



**Department of Networking and Communications**

**SRM IST, Kattankulathur – 603 203**

**Course Code: 18CSC207J**

**Course Name: Advanced Programming Practice**

<b>Experiment No</b>	1
<b>Title of Experiment</b>	To complete all the 20 problems in Jupyter environment
<b>Name of the Student</b>	Abhijay Rajvansh (RA2011003010398)
<b>Date of Experiment</b>	18 / 03 / 2022

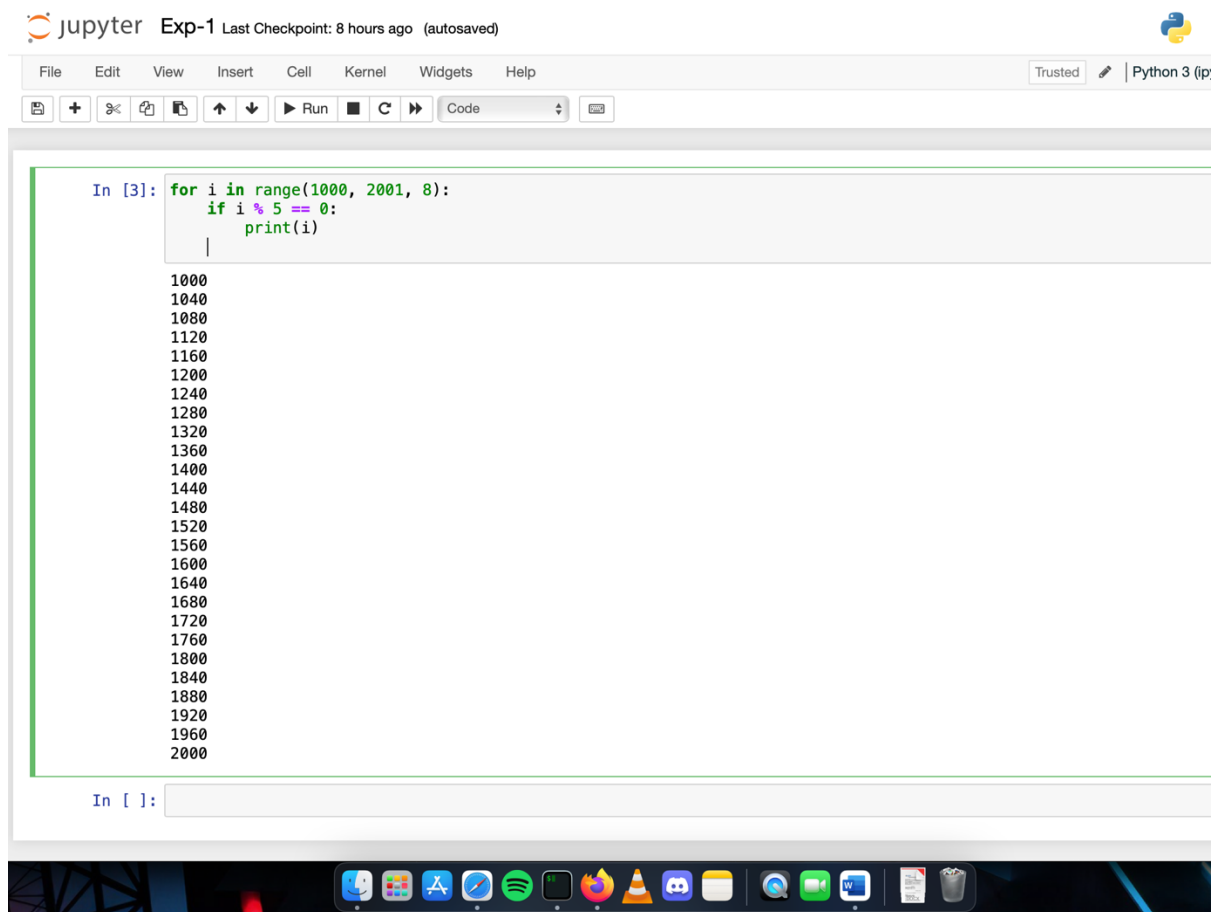
**Staff Signature with date**

## **AIM**

1. Write a Python program to find those numbers which are divisible by 8 and multiple of 5, between 1000 and 2000 (both included)

