

NAME – RAJDEEP JAISWAL

UID – 20BCS2761

BRANCH – B.TECH CSE

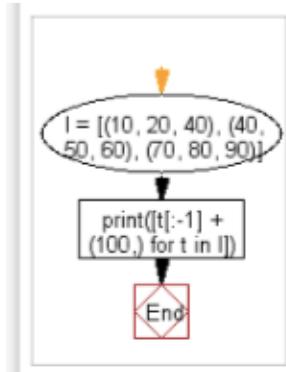
SEC- 615 B

SUBJECT – PYTHON LAB

DATE OF SUBMISSION -

1. Q1 - Write a Python program to replace last value of tuples in a list

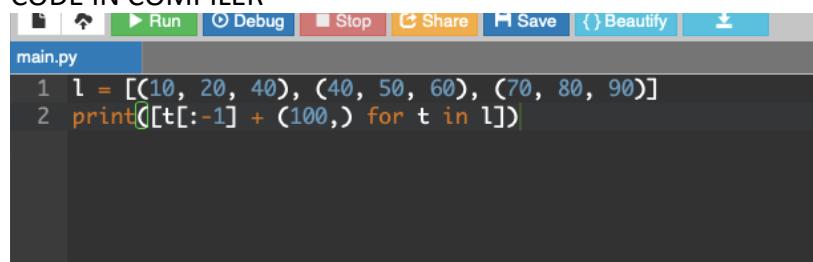
FLOWCHART-



CODE IN TEXT FORM –

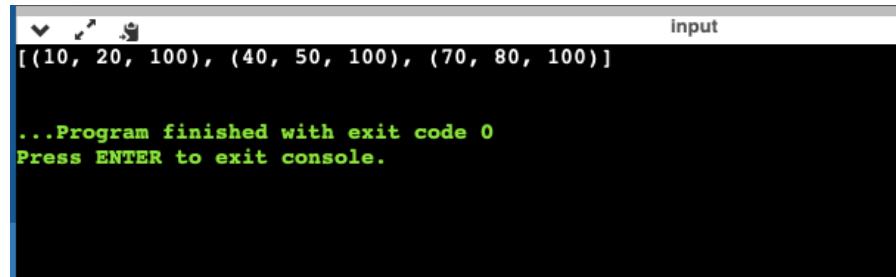
```
l = [(10, 20, 40), (40, 50, 60), (70, 80, 90)]  
print([t[:-1] + (100,) for t in l])
```

CODE IN COMPILER –



```
main.py  
1 l = [(10, 20, 40), (40, 50, 60), (70, 80, 90)]  
2 print([t[:-1] + (100,) for t in l])
```

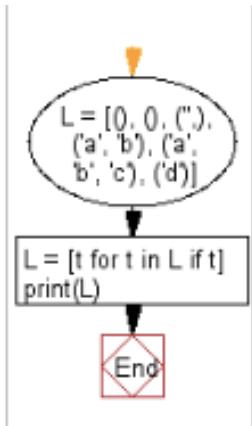
OUTPUT-



```
[(10, 20, 100), (40, 50, 100), (70, 80, 100)]  
...Program finished with exit code 0  
Press ENTER to exit console.
```

2. Write a Python program to remove an empty tuple(s) from a list of tuples

FLOWCHART -



CODE IN TEXT -

```
L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]
L = [t for t in L if t]
print(L)
```

