SRM
INSTITUTE OF SCIENCE & TECHNOLOGY
(Deemed to be University u/s 3 of UGC Act, 1956)

Department of Computing Technologies

SRM IST, Kattankulathur – 603 203

Course Code: 18CSC207J

Course Name: Advanced Programming Practice

Experiment No	2
Title of Experiment	To complete all the 13 problems in Jupyter environment
Name of the Student	**Abhijay Rajvansh** (RA2011003010398)
Date of Experiment	28 - 03 - 2022

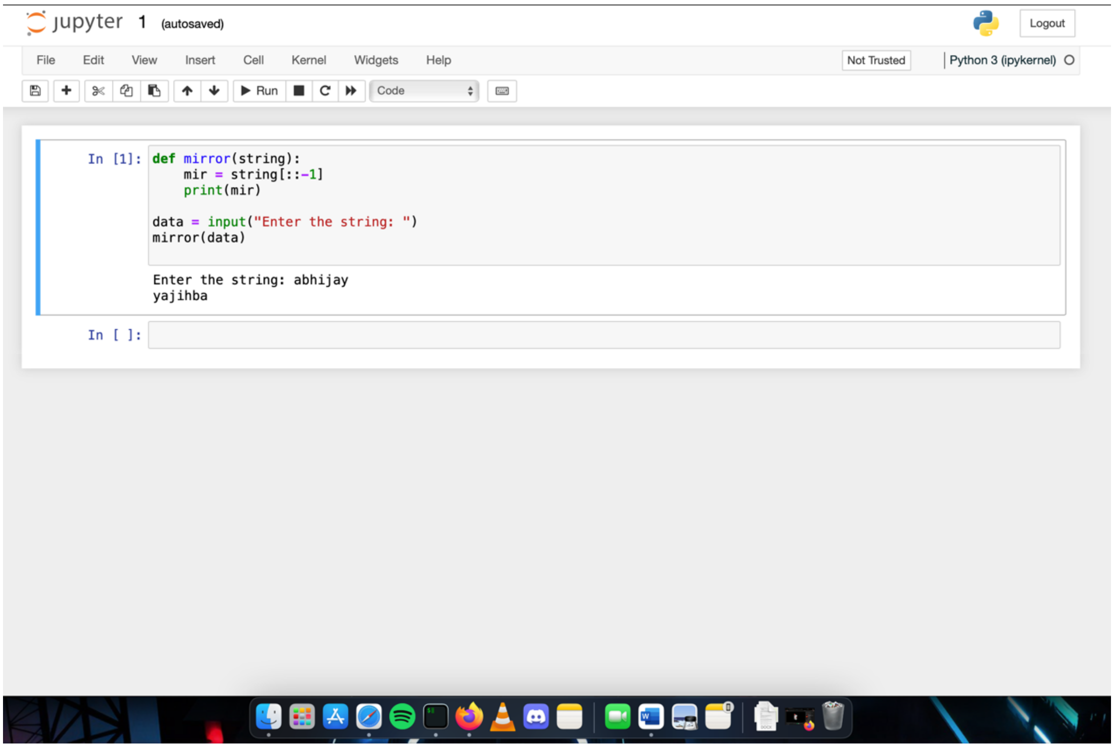
Staff Signature with date

1. AIM: Given a string, find its mirroring image

Code:

```
def mirror(string):
    mir = string[::-1]
    print(mir)

data = input("Enter the string: ")
mirror(data)
```



```
In [1]: def mirror(string):
            mir = string[::-1]
            print(mir)

        data = input("Enter the string: ")
        mirror(data)

        Enter the string: abhijay
        yajihba

In [ ]:
```

Result: Python program to mirror a given string was completed.

2. AIM:Check if two strings are Rotationally Equivalent

```
Sample Output
string 1 is : srmist
 string 2 is : tsrmis
Are two strings Rotationally equal ? : True
```

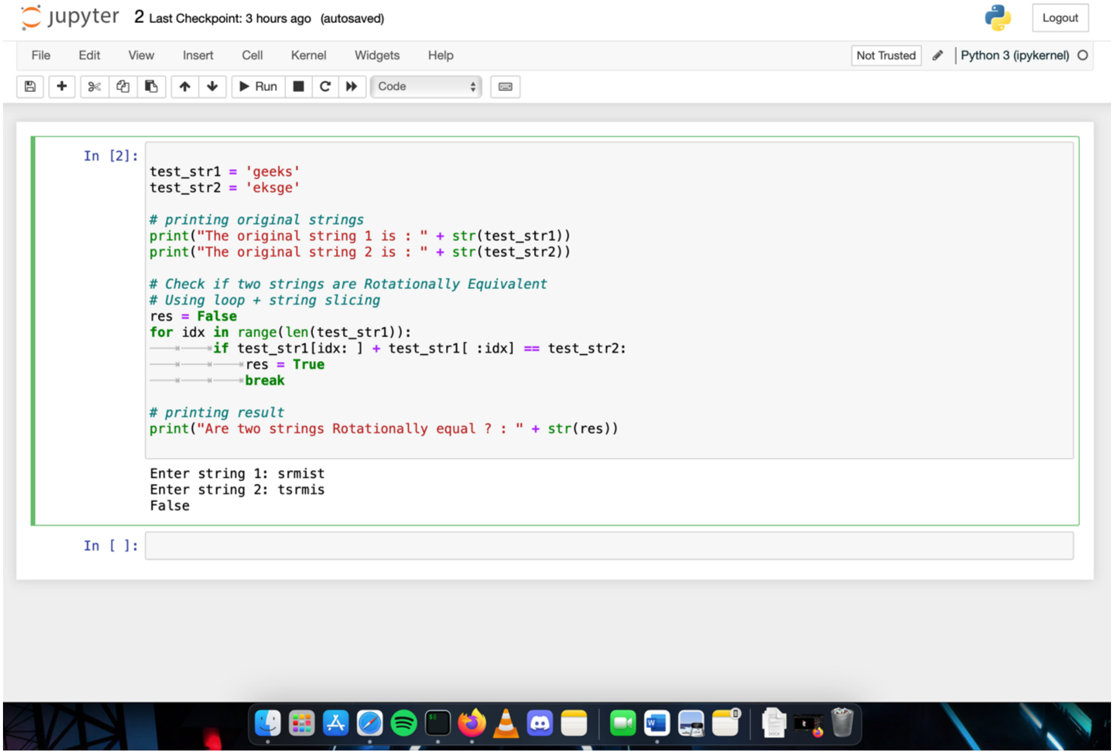
Code:

```
test_str1 = 'geeks'
test_str2 = 'eksge'

# printing original strings
print("The original string 1 is : " + str(test_str1))
print("The original string 2 is : " + str(test_str2))

# Check if two strings are Rotationally Equivalent
# Using loop + string slicing
res = False
for idx in range(len(test_str1)):
        if test_str1[idx: ] + test_str1[ :idx] == test_str2:
                res = True
                break

# printing result
print("Are two strings Rotationally equal ? : " + str(res))
```


