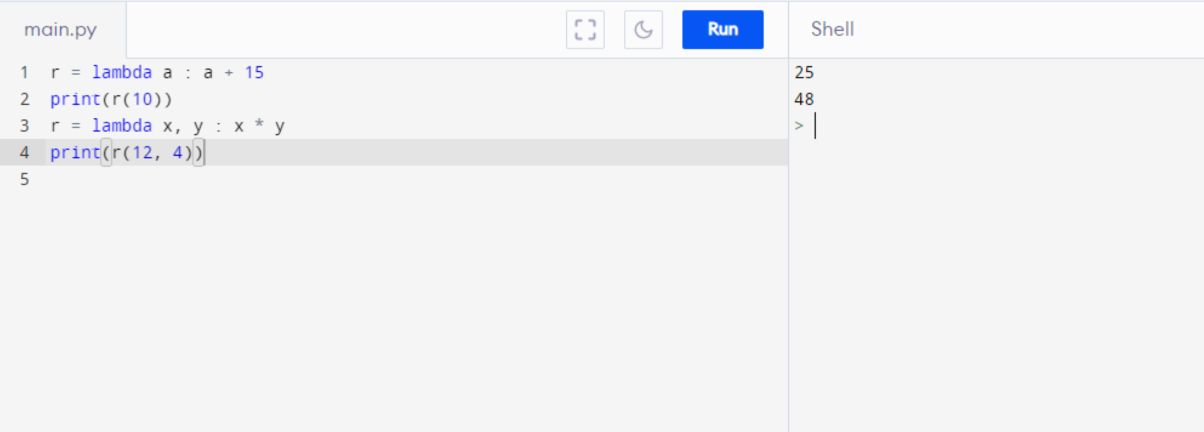
# PYTHON LAMBDA ASSIGNMENT

1) Write a Python program to create a lambda function that adds 15 to a given number passed in as an argument, also create a lambda function that multiplies argument x with argument y and prints the result.

**PROGRAM:**

```
 r = lambda a : a + 15
print(r(10))
r = lambda x, y : x * y
print(r(12, 4))
```

**OUTPUT:**



```
r = lambda a : a + 15
print(r(10))
r = lambda x, y : x * y
print(r(12, 4))
```

```
25
48
>
```

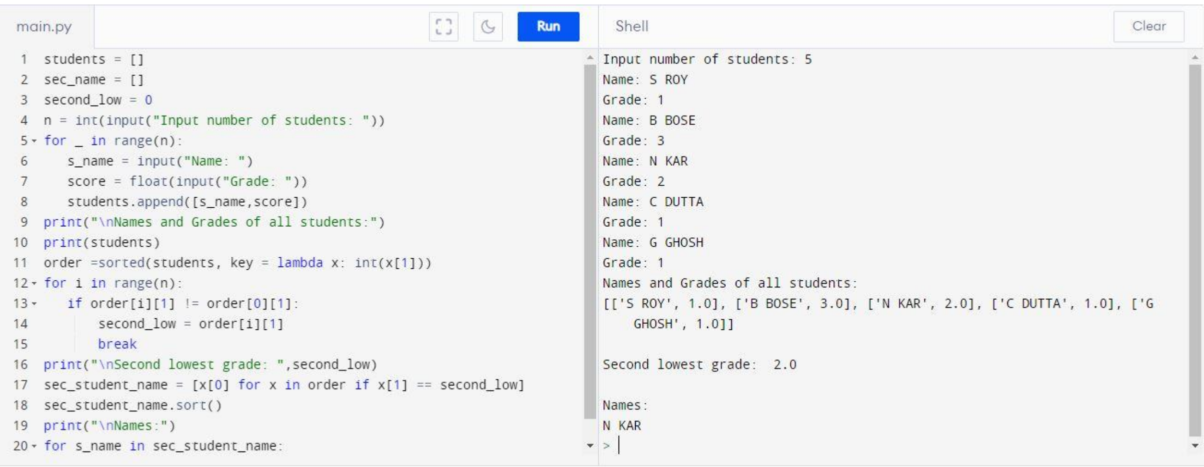
2) Write a Python program to find the second lowest total marks of any student(s) from the given names and marks of each student using lists and lambda. Input the number of students, the names and grades of each student.

