list2[j]:
16                 break
17             j += 1
18
19         if j == n2:
20             return True
21
22         i += 1
23
24     return False
25
26 print(match([1, 2, 3], [1, 2]));
27
28
```

input

True

...Program finished with exit code 0  
Press ENTER to exit console.

Q3 > Write a function with the signature `def repeats(list0)`, which returns `True` if `list0` contains at least two adjacent elements with the same value.

Ans>



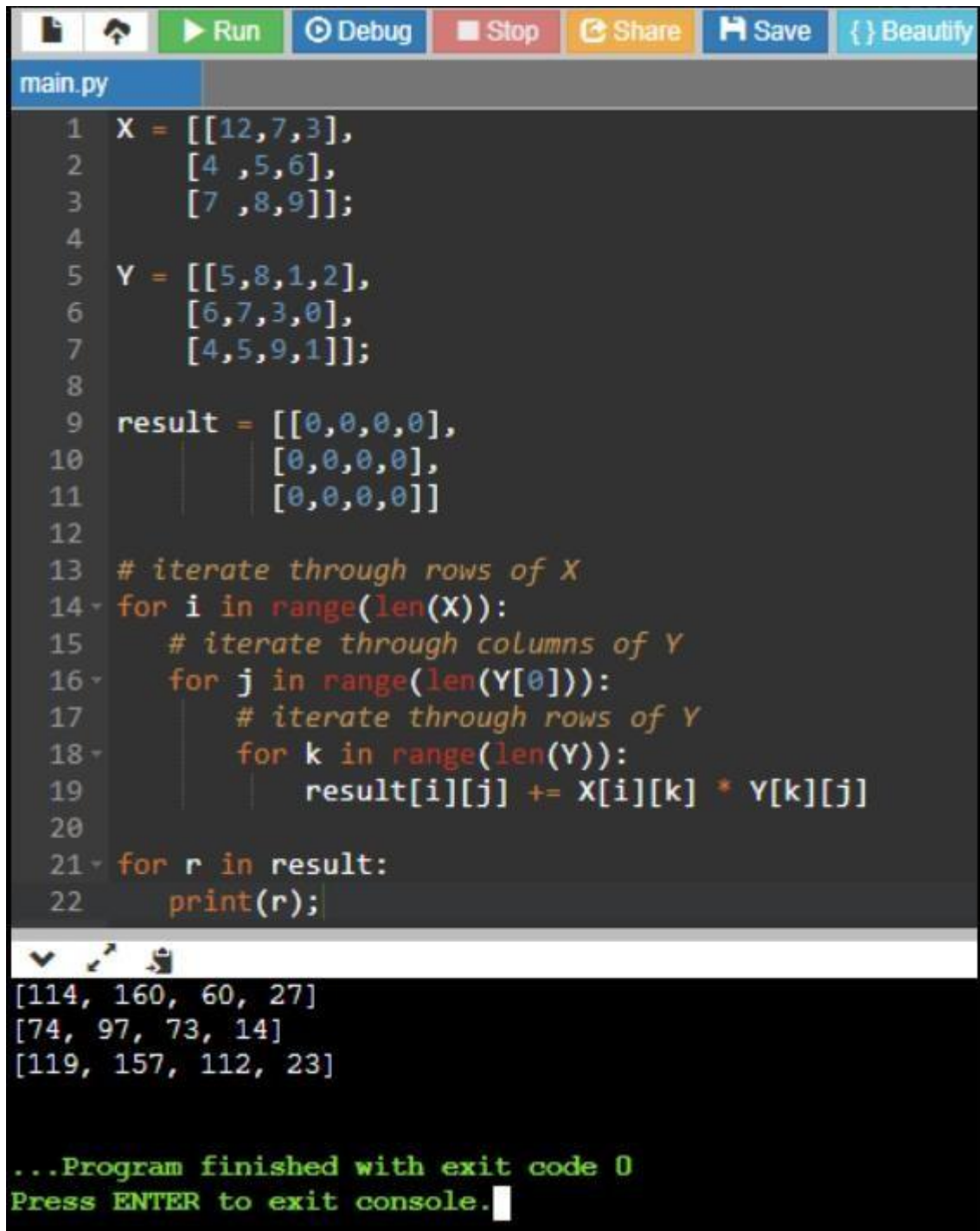
The screenshot shows a Python IDE interface. At the top, there is a toolbar with buttons for Run, Debug, Stop, Share, Save, Beautify, and a download icon. Below the toolbar, the file name 'main.py' is displayed. The code editor contains the following Python code:

```
1 def repeats(list0):
2     for i in range(0, len(list0) - 1):
3         if(list0[i] == list0[i + 1]):
4             return True;
5     return False;
6
7 print(repeats([1, 2, 3, 3]));
```

Below the code editor, there is a console window. The output of the program is displayed as 'True'. At the bottom of the console, it says '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor.