jupyter 2 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

In [2]:

```
test_str1 = 'geeks'
test_str2 = 'eksge'

# printing original strings
print("The original string 1 is : " + str(test_str1))
print("The original string 2 is : " + str(test_str2))

# Check if two strings are Rotationally Equivalent
# Using loop + string slicing
res = False
for idx in range(len(test_str1)):
        if test_str1[idx: ] + test_str1[ :idx] == test_str2:
                res = True
                break

# printing result
print("Are two strings Rotationally equal ? : " + str(res))
```

```
Enter string 1: srmist
Enter string 2: tsrmis
False
```

In [ ]:

Result: Python Program to check if two strings are Rotationally Equivalent was completed.

3. AIM: Given a number n, the task is to generate a random binary string of length n.

CODE:

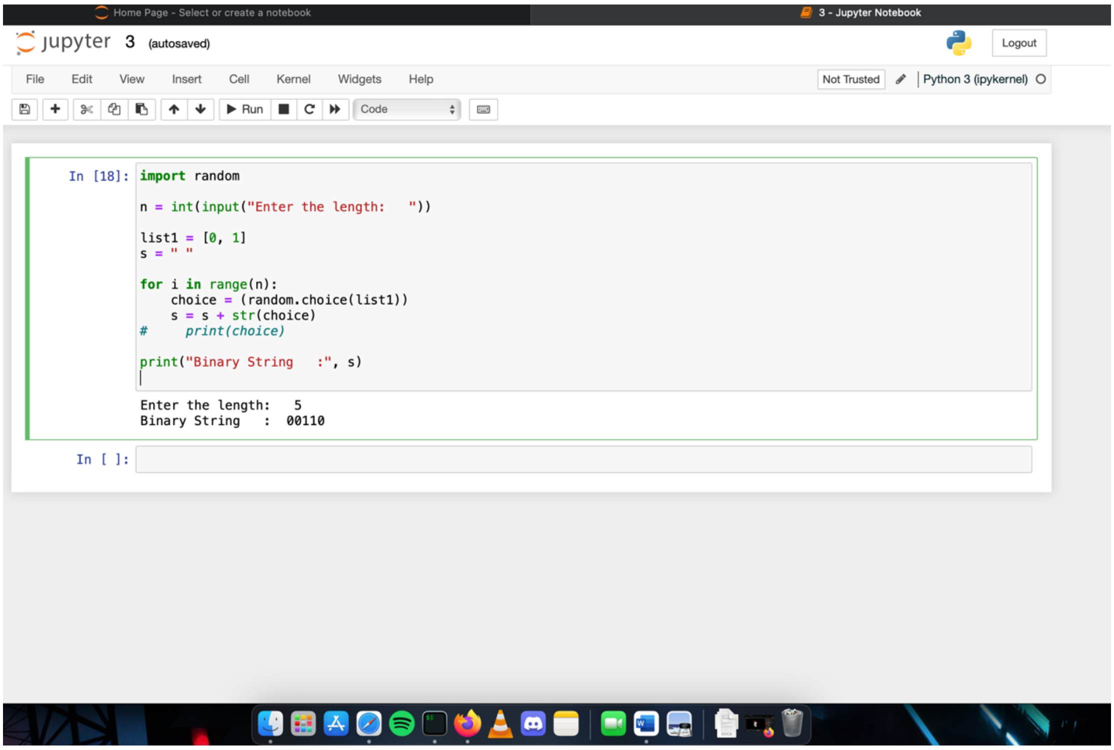
```
import random

n = int(input("Enter the length:   "))

list1 = [0, 1]
s = " "

for i in range(n):
    choice = (random.choice(list1))
    s = s + str(choice)
#    print(choice)

print("Binary String   :", s)
```



Home Page - Select or create a notebook
3 - Jupyter Notebook
jupyter 3 (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [18]: import random

         n = int(input("Enter the length:   "))

         list1 = [0, 1]
         s = " "

         for i in range(n):
             choice = (random.choice(list1))
             s = s + str(choice)
         #     print(choice)

         print("Binary String   :", s)