CODE:

```
for i in range(1000, 2001, 8):  
    if i % 5 == 0:  
        print(i)
```



The image shows a Jupyter Notebook interface. At the top, it says "jupyter Exp-1 Last Checkpoint: 8 hours ago (autosaved)". The interface includes a menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu bar is a toolbar with icons for saving, adding cells, undo, redo, running, and other functions. The main area contains a code cell with the following Python code:

```
In [3]: for i in range(1000, 2001, 8):  
        if i % 5 == 0:  
            print(i)
```

The output of the code is a list of numbers printed vertically:

```
1000  
1040  
1080  
1120  
1160  
1200  
1240  
1280  
1320  
1360  
1400  
1440  
1480  
1520  
1560  
1600  
1640  
1680  
1720  
1760  
1800  
1840  
1880  
1920  
1960  
2000
```

At the bottom of the notebook, there is an input prompt "In [ ]:" followed by a text box. The entire notebook is displayed within a browser window, with a macOS dock visible at the bottom.

## Result

Python program to find those numbers which are divisible

by 8 and multiple of 5, between 1000 and 2000 (both included) was completed.

## **Aim**

2. Write a Python program to guess a number between 1 to 9.

Note: User is prompted to enter a guess. If the user guesses wrong then the prompt appears again until the guess is correct, on successful guess, user will get a “Well guessed!” message, and the program will exit.

## **CODE:**

```
import random

num = random.choice(range(1,10))

# print(num)

# global guess = 0

while(True):

    guess = int(input("Enter your guess: "))

    if guess == num:

        print("Well guessed!")

        break

    if guess > num:

        print("Try Smaller num...")

    if guess < num:

        print("Try Greater num...")
```

```
File Edit View Insert Cell Kernel Widgets Help
```

```
Save + Undo Copy Paste Up Down Run Stop Restart Code
```

```
In [2]: import random

num = random.choice(range(1,10))
# print(num)
# global guess = 0

while(True):
    guess = int(input("Enter your guess: "))
    if guess == num:
        print("Well guessed!")
        break
    if guess > num:
        print("Try Smaller num...")
    if guess < num:
        print("Try Greater num...")|

Enter your guess: 1
Try Greater num...
Enter your guess: 9
Try Smaller num...
Enter your guess: 4
Try Smaller num...
Enter your guess: 2
Well guessed!
```

```
In [ ]:
```

## Result:

Python program to guess the num by the user was completed.

## Aim

3. Write a Python program to construct the following pattern, using a nested for loop.

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

## CODE:

```
n = 5  
  
for i in range(n):  
    for j in range(i+1):  
        print('*', end = " ")  
    print()  
  
for i in range(n, -1, -1):  
    for j in range(i+1):  
        print('*', end = " ")  
    print()
```

```
In [4]: n = 5

for i in range(n):
    for j in range(i+1):
        print('*', end = " ")
    print()

for i in range(n, -1, -1):
    for j in range(i+1):
        print('*', end = " ")
    print()
```

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

In [ ]:

## Result:

Python program to print the given pattern was completed.

**Aim**

4. Write a Python program that accepts a word from the user and reverse it. ( should not use any functions)

**CODE:**

```
word = input("Enter the word: ")
```

```
rev_str = ""
```








```
for i in word:
```





```
    rev_str = i + rev_str
```


```
print("Reversed String: ", rev_str)
```




FileEditViewInsertCellKernelWidgetsHelp



 Run

Code



```
In [1]: word = input("Enter the word: ")
rev_str = ""
for i in word:
    rev_str = i + rev_str
print("Reversed String: ", rev_str)

Enter the word: Abhijay Rajvansh
Reversed String:  hsnavjaR yajihbA

In [ ]:
```

## Result

Python program that accepts a word from the user and reverse it was completed.