Q4 > Write a Python function to perform matrix multiplication.

Ans>



The screenshot shows a Python IDE window titled 'main.py'. The code defines two matrices, X and Y, and calculates their product. Matrix X is a 3x3 matrix:  $\begin{bmatrix} 12 & 7 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$ . Matrix Y is a 3x4 matrix:  $\begin{bmatrix} 5 & 8 & 1 & 2 \\ 6 & 7 & 3 & 0 \\ 4 & 5 & 9 & 1 \end{bmatrix}$ . The result is a 3x4 matrix calculated using nested loops. The output shows the resulting matrix:  $\begin{bmatrix} 114 & 160 & 60 & 27 \\ 74 & 97 & 73 & 14 \\ 119 & 157 & 112 & 23 \end{bmatrix}$ . The program finishes with exit code 0.

```
1 X = [[12,7,3],
2       [4 ,5,6],
3       [7 ,8,9]];
4
5 Y = [[5,8,1,2],
6       [6,7,3,0],
7       [4,5,9,1]];
8
9 result = [[0,0,0,0],
10           [0,0,0,0],
11           [0,0,0,0]]
12
13 # iterate through rows of X
14 for i in range(len(X)):
15     # iterate through columns of Y
16     for j in range(len(Y[0])):
17         # iterate through rows of Y
18         for k in range(len(Y)):
19             result[i][j] += X[i][k] * Y[k][j]
20
21 for r in result:
22     print(r);
```

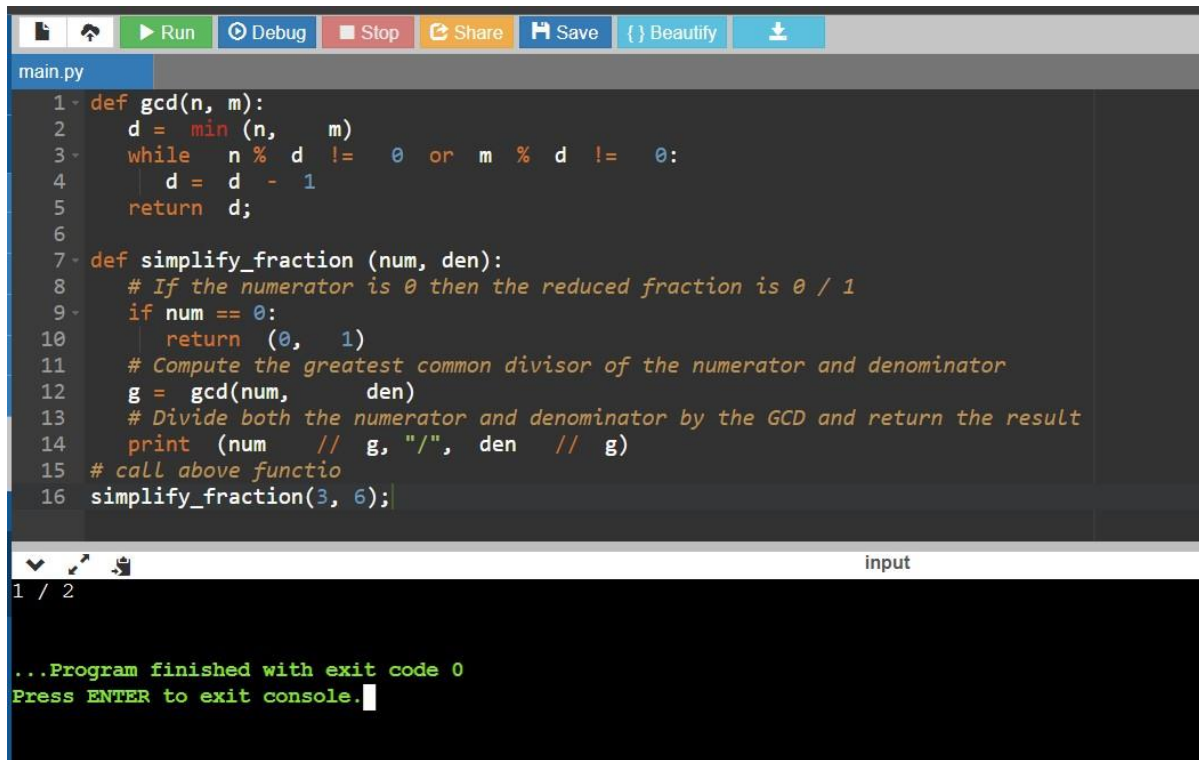
[114, 160, 60, 27]  
[74, 97, 73, 14]  
[119, 157, 112, 23]

...Program finished with exit code 0  
Press ENTER to exit console.