         Enter the length:   5
         Binary String   :  00110

In [ ]:
```

RESULT: Python program to generate a random binary string of length n was completed.

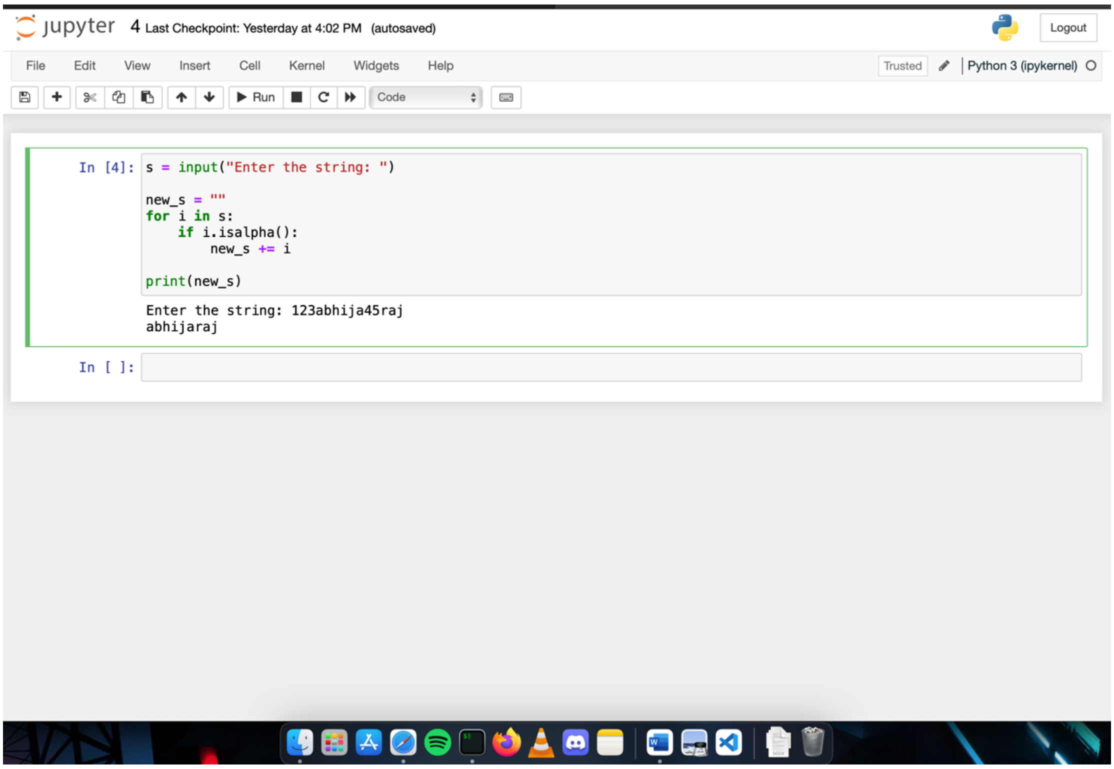
4. AIM: Given a string, remove punctuation and any special characters

CODE

```
s = input("Enter the string: ")

new_s = ""
for i in s:
    if i.isalpha():
        new_s += i

print(new_s)
```



jupyter 4 Last Checkpoint: Yesterday at 4:02 PM (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Trusted
Python 3 (ipykernel)
Run
Code

```
In [4]: s = input("Enter the string: ")

        new_s = ""
        for i in s:
            if i.isalpha():
                new_s += i

        print(new_s)

        Enter the string: 123abhija45raj
        abhijaraj

In [ ]:
```

RESULT: Python program to remove punctuation and any special characters for a given string was completed.

5. AIM: Write a Python program to compute element-wise sum of given tuples.

Input
(11, 2, 3, 14)
(13, 5, 22, 10)
(12, 2, 3, 10)
Output
(36, 9, 28, 34)

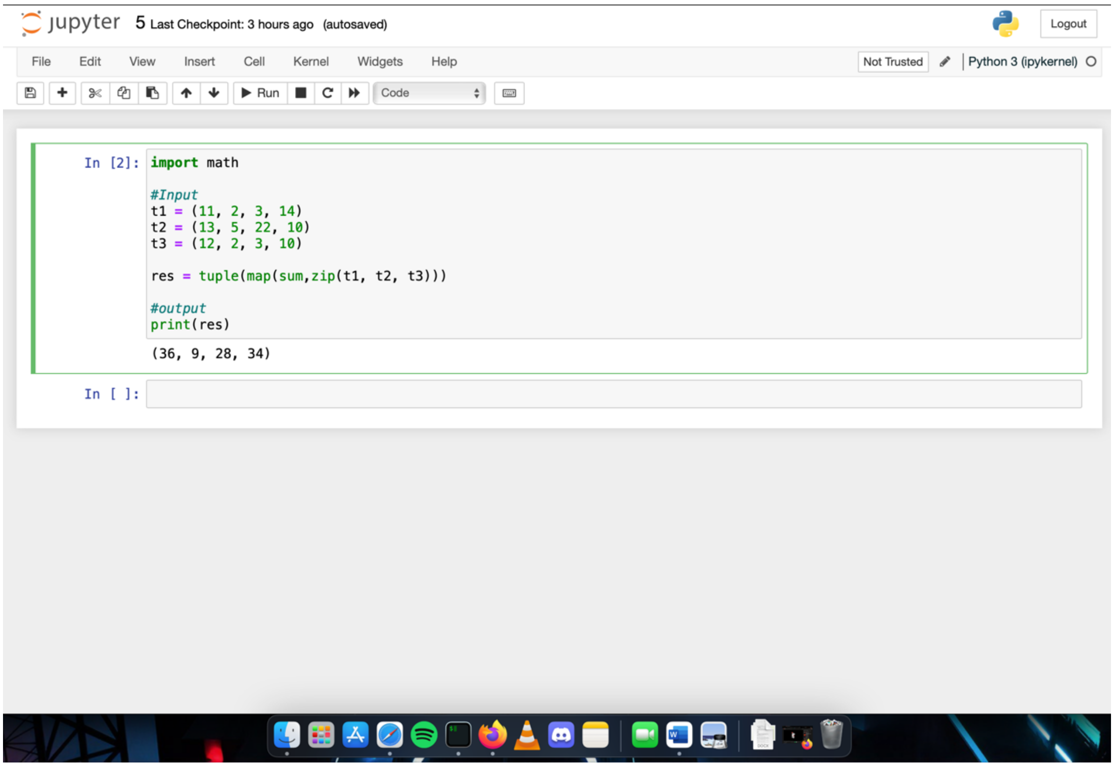
CODE:

```
import math

#Input
t1 = (11, 2, 3, 14)
t2 = (13, 5, 22, 10)
t3 = (12, 2, 3, 10)

res = tuple(map(sum,zip(t1, t2, t3)))

#output
print(res)
```



```
In [2]: import math

        #Input
        t1 = (11, 2, 3, 14)
        t2 = (13, 5, 22, 10)
        t3 = (12, 2, 3, 10)

        res = tuple(map(sum,zip(t1, t2, t3)))

        #output
        print(res)

        (36, 9, 28, 34)

In [ ]:
```

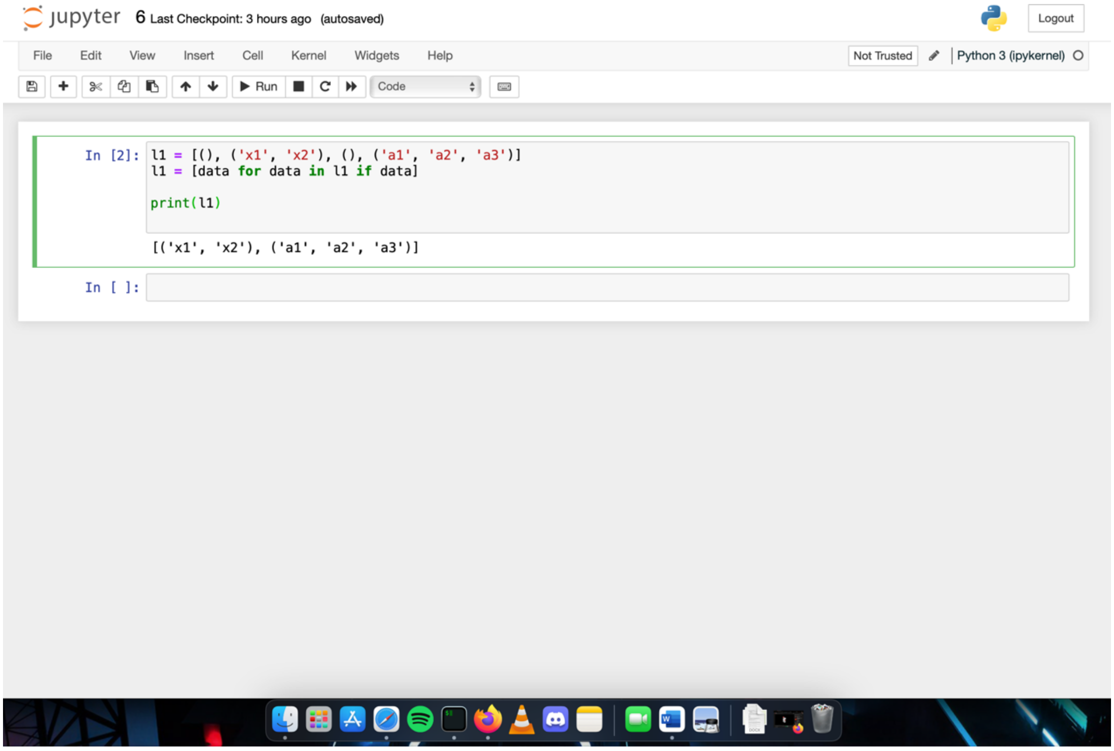
Result: Python program to compute element-wise sum of given tuples was completed.

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

CODE:

```
l1 = [(), ('x1', 'x2'), (), ('a1', 'a2', 'a3')]
l1 = [data for data in l1 if data]

print(l1)
```



```
In [2]: l1 = [(), ('x1', 'x2'), (), ('a1', 'a2', 'a3')]
        l1 = [data for data in l1 if data]

        print(l1)

        [('x1', 'x2'), ('a1', 'a2', 'a3')]

In [ ]:
```

Result: Python program to remove an empty tuple(s) from a list of tuples was completed.

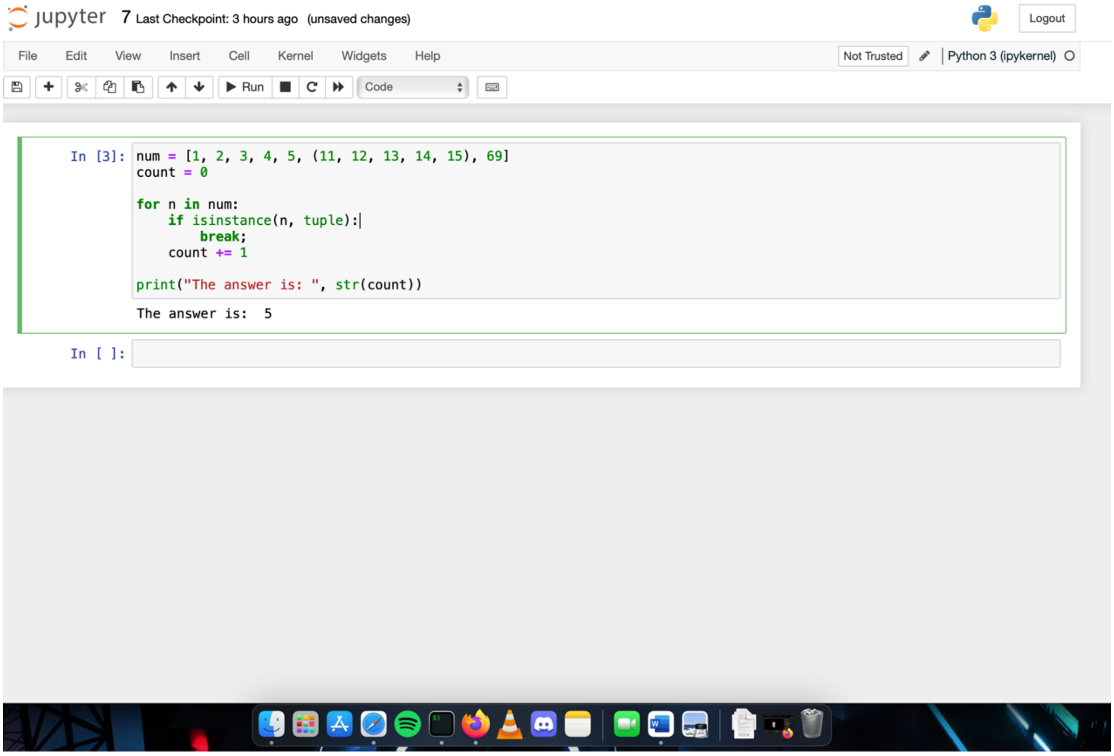
7. AIM: Write a Python program to count the elements in a list until an element is a tuple.

CODE:

```
num = [1, 2, 3, 4, 5, (11, 12, 13, 14, 15), 69]
count = 0

for n in num:
   if isinstance(n, tuple):
      break;
   count += 1

print(count)
```



```
In [3]: num = [1, 2, 3, 4, 5, (11, 12, 13, 14, 15), 69]
        count = 0

        for n in num:
            if isinstance(n, tuple):
                break;
            count += 1

        print("The answer is: ", str(count))

        The answer is:  5

In [ ]:
```

Result: Python program to count the elements in a list until an element is a tuple was completed.

8. AIM: Write a Python program to Convert Tuple Matrix to Tuple List

CODE:

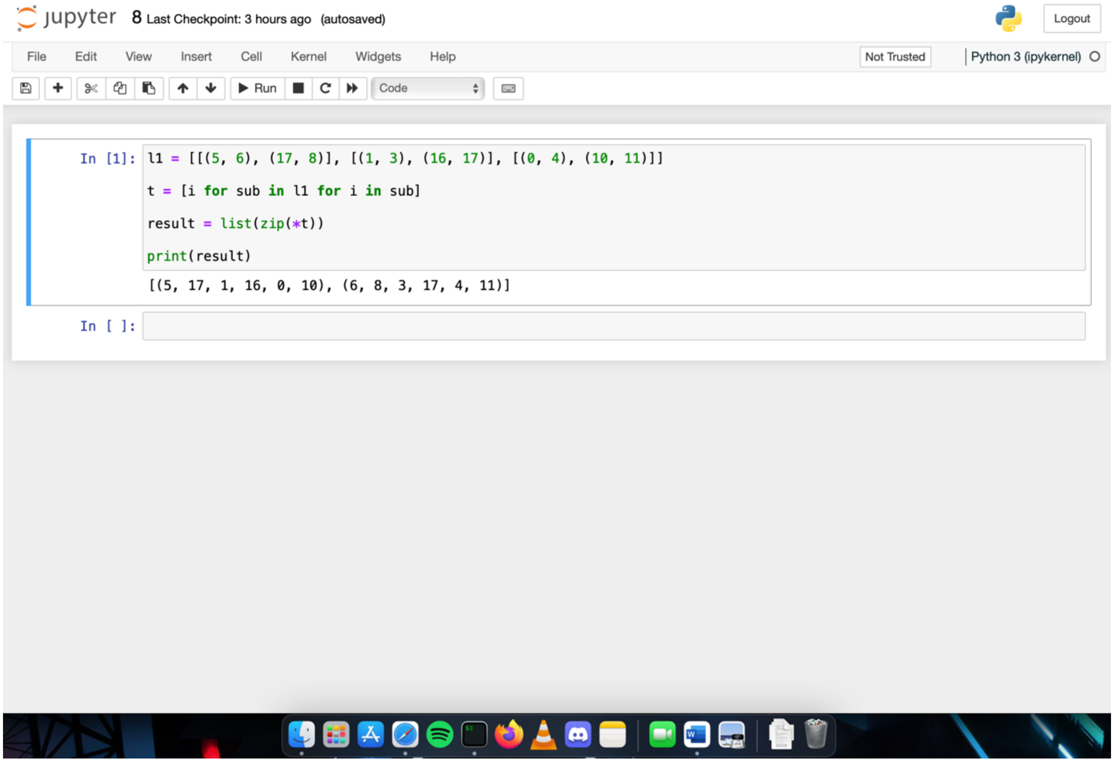
```
l1 = [[(5, 6), (17, 8)], [(1, 3), (16, 17)], [(0, 4), (10, 11)]]

t = [i for sub in l1 for i in sub]

result = list(zip(*t))

print(result)
```

Sample Input : [[(9, 51), (7, 9)], [(11, 1), (22, 19)]]
Output : [(9, 7, 11, 22), (51, 9, 1, 19)]



jupyter 8 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [1]: l1 = [[(5, 6), (17, 8)], [(1, 3), (16, 17)], [(0, 4), (10, 11)]]
        t = [i for sub in l1 for i in sub]
        result = list(zip(*t))
        print(result)
        [(5, 17, 1, 16, 0, 10), (6, 8, 3, 17, 4, 11)]
In [ ]:
```

Result: Python program to Convert Tuple Matrix to Tuple List was completed.