**AIM**

5. Write a Python program which takes two digits m (row) and n (column) as input and generates a two-dimensional array. The element value in the i-th row and j-th column of the array should be  $i*j$ . Note :

$i = 0, 1, \dots, m-1$

$j = 0, 1, \dots, n-1$ .

Test Data : Rows = 3, Columns = 4

Expected Result : `[[0, 0, 0, 0], [0, 1, 2, 3], [0, 2, 4, 6]]`

**CODE:**

```
# Matrix
```

```
n = int(input("Enter Rows no: "))
```

```
m = int(input("Enter Column no: "))
```

```
mat = []
```

```
for i in range(n):
```

```
    for j in range(m):
```

```
        print(int(i * j), end = " ")
```

```
    print()
```

```
File Edit View Insert Cell Kernel Widgets Help
[Save] [New] [Cut] [Copy] [Paste] [Up] [Down] [Run] [Stop] [Refresh] [Next] Code [Terminal]

In [1]: # Matrix

n = int(input("Enter Rows no: "))
m = int(input("Enter Column no: "))

mat = []

for i in range(n):
    for j in range(m):
        print(int(i * j), end = " ")
    print()

Enter Rows no: 3
Enter Column no: 4
0 0 0 0
0 1 2 3
0 2 4 6

In [ ]: 
```

**Result:**

Python program which takes two digits m (row) and n (column) as input and generates a two-dimensional array was completed.

**Aim**

6. Write a Python program that accepts a string and calculate the number of digits and letters.

Sample Data : SRMIST 2022

Expected Output :

Letters 6

Digits 4

**CODE:**

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

```
num = 0
```

```
alpha = 0
```

```
for i in s:
```

```
    if i.isalpha():
```

```
        alpha += 1
```

```
    elif i.isnumeric():
```

```
        num += 1
```

```
print("Letters: ",(alpha))
```

```
print("Digits: " ,(num))
```

```
File Edit View Insert Cell Kernel Widgets Help
[Save] [New] [Cut] [Copy] [Paste] [Up] [Down] [Run] [Stop] [Refresh] [Next] [Code] [Terminal]

In [1]: s = input("Enter the string: ")
        num = 0
        alpha = 0

        for i in s:
            if i.isalpha():
                alpha += 1
            elif i.isnumeric():
                num += 1

        print("Letters: ",(alpha))
        print("Digits: " ,(num))

Enter the string: SRMIST 2022
Letters:  6
Digits:  4

In [ ]:
```

### Result:

Python program that accepts a string and calculate the number of digits and letters was completed.

## AIM

7. Write a Python program to check the validity of password input by users.

- At least 1 letter between [a-z] and 1 letter between [A-Z].
- At least 1 number between [0-9].
- At least 1 character from [\$#@].
- Minimum length 6 characters.
- Maximum length 16 characters.