Q5 > Write a function with the signature `simplify_fraction(n, m)` which prints the simplified version of the fraction  $n/m$ .

For example, `simplify_fraction(3,6)` should print  $1/2$  and `simplify_fraction(8, 4)` should print  $2/1$  Hint: use a similar technique to the one we used when determining whether a number is prime. That is, try dividing both the numerator and the denominator by every possible divisor in turn. For example, if you are simplifying  $16/12$ , you can try dividing both by 16, 15, 14, ..., 1.

Ans>



```
main.py
1 def gcd(n, m):
2     d = min(n, m)
3     while n % d != 0 or m % d != 0:
4         d = d - 1
5     return d;
6
7 def simplify_fraction(num, den):
8     # If the numerator is 0 then the reduced fraction is 0 / 1
9     if num == 0:
10        return (0, 1)
11    # Compute the greatest common divisor of the numerator and denominator
12    g = gcd(num, den)
13    # Divide both the numerator and denominator by the GCD and return the result
14    print(num // g, "/", den // g)
15    # call above function
16    simplify_fraction(3, 6);
```

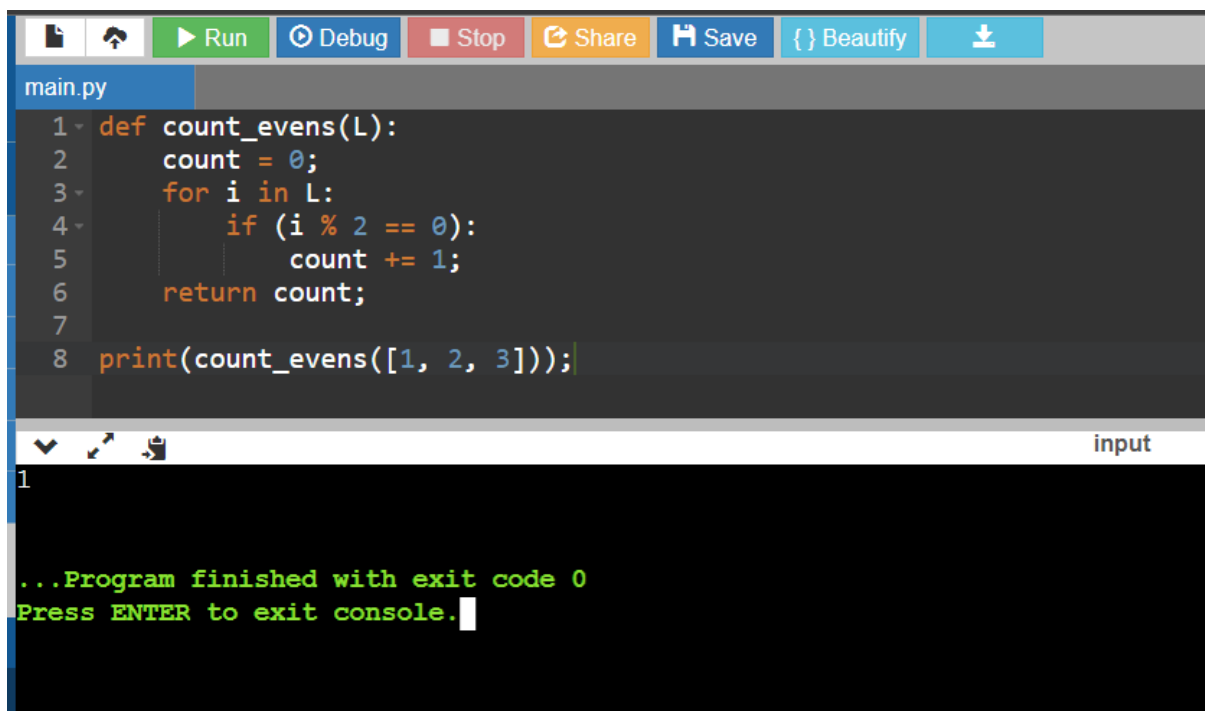
input

1 / 2

...Program finished with exit code 0  
Press ENTER to exit console.

Q6> Write a function with the signature `def count_evens(L)` that returns the number of even integers in the list `L`. Assume `L` only contains integers.