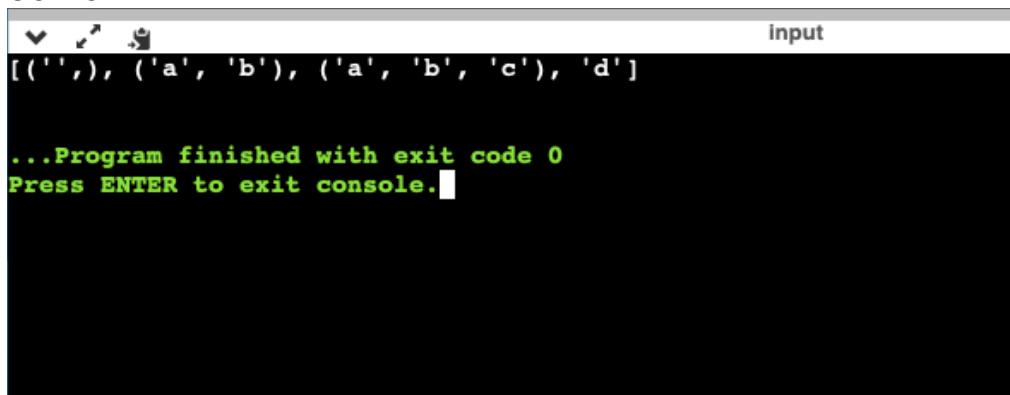
CODE IN COMPILER -



```

main.py
1 L = [(), (), ('',), ('a', 'b'), ('a', 'b', 'c'), ('d')]
2 L = [t for t in L if t]
3 print(L)
  
```

OUTPUT -



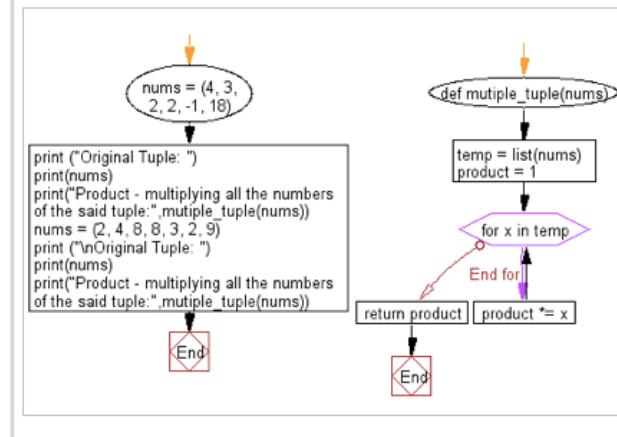
```

[(''), ('a', 'b'), ('a', 'b', 'c'), 'd']

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

Q3 Write a Python program calculate the product, multiplying all the numbers of a given tuple

FLOW CHART-



CODE IN TEXT FORM –

```

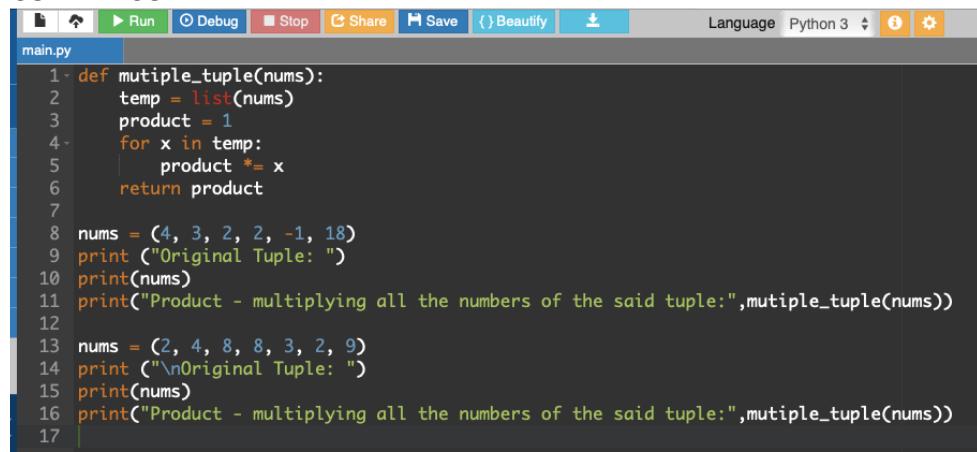
def mutiple_tuple(nums):
    temp = list(nums)
    product = 1
    for x in temp:
        product *= x
    return product

nums = (4, 3, 2, 2, -1, 18)
print ("Original Tuple: ")
print(nums)
print("Product - multiplying all the numbers of the said tuple:", mutiple_tuple(nums))

nums = (2, 4, 8, 8, 3, 2, 9)
print ("\nOriginal Tuple: ")
print(nums)
print("Product - multiplying all the numbers of the said tuple:", mutiple_tuple(nums))

```

CODE IN COMPILER-