CODE:

```
password = input("Enter Your Password: ")

password_length = len(password)

num = False

cap = False

small = False

special_char = False

for i in password:

    if i.isupper():

        cap = True

    if i.islower():

        small = True

    if i.isnumeric():

        num = True

    if i == '$' or i == '#' or i == '@':

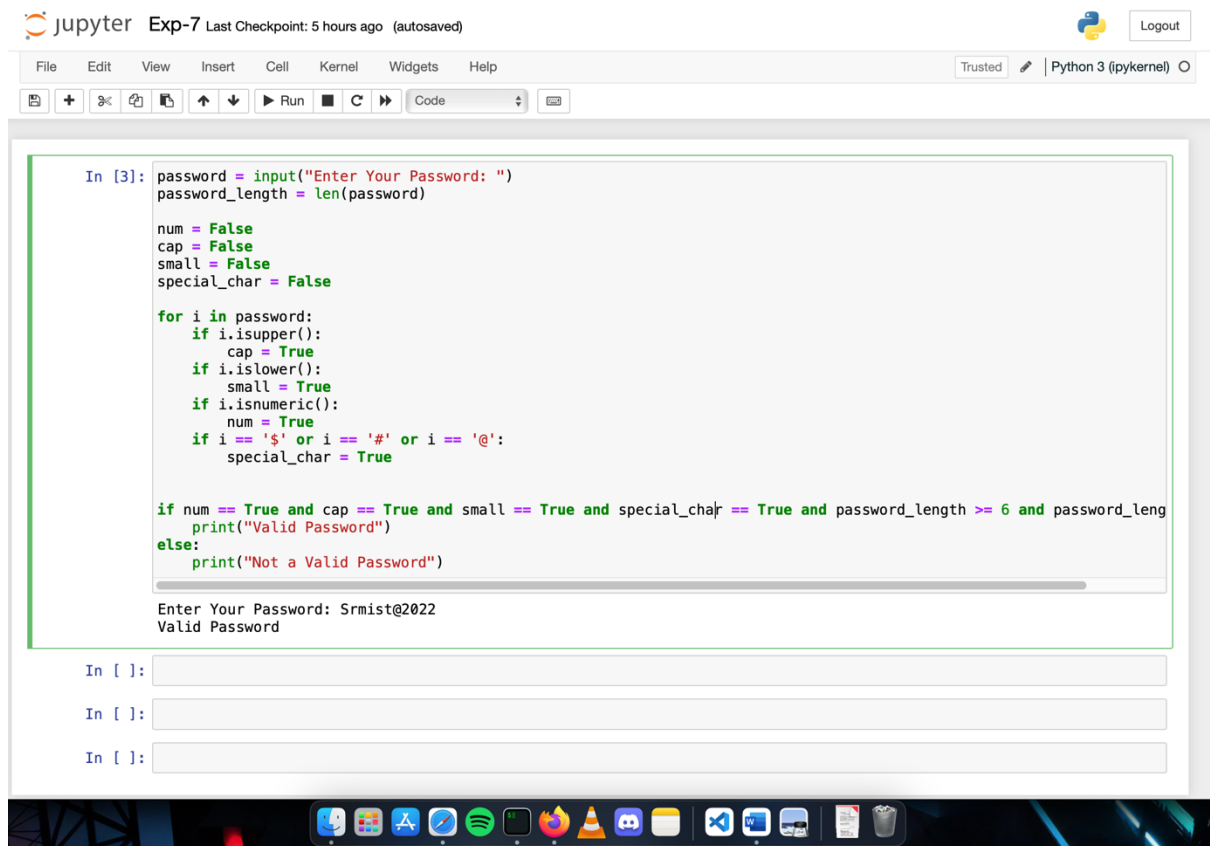
        special_char = True

if num == True and cap == True and small == True and special_char == True and password_length >= 6 and password_length <= 16:

    print("Valid Password")

else:

    print("Not a Valid Password")
```



## Result

Python program to check the validity of password input by users was completed.

**AIM**

8. Write a Python program to find numbers between 100 and 400 (both included) where each digit of a number is an even number. The numbers obtained should be printed in a comma-separated sequence.

**Code:**

```
def checkEvenDigitsOnly(i):
```

```
    value = str(i)
```

```
    ans = True
```

```
    for num in value:
```

```
        num = int(num)
```

```
        if num % 2 != 0:
```

```
            ans = False
```

```
        return ans
```