Ans>



The screenshot shows a Python IDE with a toolbar at the top containing icons for file operations, a 'Run' button, 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a download icon. Below the toolbar, a file named 'main.py' is open. The code in the editor is as follows:

```
1 def count_evens(L):
2     count = 0;
3     for i in L:
4         if (i % 2 == 0):
5             count += 1;
6     return count;
7
8 print(count_evens([1, 2, 3]));
```

Below the code editor, there is an 'input' section. It shows the number '1' entered. At the bottom, a green message states: '...Program finished with exit code 0' and 'Press ENTER to exit console.' with a cursor.

Q7> Recall that you can compute the sum  $1+2+\dots+n$  using  $\text{total} = 0$

for  $i$  in  $\text{range}(1, n+1)$ :

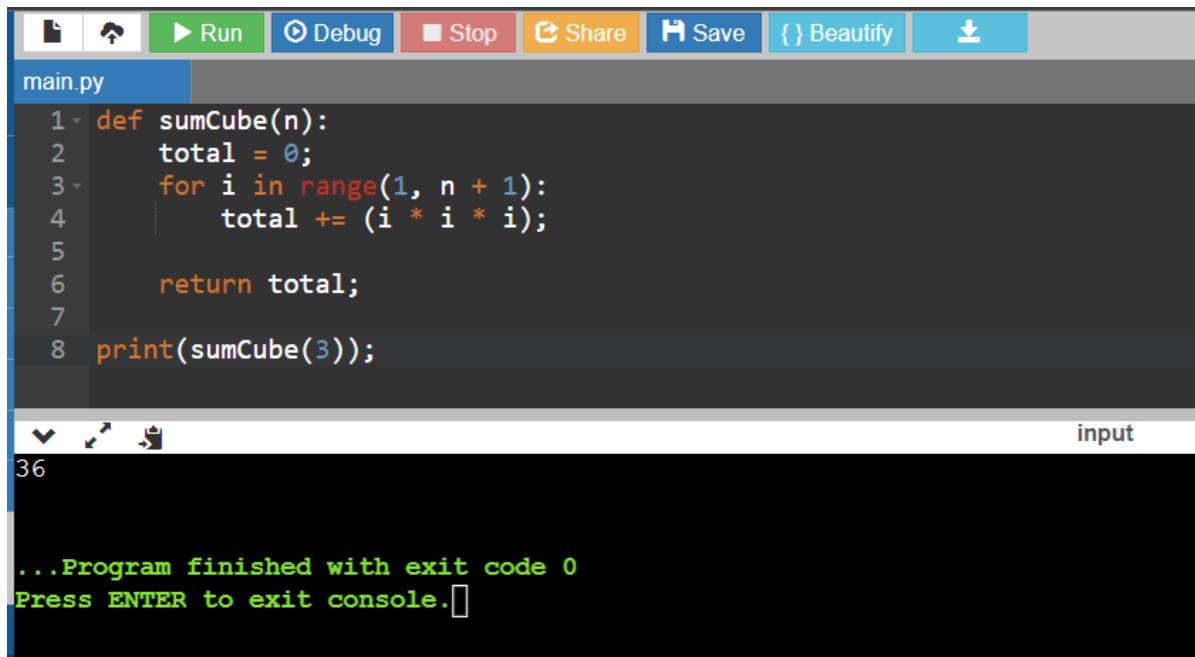
$\text{total} += i$

Write a function that computes  $1^3+2^3+\dots+n^3$  using a loop. Write another function that computes the sum using the formula here.

Now write a function with the signature `def check_sum(n)` that computes the sum of consecutive cubes using a loop and using the formula, and return True iff the results match. (Recall: in this class, iff implies that False is returned if True is not returned.)

Now, write a function with the signature `def check_sums_up_to_n(N)` that checks that the formula works for every  $n \leq N$  and returns True iff for every  $n \leq N$ , the formula works

Ans>



The image shows a screenshot of a Python IDE. The top toolbar contains icons for file operations and buttons for 'Run', 'Debug', 'Stop', 'Share', 'Save', 'Beautify', and a download icon. The editor window, titled 'main.py', contains the following Python code:

```
1 def sumCube(n):
2     total = 0;
3     for i in range(1, n + 1):
4         total += (i * i * i);
5
6     return total;
7
8 print(sumCube(3));
```

Below the editor is a console window with a dark background. It shows the output '36' and a message: '...Program finished with exit code 0' followed by 'Press ENTER to exit console.' with a cursor.