**PROGRAM:**

```
    students = []
    sec_name = []
    second_low = 0
    n = int(input("Input number of students: "))
    for _ in range(n):
   s_name = input("Name: ")
  score = float(input("Grade: "))
  students.append([s_name,score])
  print("\nNames and Grades of all students:")
  print(students)
  order =sorted(students, key = lambda x: int(x[1]))
  for i in range(n):
 if order[i][1] != order[0][1]:
 second_low = order[i][1]
 break
print("\nSecond lowest grade: ",second_low)
sec_student_name = [x[0] for x in order if x[1] ==     second_low]
  sec_student_name.sort()
print("\nNames:")
```

```
for s_name in sec_student_name:
 print(s_name)
```

**OUTPUT:**



```
students = []
sec_name = []
second_low = 0
n = int(input("Input number of students: "))
for _ in range(n):
   s_name = input("Name: ")
   score = float(input("Grade: "))
   students.append([s_name,score])
print("\nNames and Grades of all students:")
print(students)
order =sorted(students, key = lambda x: int(x[1]))
for i in range(n):
   if order[i][1] != order[0][1]:
       second_low = order[i][1]
       break
print("\nSecond lowest grade: ",second_low)
sec_student_name = [x[0] for x in order if x[1] == second_low]
sec_student_name.sort()
print("\nNames:")
for s_name in sec_student_name:
```

```
Input number of students: 5
Name: S ROY
Grade: 1
Name: B BOSE
Grade: 3
Name: N KAR
Grade: 2
Name: C DUTTA
Grade: 1
Name: G GHOSH
Grade: 1
Names and Grades of all students:
[['S ROY', 1.0], ['B BOSE', 3.0], ['N KAR', 2.0], ['C DUTTA', 1.0], ['G
    GHOSH', 1.0]]

Second lowest grade:  2.0

Names:
N KAR
> |
```

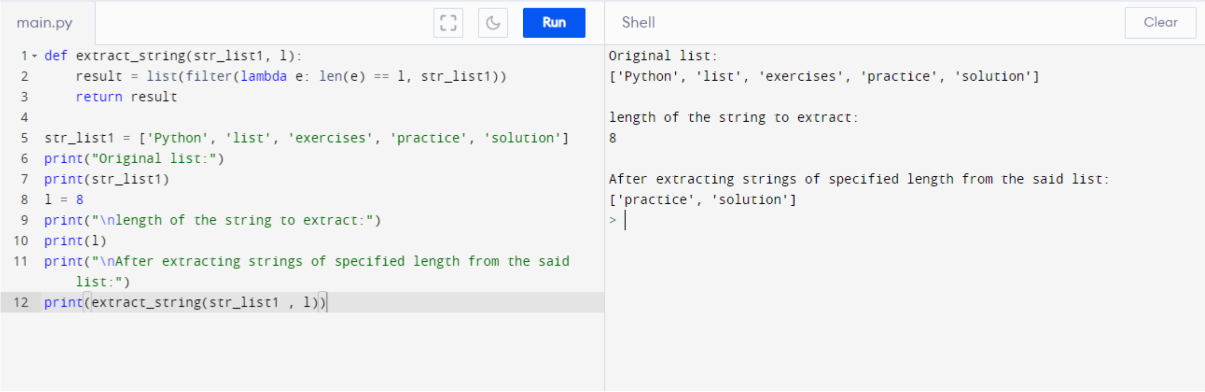
3) Write a Python program to extract a specified size of strings from a given list of string values using lambda
4) Original list:
   ['Python','list','exercises','practice','solution']
   length of the string to extract:
   8
   After extracting strings of specified length from the said list:
   [&#39;practice&#39;, &#39;solution&#39;]

**PROGRAM:**

```
def extract_string(str_list1, l):
result = list(filter(lambda e: len(e) == l, str_list1))
return result

str_list1 = ['Python', 'list', 'exercises', 'practice', 'solution']
print("Original list:")
print(str_list1)
l = 8
print("\nlength of the string to extract:")
print(l)
print("\nAfter extracting strings of specified length from the said list:")
print(extract_string(str_list1 , l))
```