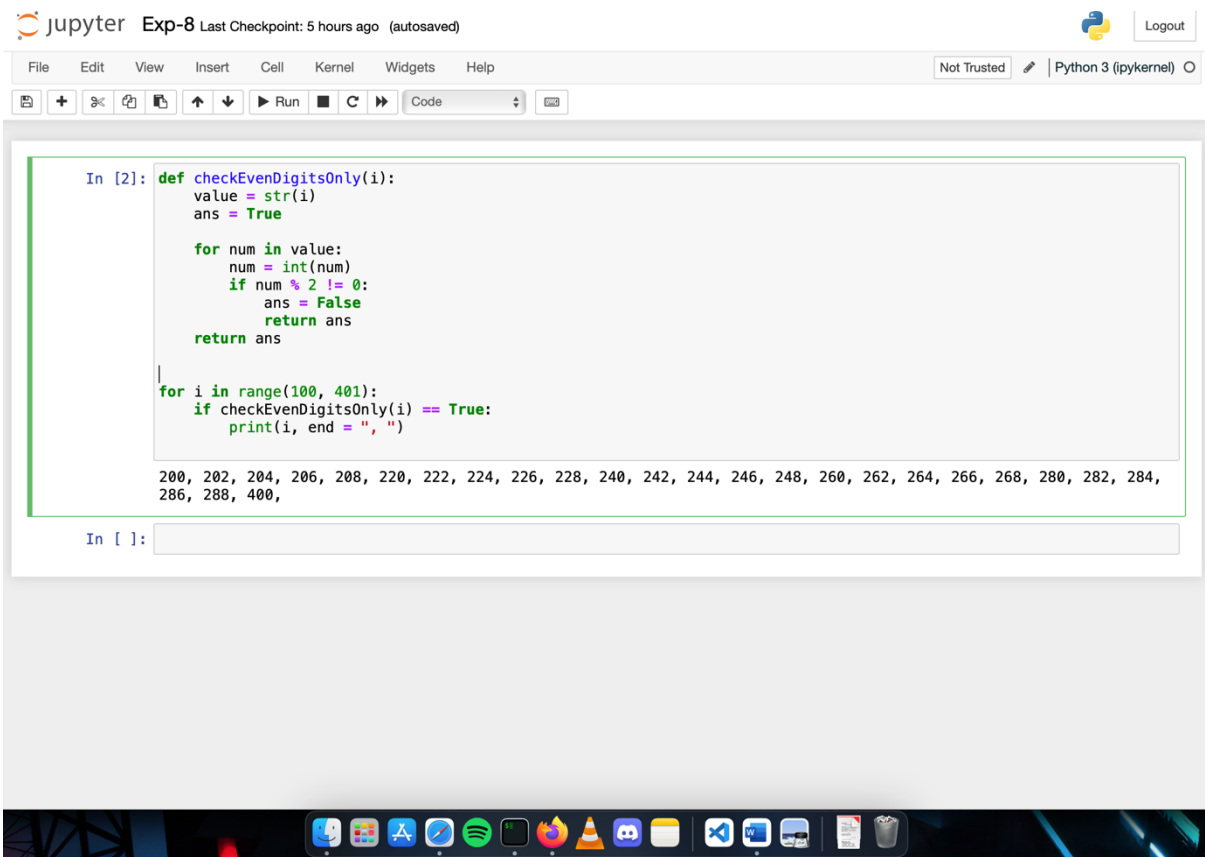
```
    return ans
```

```
for i in range(100, 401):
```

```
    if checkEvenDigitsOnly(i) == True:
```

```
        print(i, end = ", ")
```





## Result

Python program to find numbers between 100 and 400 (both included) where each digit of a number is an even number was completed.

**AIM**

9. Write a Python program to convert month name to a number of days.

**CODE:**

```
month = input("Enter Month Name: ")

l1 = ("january", "march", "may", "july", "august", "october", "december")
l2 = ("april", "june", "september", "november")

if month == "february":
    print("28 or 29")
elif month in l1:
    print(31)
elif month in l2:
    print(30)
else:
    print("Enter a valid month name!")
```

```
File Edit View Insert Cell Kernel Widgets Help
+ < > Run [Code]
In [12]: month = input("Enter Month Name: ")
        l1 = ("january", "march", "may", "july", "august", "october", "december")
        l2 = ("april", "june", "september", "november")
        if month == "february":
            print("28 or 29")
        elif month in l1:
            print(31)
        elif month in l2:
            print(30)
        else:
            print("Enter a valid month name!")
        Enter Month Name: april
        30
In [ ]:
```

## Result

Python program to convert month name to a number of days was completed.