```

main.py
1 def mutiple_tuple(nums):
2     temp = list(nums)
3     product = 1
4     for x in temp:
5         product *= x
6     return product
7
8 nums = (4, 3, 2, 2, -1, 18)
9 print ("Original Tuple: ")
10 print(nums)
11 print("Product - multiplying all the numbers of the said tuple:", mutiple_tuple(nums))
12
13 nums = (2, 4, 8, 8, 3, 2, 9)
14 print ("\nOriginal Tuple: ")
15 print(nums)
16 print("Product - multiplying all the numbers of the said tuple:", mutiple_tuple(nums))
17

```

OUTPUT –

```

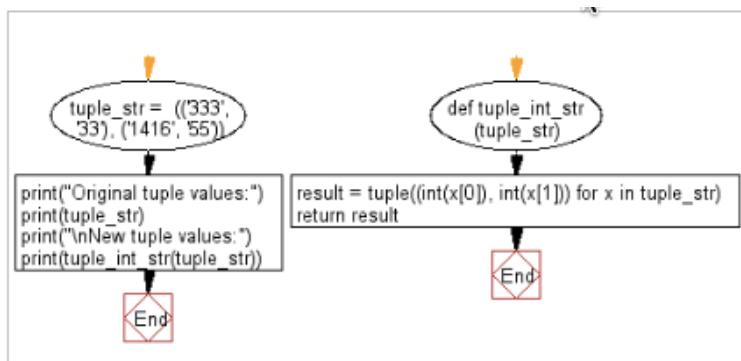
input
Original Tuple:
(4, 3, 2, 2, -1, 18)
Product - multiplying all the numbers of the said tuple: -864

Original Tuple:
(2, 4, 8, 8, 3, 2, 9)
Product - multiplying all the numbers of the said tuple: 27648

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

```

Q4 - Write a Python program to convert a tuple of string values to a tuple of integer values.

FLOWCHART-

CODE IN TEXT –

```

def tuple_int_str(tuple_str):
    result = tuple((int(x[0]), int(x[1])) for x in tuple_str)
    return result

tuple_str = (('333', '33'), ('1416', '55'))
print("Original tuple values:")
print(tuple_str)
print("\nNew tuple values:")
print(tuple_int_str(tuple_str))

