

**Name - Vipul S. Tapare**

**Reg . No. – 22MCA1005**

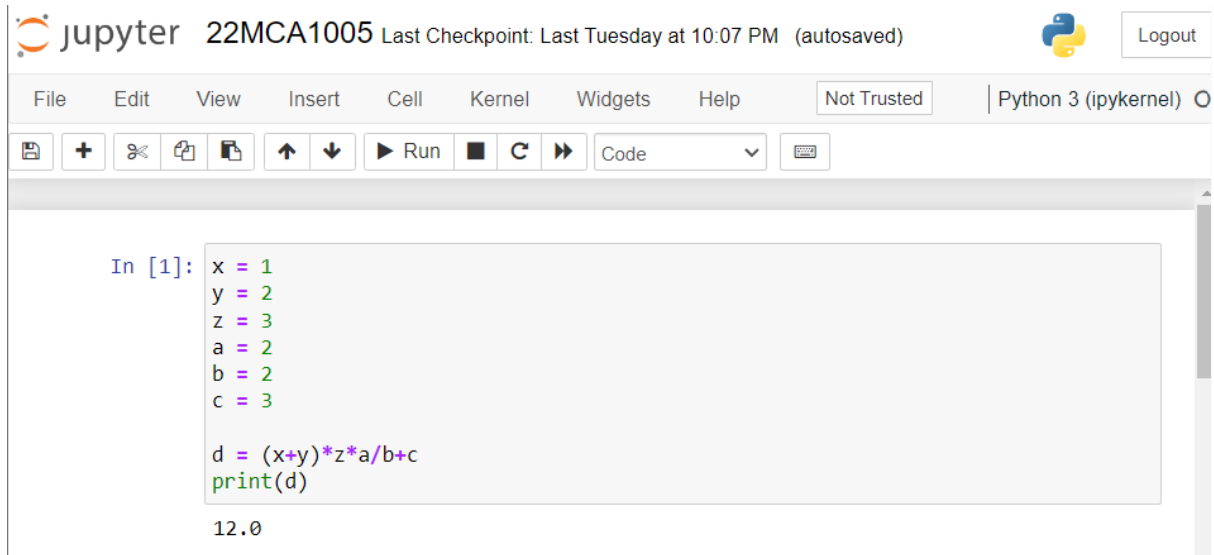
**Faculty Name – Dr. Shahjahan Sofia Nishath**

**Python\_LAB1**

1) Write a program to evaluate the expression:

$$D = (x+y)*z**a//b+c$$

$$D = (1+2)*3**2//2+3$$



```
jupyter 22MCA1005 Last Checkpoint: Last Tuesday at 10:07 PM (autosaved) Logout
```

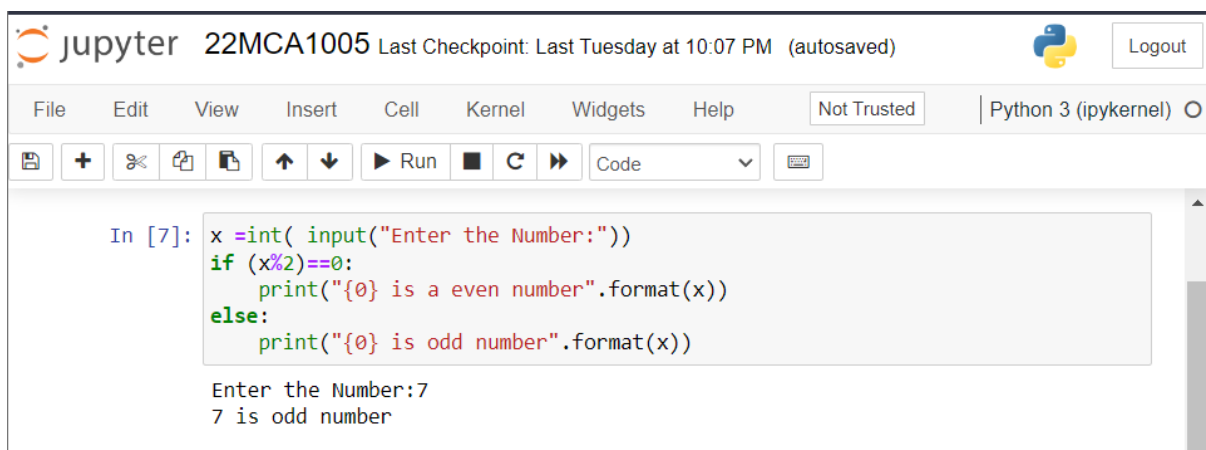
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [1]: x = 1
        y = 2
        z = 3
        a = 2
        b = 2
        c = 3