**AIM**

10. Write a Python program to sum of two given integers. However, if the sum is between 105 to 200 it will return 200.

**CODE:**

```
a = int(input("Enter 1st num: "))
```

```
b = int(input("Enter 2nd num: "))
```

```
compute = a+b
```

```
if compute >= 105 and compute <= 200:
```

```
    print(200)
```

```
else:
```

```
    print(compute)
```

```
In [4]: a = int(input("Enter 1st num: "))
        b = int(input("Enter 2nd num: "))

        compute = a+b

        if compute >= 105 and compute <= 200:
            print(200)
        else:
            print(compute)

        Enter 1st num: 1
        Enter 2nd num: 1
        2
```

In [ ]:

In [ ]:

In [ ]:

## Result

Python program to sum of two given integers. However, if the sum is between 105 to 200 it will return 200 was completed.

**AIM**

11. Write a Python program to construct the following pattern, using a nested loop number.

Expected Output:

999999999

88888888

7777777

666666

55555

4444

333

22

1

**CODE:**

```
n = 9
```

```
k = n
```

```
for i in range(n, 0, -1):
```













```
    for j in range(i):
```

```
        print(k, end = "")
```

```
    k -= 1
```

```
    print()
```

File Edit View Insert Cell Kernel Widgets Help

        Run    Code 

---

In [16]:

```
n = 9
k = n
for i in range(n, 0, -1):
    for j in range(i):
        print(k, end = " ")
    k -= 1
    print()
```

999999999
888888888
7777777
666666
55555
4444
333
22
1

In [ ]:

In [ ]:

## Result

Python program to construct the following pattern, using a nested loop number was completed.

## AIM

12. Write a Python program to create a histogram from a given list of integers.

## CODE:

```
from matplotlib import pyplot as plt  
  
import numpy as np  
  
#sample list of integers  
a = np.array([87, 5, 43, 56, 73, 55, 54, 11, 20, 51, 5, 79, 31,27])  
  
fig, ax = plt.subplots(figsize =(6, 4))  
  
ax.hist(a, bins = [0, 25, 50, 75, 100])  
  
plt.show()
```

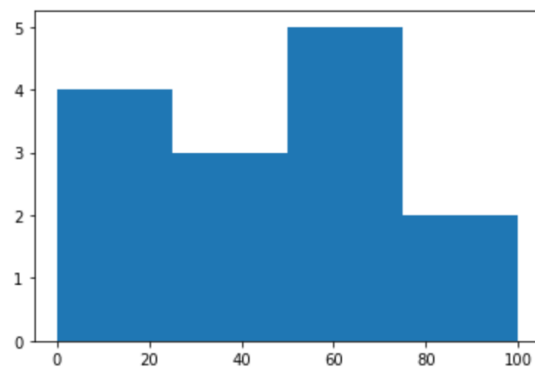


```
In [7]: from matplotlib import pyplot as plt
import numpy as np

#sample list of integers
a = np.array([87, 5, 43, 56, 73, 55, 54, 11, 20, 51, 5, 79, 31,27])

fig, ax = plt.subplots(figsize =(6, 4))
ax.hist(a, bins = [0, 25, 50, 75, 100])

plt.show()
```



In [ ]:

## Result

Python program to create a histogram from a given list of integers was completed.

**AIM**

13. Write a Python program that will return true if the two given integer values are equal or their sum or difference is 5.

**CODE:**

```
a = int(input("Enter 1st num: "))
```

```
b = int(input("Enter 2nd num: "))
```

```
sum = a + b
```

```
diff = abs(a - b)
```

```
if a == b or sum == 5 or diff == 5:
```

```
    print(True)
```

```
else:
```

```
    print(False)
```

jupyter Exp-13 Last Checkpoint: 2 hours ago (autosaved)

## Result

Python program that will return true if the two given integer values are equal or their sum or difference is 5 was completed.