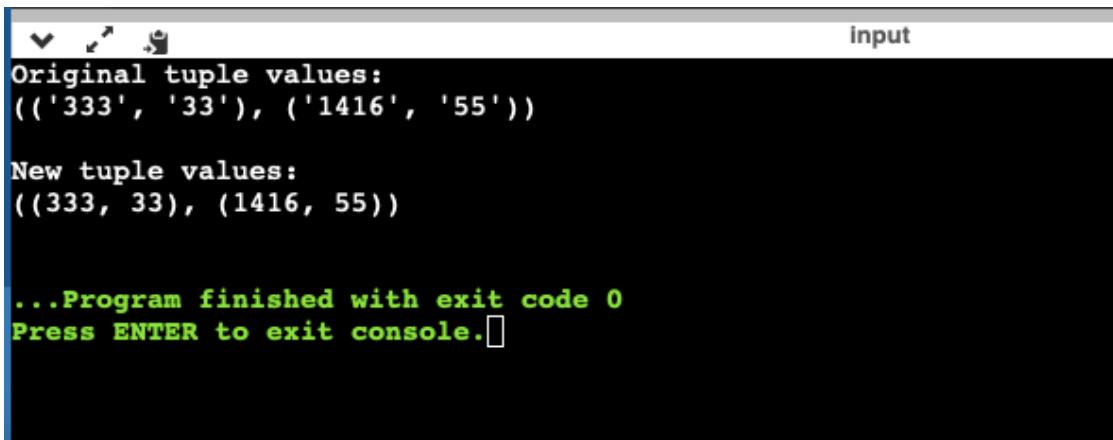
```

CODE IN COMPILER –



```
main.py
1 def tuple_int_str(tuple_str):
2     result = tuple((int(x[0]), int(x[1])) for x in tuple_str)
3     return result
4
5 tuple_str = (('333', '33'), ('1416', '55'))
6 print("Original tuple values:")
7 print(tuple_str)
8 print("\nNew tuple values:")
9 print(tuple_int_str(tuple_str))
10 |
```

OUTPUT –



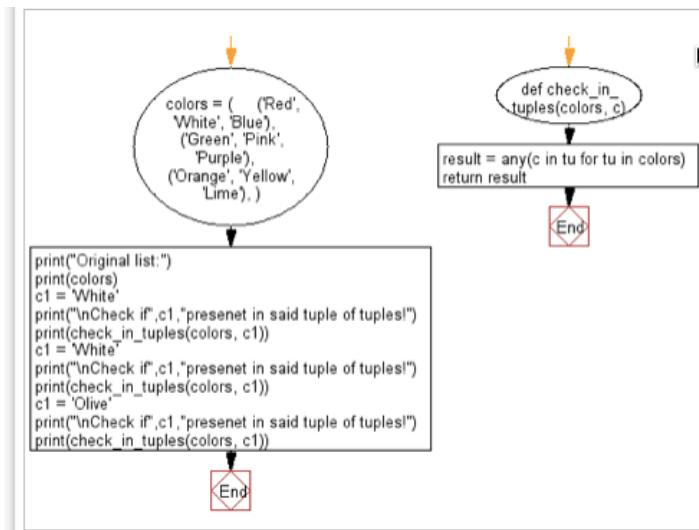
```
input
Original tuple values:
(('333', '33'), ('1416', '55'))

New tuple values:
((333, 33), (1416, 55))