        d = (x+y)*z**a//b+c
        print(d)

12.0
```

2) Write a program to check whether number is even or not.



```
jupyter 22MCA1005 Last Checkpoint: Last Tuesday at 10:07 PM (autosaved) Logout
```

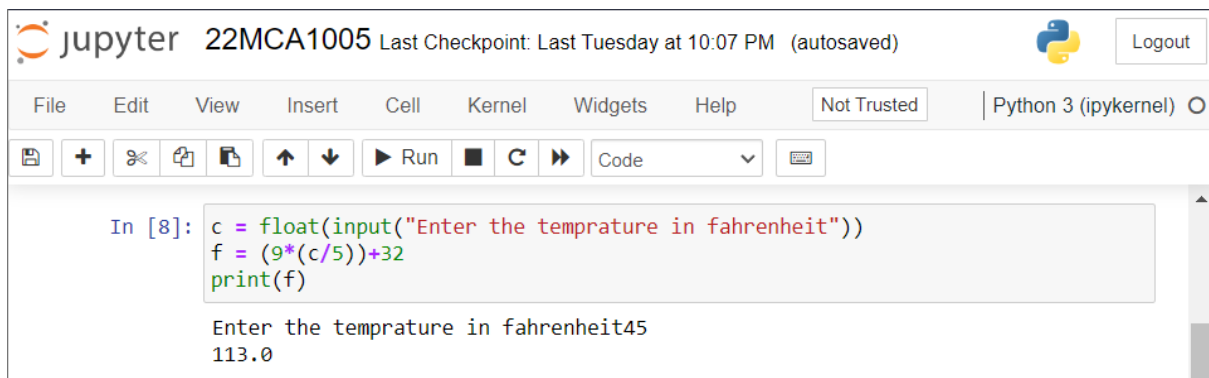
File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

```
In [7]: x =int( input("Enter the Number:"))
        if (x%2)==0:
            print("{0} is a even number".format(x))
        else:
            print("{0} is odd number".format(x))