9. AIM: Write a Python program to count unique values in the list.

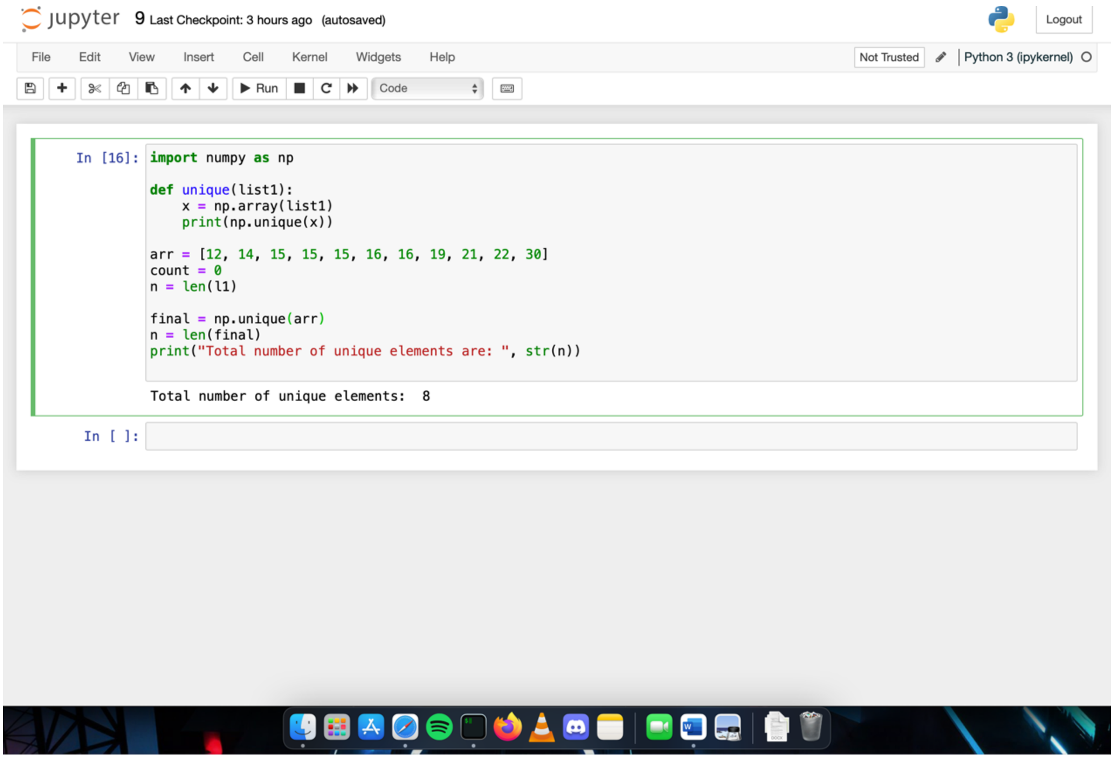
CODE:

```
import numpy as np

def unique(list1):
    x = np.array(list1)
    print(np.unique(x))

arr = [12, 14, 15, 15, 15, 16, 16, 19, 21, 22, 30]
count = 0
n = len(l1)

final = np.unique(arr)
n = len(final)
print("Total number of unique elements are: ", str(n))
```



```
In [16]: import numpy as np

         def unique(list1):
             x = np.array(list1)
             print(np.unique(x))

         arr = [12, 14, 15, 15, 15, 16, 16, 19, 21, 22, 30]
         count = 0
         n = len(l1)

         final = np.unique(arr)
         n = len(final)
         print("Total number of unique elements are: ", str(n))

         Total number of unique elements:  8

In [ ]:
```

Result: Python program to count unique values in the list was completed.

10. AIM: Python Program to print all Possible Combinations from the three Digits

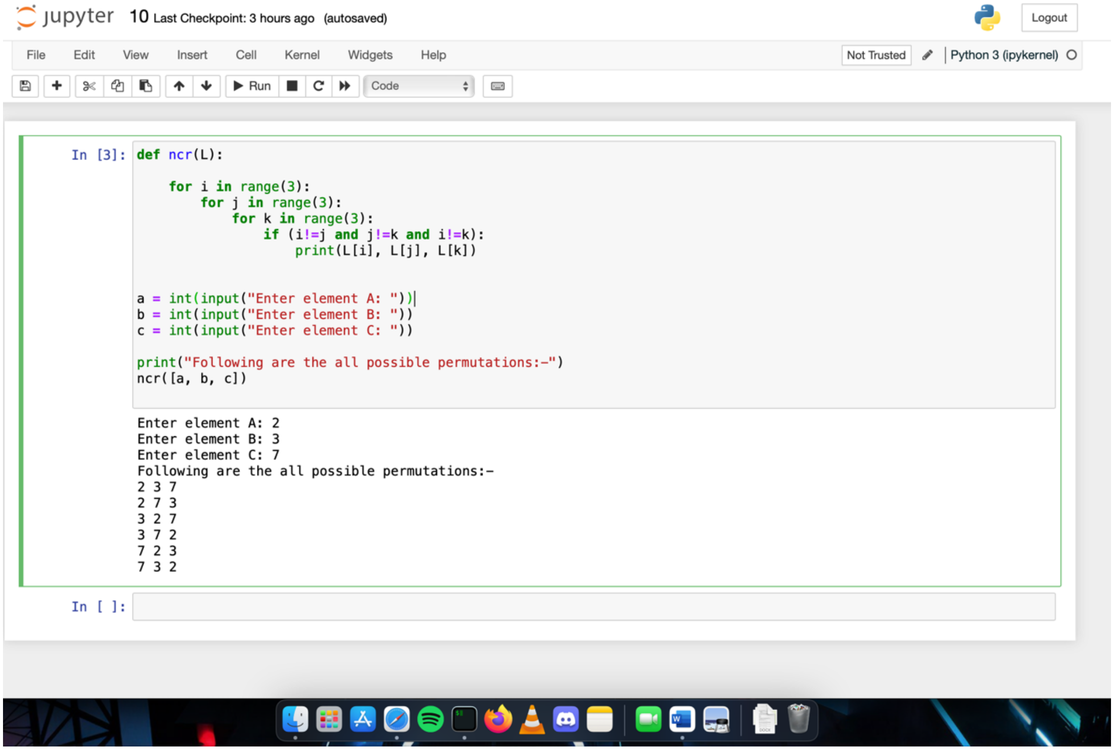
CODE:

```
def ncr(L):

    for i in range(3):
        for j in range(3):
            for k in range(3):
                if (i!=j and j!=k and i!=k):
                    print(L[i], L[j], L[k])


a = int(input("Enter element A: "))
b = int(input("Enter element B: "))
c = int(input("Enter element C: "))

print("Following are the all possible permutations:-")
ncr([a, b, c])
```



jupyter 10 Last Checkpoint: 3 hours ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code

```
In [3]: def ncr(L):

            for i in range(3):
                for j in range(3):
                    for k in range(3):
                        if (i!=j and j!=k and i!=k):
                            print(L[i], L[j], L[k])


        a = int(input("Enter element A: "))
        b = int(input("Enter element B: "))
        c = int(input("Enter element C: "))

        print("Following are the all possible permutations:-")
        ncr([a, b, c])

        Enter element A: 2
        Enter element B: 3
        Enter element C: 7
        Following are the all possible permutations:-
        2 3 7
        2 7 3
        3 2 7
        3 7 2
        7 2 3
        7 3 2

In [ ]:
```

RESULT: Python program to print all Possible Combinations from the three Digits was completed.