**AIM**

14. Write a Python program to compute the distance between the points (x1, y1) and (x2, y2).

**CODE:**

```
import math
```

```
x1 = int(input("Enter x1: "))
```

```
x2 = int(input("Enter x2: "))
```

```
y1 = int(input("Enter y1: "))
```

```
y2 = int(input("Enter y2: "))
```

```
ans = math.sqrt(math.pow(x2 - x1, 2) + math.pow(y2 - y1, 2) * 1.0);
```

```
print("Distance between two coordinates: ", round(ans, 2))
```

File
Edit
View
Insert
Cell
Kernel
Widgets
Help

Run



Code

```

In [1]: import math

x1 = int(input("Enter x1: "))
x2 = int(input("Enter x2: "))
y1 = int(input("Enter y1: "))
y2 = int(input("Enter y2: "))

ans = math.sqrt(math.pow(x2 - x1, 2) + math.pow(y2 - y1, 2) * 1.0);

print("Distance between two coordinates: ", round(ans, 2))

Enter x1: 3
Enter x2: 5
Enter y1: 7
Enter y2: 9
Distance between two coordinates:  2.83

```

In [ ]:

## Result

Python program to compute the distance between the points (x1, y1) and (x2, y2) was completed

**AIM**

15. Function that takes a sequence of numbers and determines whether all are different from each other.

**CODE:**

```
def checkDistinct(arr):
```

```
    n = len(arr)
```

```
    ans = True
```

```
    for i in range(n):
```

```
        curr = arr[i]
```

```
        for j in range(i + 1, n):
```

```
            if arr[j] == curr:
```

```
                ans = False
```

```
                return ans
```

```
    return ans
```

```
arr = []
```

```
n = int(input("Enter the number of elements: "))
```

```
for i in range(n):
```

```
    ele = int(input())
```

```
    arr.append(ele)
```

```
if checkDistinct(arr) == True:
```

```
    print("All elements are different from each other")
```

```
else:
```

```
    print("Repetition found")
```

The screenshot shows a Jupyter Notebook window titled "jupyter Exp-15 Last Checkpoint: 2 hours ago (unsaved changes)". The interface includes a top menu bar with "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". Below the menu is a toolbar with icons for saving, running, and other actions. The main area displays a code cell with the following Python code:

```
In [1]: def checkDistinct(arr):
        n = len(arr)
        ans = True

        for i in range(n):
            curr = arr[i]
            for j in range(i + 1, n):
                if arr[j] == curr:
                    ans = False
                    return ans
            return ans

arr = []

n = int(input("Ente the number of elements: "))

for i in range(n):
    ele = int(input())
    arr.append(ele)

if checkDistinct(arr) == True:
    print("All elements are different from each other")
else:
    print("Repetition found")

Ente the number of elements: 5
1
2
3
3
5
Repetition found
```

The output of the code is visible below the code cell, showing the input sequence and the final result "Repetition found". The bottom of the image shows a macOS dock with various application icons.

## Result

Python program to create a function that takes a sequence of numbers and determines whether all are different from each other was completed.

## AIM

16. Write a Python program to count the number of each character of a given text :-

CODE:

```
import collections
```

```
def frequency(s):
```

```
    return collections.Counter(s)
```

```
if __name__ == "__main__":
```

```
    s = "ABHIJAY"
```

```
    freq = frequency(s)
```

```
    for (key, value) in freq.items():
```

```
        print (key, "-> ", value)
```



```
In [2]: import collections

def frequency(s):
    return collections.Counter(s)

if __name__ == "__main__":
    s = "ABHIJAY"
    freq = frequency(s)

    for (key, value) in freq.items():
        print (key, " -> ", value)
```

```
A -> 2
B -> 1
H -> 1
I -> 1
J -> 1
Y -> 1
```

In [ ]:

## Result

Python program to count the number of each character of a given text was completed.

**AIM**

17. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive.

**CODE:**

```
num = int(input("Enter a positive num: "))
```

```
def sumofdigit(num):
```

```
    sum = 0
```

```
    while num != 0:
```

```
        ld = num % 10
```

```
        sum += ld;
```

```
        num //= 10;
```

```
    return sum
```

```
while num > 0:
```

```
    num -= sumofdigit(num)
```

```
print(num)
```

```
File Edit View Insert Cell Kernel Widgets Help
[Icons] Run [Code]

In [20]: num = int(input("Enter a positive num: "))

def sumofdigit(num):
    sum = 0
    while num != 0:
        ld = num % 10
        sum += ld
        num //= 10
    return sum

while num > 0:
    num -= sumofdigit(num)
    print(num)

Enter a positive num: 123
117
108
99
81
72
63
54
45
36
27
18
9
0

In [ ]:
In [ ]:
```

## Result

Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive was completed.