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

1. Q5 – Write a Python program to check if a specified element presents in a tuple of tuples

FLOWCHART –



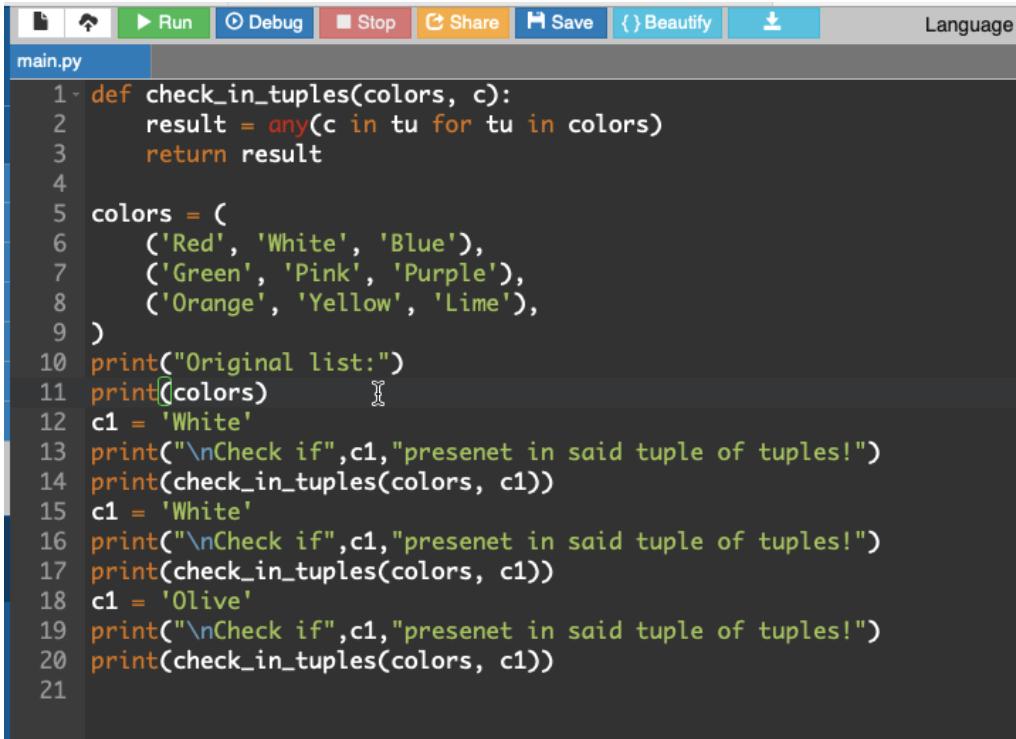
CODE IN TEXT FORM –

```

def check_in_tuples(colors, c):
    result = any(c in tu for tu in colors)
    return result

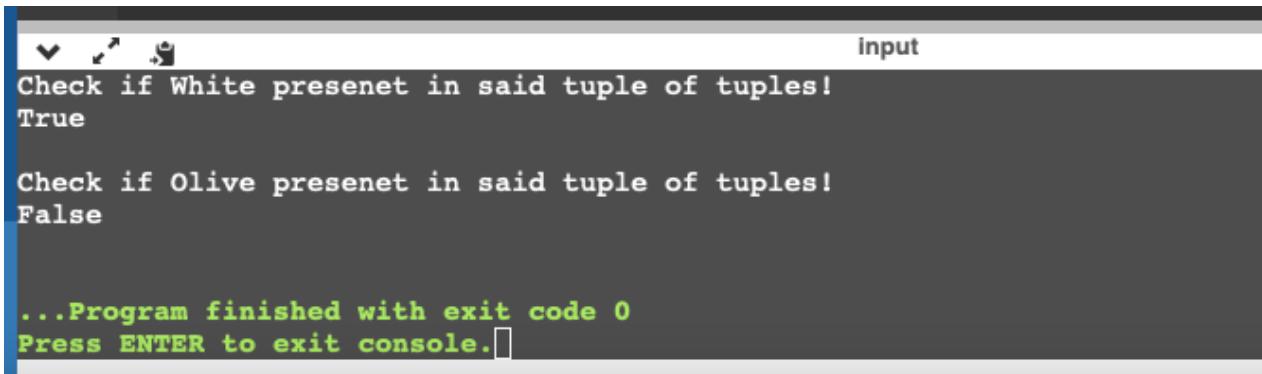
colors = (
    ('Red', 'White', 'Blue'),
    ('Green', 'Pink', 'Purple'),
    ('Orange', 'Yellow', 'Lime'),
)
print("Original list:")
print(colors)
c1 = 'White'
print("\nCheck if",c1,"presenet in said tuple of tuples!")
print(check_in_tuples(colors, c1))
c1 = 'White'
print("\nCheck if",c1,"presenet in said tuple of tuples!")
print(check_in_tuples(colors, c1))
c1 = 'Olive'
print("\nCheck if",c1,"presenet in said tuple of tuples!")
print(check_in_tuples(colors, c1))
    
```

CODE IN COMPILER-



```
main.py
1 def check_in_tuples(colors, c):
2     result = any(c in tu for tu in colors)
3     return result
4
5 colors = (
6     ('Red', 'White', 'Blue'),
7     ('Green', 'Pink', 'Purple'),
8     ('Orange', 'Yellow', 'Lime'),
9 )
10 print("Original list:")
11 print(colors)
12 c1 = 'White'
13 print("\nCheck if",c1,"presenet in said tuple of tuples!")
14 print(check_in_tuples(colors, c1))
15 c1 = 'White'
16 print("\nCheck if",c1,"presenet in said tuple of tuples!")
17 print(check_in_tuples(colors, c1))
18 c1 = 'Olive'
19 print("\nCheck if",c1,"presenet in said tuple of tuples!")
20 print(check_in_tuples(colors, c1))
21
```

OUTPUT –



```
input
Check if White presenet in said tuple of tuples!
True

Check if Olive presenet in said tuple of tuples!
False

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