11. AIM: Write a Python program (using function) to print the even numbers from a given list.

Note : function type - with arguments but no return values

CODE:

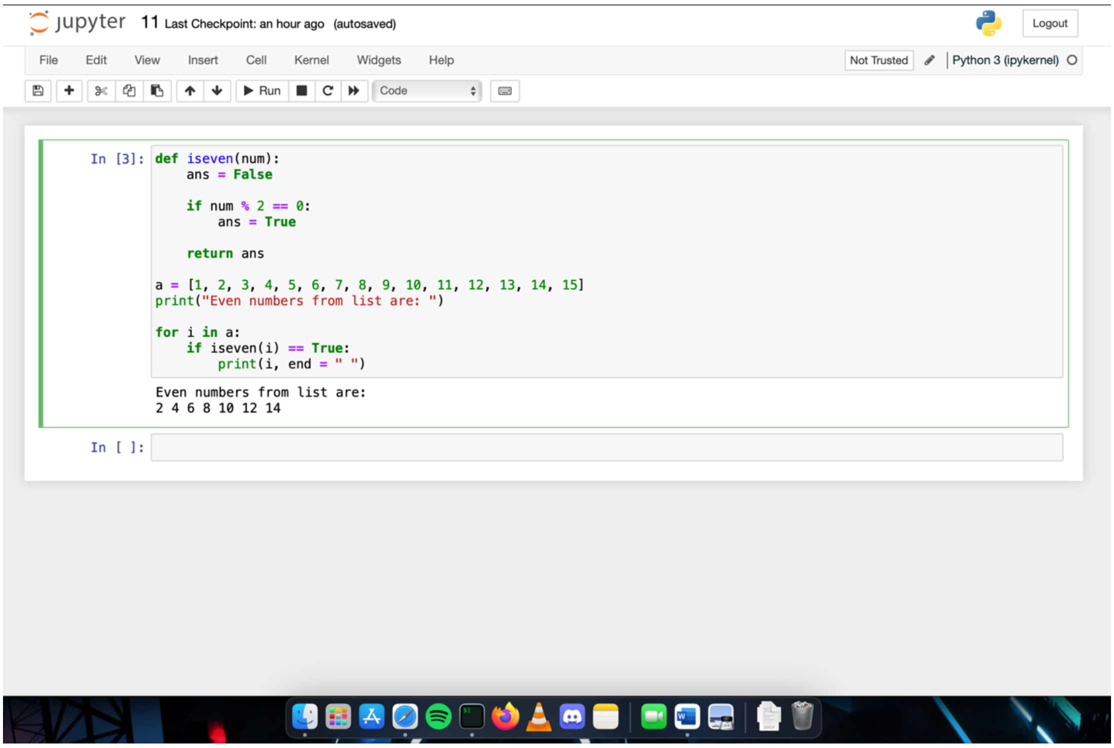
```
def iseven(num):
    ans = False

    if num % 2 == 0:
        ans = True

    return ans

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
print("Even numbers from list are: ")

for i in a:
    if iseven(i) == True:
        print(i, end = " ")
```



jupyter 11 Last Checkpoint: an hour ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [3]:
In [ ]:

```
def iseven(num):
    ans = False

    if num % 2 == 0:
        ans = True

    return ans

a = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
print("Even numbers from list are: ")

for i in a:
    if iseven(i) == True:
        print(i, end = " ")
```

```
Even numbers from list are:
2 4 6 8 10 12 14
```

RESULT: Python program (using function) to print the even numbers from a given list was completed.

12. AIM: Write a Python function (using function) that checks whether a passed string is palindrome or not.
Note : function type - No arguments with return values

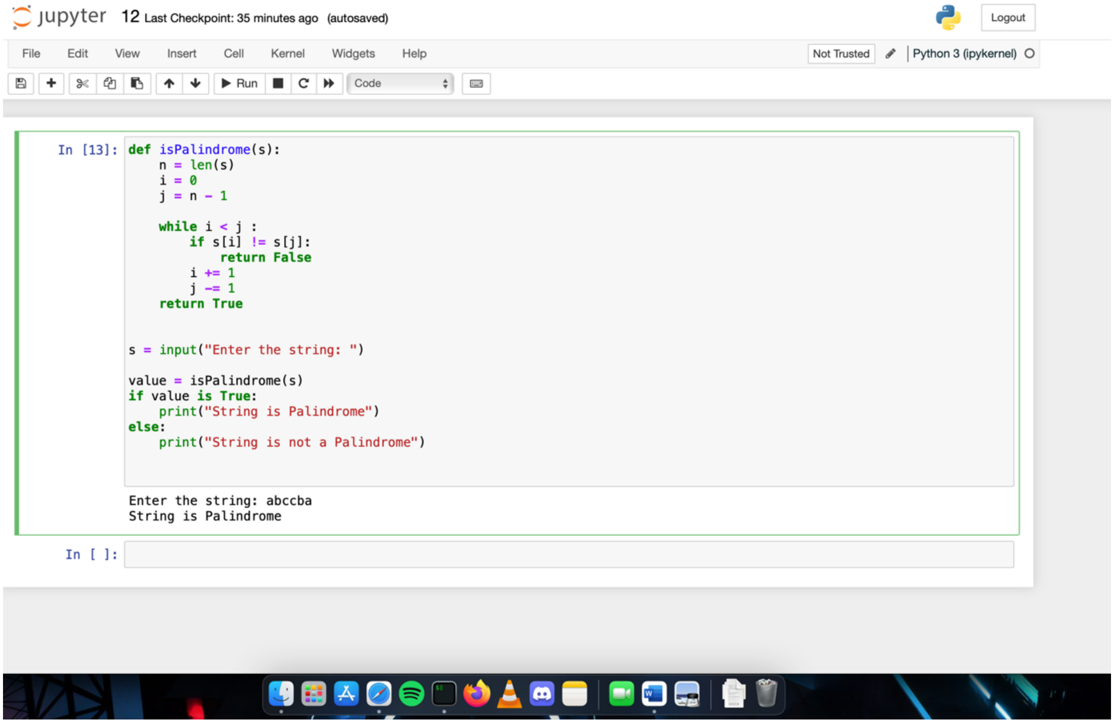
CODE:

```
def isPalindrome(s):
   n = len(s)
   i = 0
   j = n - 1

   while i < j :
      if s[i] != s[j]:
         return False
      i += 1
      j -= 1
   return True


s = input("Enter the string: ")

value = isPalindrome(s)
if value is True:
   print("String is Palindrome")
else:
   print("String is not a Palindrome")
```