**AIM**

18. Write a Python program to find the digits which are absent in a given mobile number.

**CODE:**

```
def absent_digits(n):  
    all_nums = set([0,1,2,3,4,5,6,7,8,9])  
    n = set([int(i) for i in n])  
    n = n.symmetric_difference(all_nums)  
    n = sorted(n)  
  
    return n  
  
num = int(input("Enter number: "))  
res = [int(x) for x in str(num)]  
  
print(absent_digits(res))
```

File
Edit
View
Insert
Cell
Kernel
Widgets
Help

Code

```

In [8]: def absent_digits(n):
        all_nums = set([0,1,2,3,4,5,6,7,8,9])
        n = set([int(i) for i in n])
        n = n.symmetric_difference(all_nums)
        n = sorted(n)

        return n

        num = int(input("Enter number: "))
        res = [int(x) for x in str(num)]

        print(absent_digits(res))

Enter number: 8709247743
[1, 5, 6]

```

In [ ]:

## Result

Python program to find the digits which are absent in a given mobile number was completed.

## AIM

19. Write a Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure

CODE:

```
def rev_number(n):
```

```
    s = 0
```

```
    while True:
```

```
        k = str(n)
```

```
        if k == k[::-1]:
```

```
            break
```

```
        else:
```

```
            m = int(k[::-1])
```

```
            n += m
```

```
            s += 1
```

```
    return n
```

```
num = int(input("Enter number: "))
```

```
print(rev_number(num))
```

```
File Edit View Insert Cell Kernel Widgets Help
```

```

In [4]: def rev_number(n):
        s = 0
        while True:
            k = str(n)
            if k == k[::-1]:
                break
            else:
                m = int(k[::-1])
                n += m
                s += 1
        return n

num = int(input("Enter number: "))
print(rev_number(num))

Enter number: 1212121212
3333333333

In [ ]:

In [ ]:

```

## Result

Python program to reverse the digits of a given number and add it to the original, If the sum is not a palindrome repeat this procedure was completed.

## Aim

20. Write a Python program to print the length of the series and the series from the given 3rd term, 3rd last term and the sum of a series.

### CODE:

```
tn = int(input("Input third term of the series:"))
```

```
tltn = int(input("Input third last term:"))
```

```
s_sum = int(input("Sum of the series:"))
```

```
n = int(2*s_sum/(tn+tltn))
```

```
print("Length of the series: ",n)
```

```
if n-5==0:
```

```
    d = (s_sum-3*tn)//6
```

```
else:
```

```
    d = (tltn-tn)/(n-5)
```

```
a = tn-2*d
```

```
j = 0
```

```
print("\nSeries:")
```

```
for j in range(n-1):
```

```
    print(int(a),end=" ")
```

```
    a+=d
```

```
print(int(a),end=" ")
```



```
In [3]: tn = int(input("Input third term of the series:"))
        tln = int(input("Input third last term:"))
        s_sum = int(input("Sum of the series:"))

        n = int(2*s_sum/(tn+tln))

        print("Length of the series: ",n)

        if n-5==0:
            d = (s_sum-3*tn)//6
        else:
            d = (tln-tn)/(n-5)

        a = tn-2*d
        j = 0

        print("\nSeries:")

        for j in range(n-1):
            print(int(a),end=" ")
            a+=d

        print(int(a),end=" ")

        Input third term of the series:3
        Input third last term:8
        Sum of the series:55
        Length of the series:  10

        Series:
        1 2 3 4 5 6 7 8 9 10
```

## Result:

Python program to print the length of the series and the series from the given 3rd term, 3rd last term and the sum of a series was completed.