**OUTPUT:**



```
def extract_string(str_list1, l):
    result = list(filter(lambda e: len(e) == l, str_list1))
    return result

str_list1 = ['Python', 'list', 'exercises', 'practice', 'solution']
print("Original list:")
print(str_list1)
l = 8
print("\nlength of the string to extract:")
print(l)
print("\nAfter extracting strings of specified length from the said list:")
print(extract_string(str_list1 , l))
```

```
Original list:
['Python', 'list', 'exercises', 'practice', 'solution']

length of the string to extract:
8

After extracting strings of specified length from the said list:
['practice', 'solution']
>
```

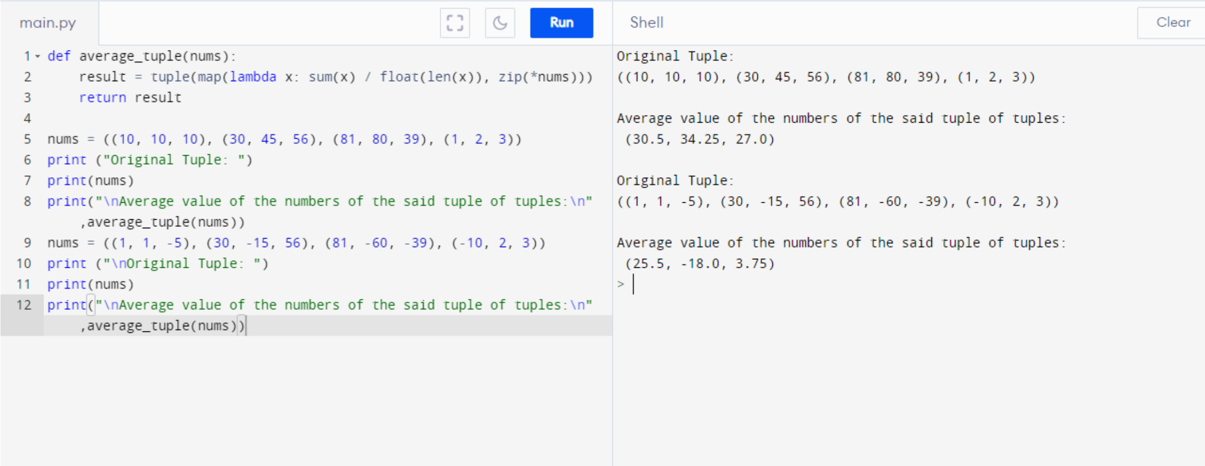
5) Write a Python program to calculate the average value of the numbers in a given tuple of tuples using lambda.

**PROGRAM:**

```
def average_tuple(nums):
result = tuple(map(lambda x: sum(x) / float(len(x)), zip(*nums)))
return result

nums = ((10, 10, 10), (30, 45, 56), (81, 80, 39), (1, 2, 3))
print ("Original Tuple: ")
print(nums)
print("\nAverage value of the numbers of the said tuple of tuples:\n",average_tuple(nums))
nums = ((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))
print ("\nOriginal Tuple: ")
print(nums)
print("\nAverage value of the numbers of the said tuple of tuples:\n",average_tuple(nums))
```

**OUTPUT:**



```
def average_tuple(nums):
    result = tuple(map(lambda x: sum(x) / float(len(x)), zip(*nums)))
    return result

nums = ((10, 10, 10), (30, 45, 56), (81, 80, 39), (1, 2, 3))
print ("Original Tuple: ")
print(nums)
print("\nAverage value of the numbers of the said tuple of tuples:\n",average_tuple(nums))
nums = ((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))
print ("\nOriginal Tuple: ")
print(nums)
print("\nAverage value of the numbers of the said tuple of tuples:\n",average_tuple(nums))
```

```
Original Tuple:
((10, 10, 10), (30, 45, 56), (81, 80, 39), (1, 2, 3))

Average value of the numbers of the said tuple of tuples:
 (30.5, 34.25, 27.0)

Original Tuple:
((1, 1, -5), (30, -15, 56), (81, -60, -39), (-10, 2, 3))

Average value of the numbers of the said tuple of tuples:
 (25.5, -18.0, 3.75)
>
```