jupyter 12 Last Checkpoint: 35 minutes ago (autosaved)
Logout
File Edit View Insert Cell Kernel Widgets Help
Not Trusted
Python 3 (ipykernel)
Run
Code
In [13]:
In [ ]:

```
def isPalindrome(s):
    n = len(s)
    i = 0
    j = n - 1

    while i < j :
        if s[i] != s[j]:
            return False
        i += 1
        j -= 1
    return True


s = input("Enter the string: ")

value = isPalindrome(s)
if value is True:
    print("String is Palindrome")
else:
    print("String is not a Palindrome")
```

```
Enter the string: abccba
String is Palindrome
```

RESULT: Python program to function (using function) that checks whether a passed string is palindrome or not was completed.

13. AIM: Write a Python function (using function) that checks whether a given number is prime or not

Note : function type - with arguments with return values

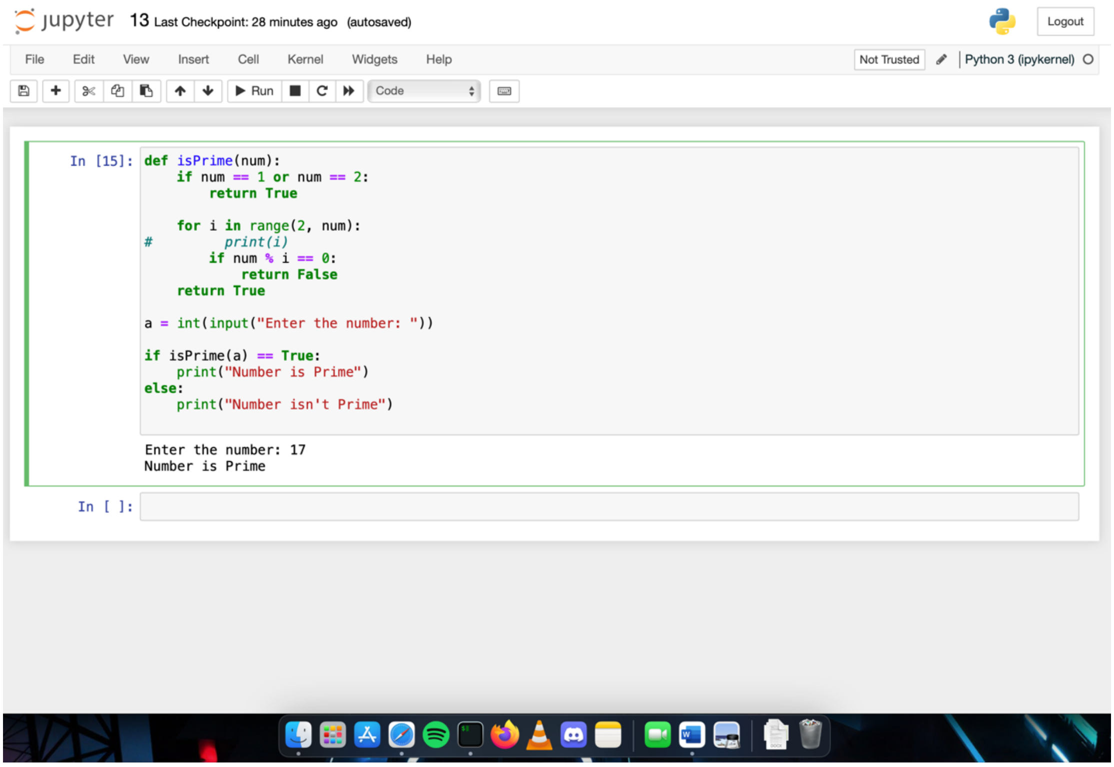
CODE:

```
def isPrime(num):
    if num == 1 or num == 2:
        return True

    for i in range(2, num):
#        print(i)
        if num % i == 0:
            return False
    return True

a = int(input("Enter the number: "))

if isPrime(a) == True:
    print("Number is Prime")
else:
    print("Number isn't Prime")
```



```
In [15]: def isPrime(num):
             if num == 1 or num == 2:
                 return True

             for i in range(2, num):
         #           print(i)
                 if num % i == 0:
                     return False
             return True

         a = int(input("Enter the number: "))

         if isPrime(a) == True:
             print("Number is Prime")
         else:
             print("Number isn't Prime")

         Enter the number: 17
         Number is Prime

In [ ]:
```

RESULT: Python program to function (using function) that checks whether a given number is prime or not was completed.