Enter the Number:7
7 is odd number
```

3) Write a program to convert the temperature from degree centigrade to Fahrenheit.

$$F=(9*(int(c))/5) +32$$



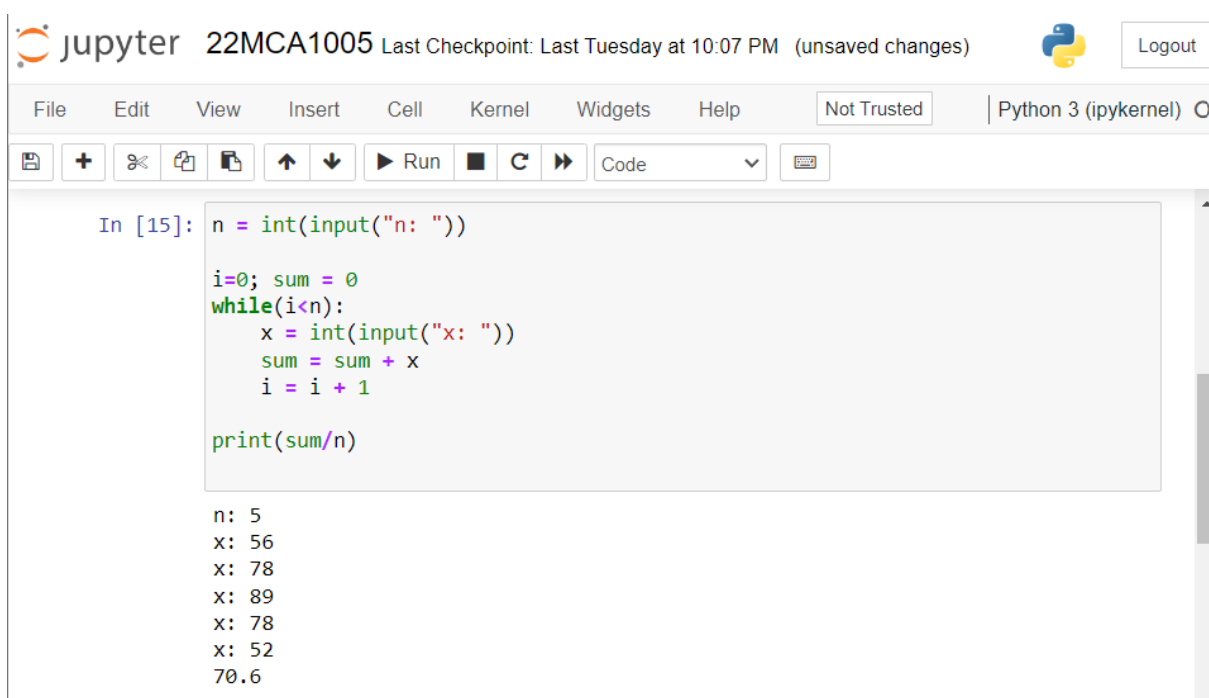
The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar indicates 'Last Checkpoint: Last Tuesday at 10:07 PM (autosaved)' and a 'Logout' button. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The status bar shows 'Not Trusted' and 'Python 3 (ipykernel)'. The code cell contains the following Python code:

```
In [8]: c = float(input("Enter the temprature in fahrenheit"))
        f = (9*(c/5))+32
        print(f)
```

The output of the code is:

```
Enter the temprature in fahrenheit45
113.0
```

**4) Write a program to find the average of a set of n numbers.**



The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar indicates 'Last Checkpoint: Last Tuesday at 10:07 PM (unsaved changes)' and a 'Logout' button. The menu bar includes 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The status bar shows 'Not Trusted' and 'Python 3 (ipykernel)'. The code cell contains the following Python code:

```
In [15]: n = int(input("n: "))

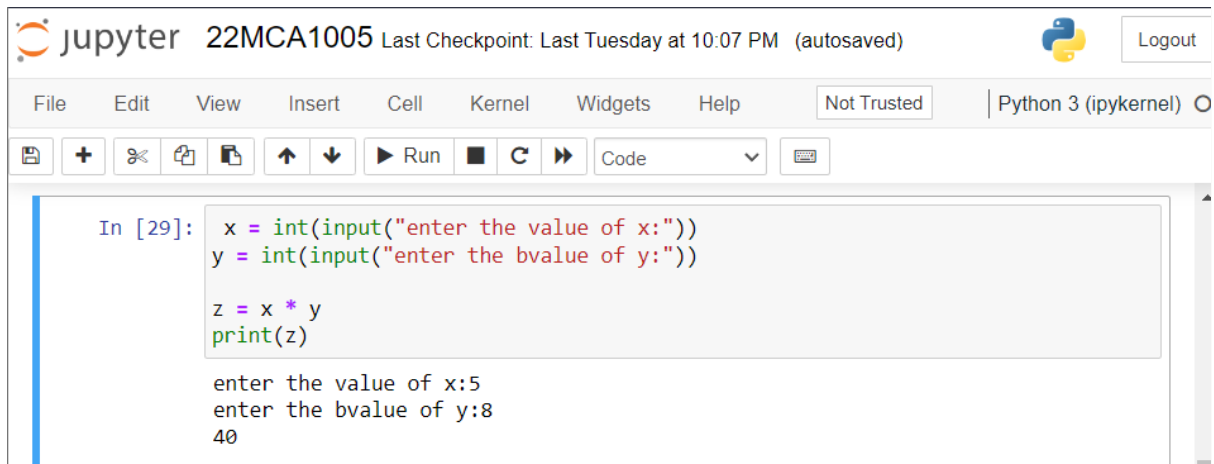
        i=0; sum = 0
        while(i<n):
            x = int(input("x: "))
            sum = sum + x
            i = i + 1

        print(sum/n)
```

The output of the code is:

```
n: 5
x: 56
x: 78
x: 89
x: 78
x: 52
70.6
```

**5) Write a program to find the product of a set of real numbers.**



The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar includes the Jupyter logo, the title, and a 'Last Checkpoint' timestamp. The menu bar contains 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar has icons for saving, adding cells, undo, redo, and running code. The code cell contains the following Python code:

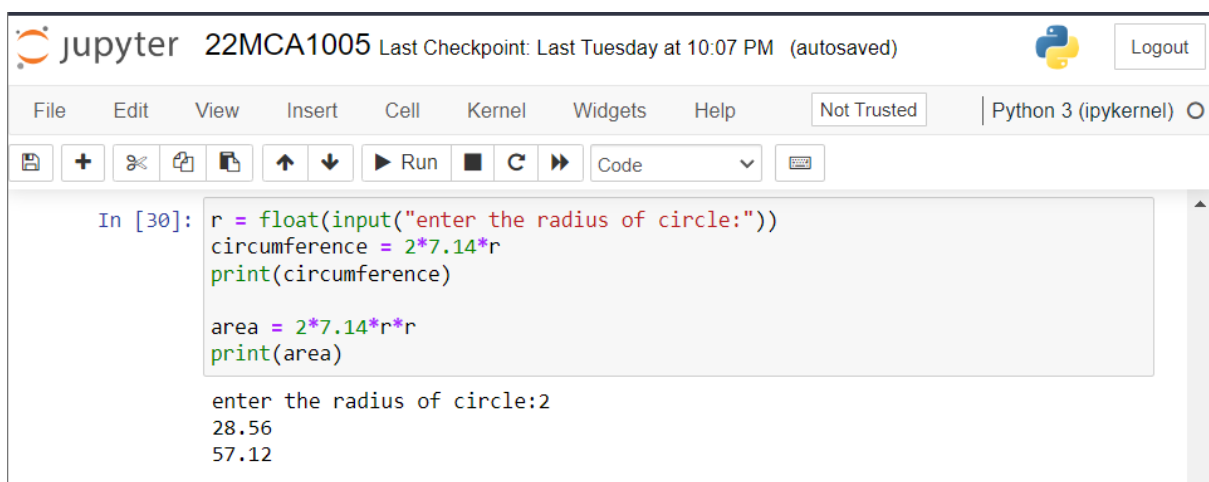
```
In [29]: x = int(input("enter the value of x:"))
y = int(input("enter the bvalue of y:"))

z = x * y
print(z)
```

The output of the code is:

```
enter the value of x:5
enter the bvalue of y:8
40
```

6) Write a program to find the circumference and area of a circle given the radius.



The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar includes the Jupyter logo, the title, and a 'Last Checkpoint' timestamp. The menu bar contains 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The toolbar has icons for saving, adding cells, undo, redo, and running code. The code cell contains the following Python code:

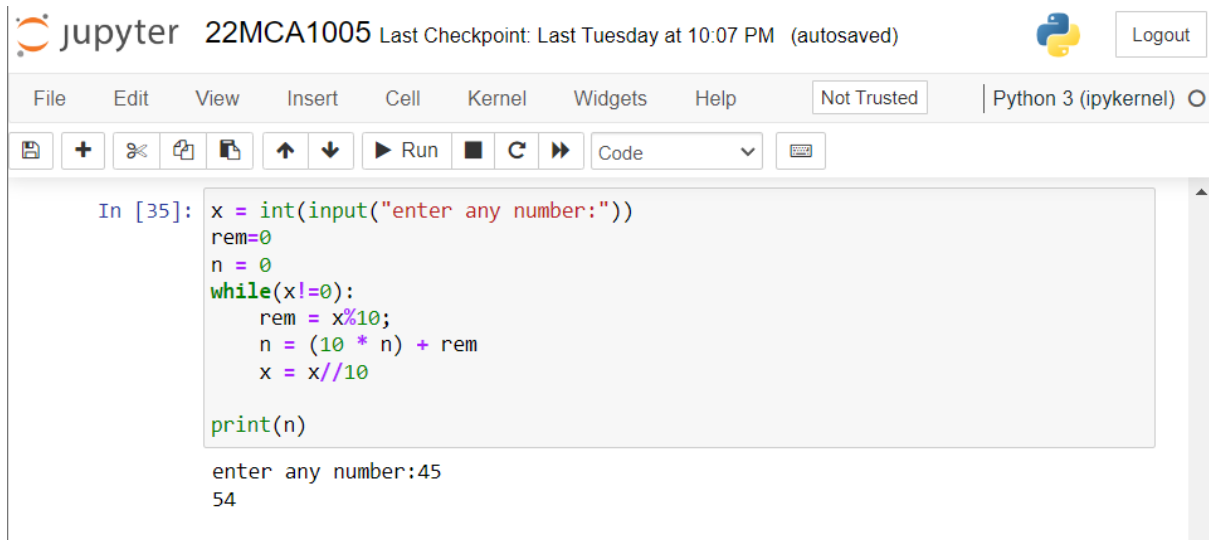
```
In [30]: r = float(input("enter the radius of circle:"))
circumference = 2*7.14*r
print(circumference)

area = 2*7.14*r*r
print(area)
```

The output of the code is:

```
enter the radius of circle:2
28.56
57.12
```

7) Write a program to display the given integer in a reverse manner.



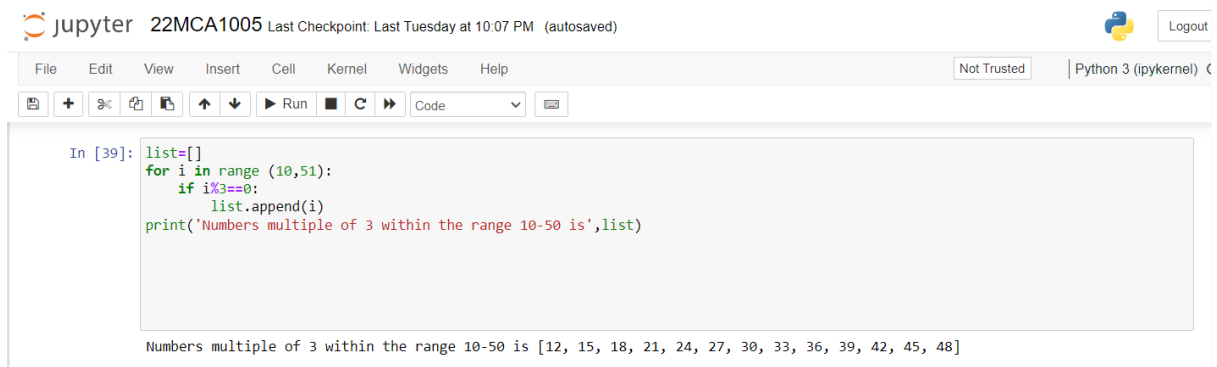
Jupyter 22MCA1005 Last Checkpoint: Last Tuesday at 10:07 PM (autosaved) Python 3 (ipykernel)

```
In [35]: x = int(input("enter any number:"))
rem=0
n = 0
while(x!=0):
    rem = x%10;
    n = (10 * n) + rem
    x = x//10

print(n)

enter any number:45
54
```

**8) Write a program to display all numbers of multiples of 3 within the range 10 to 50.**

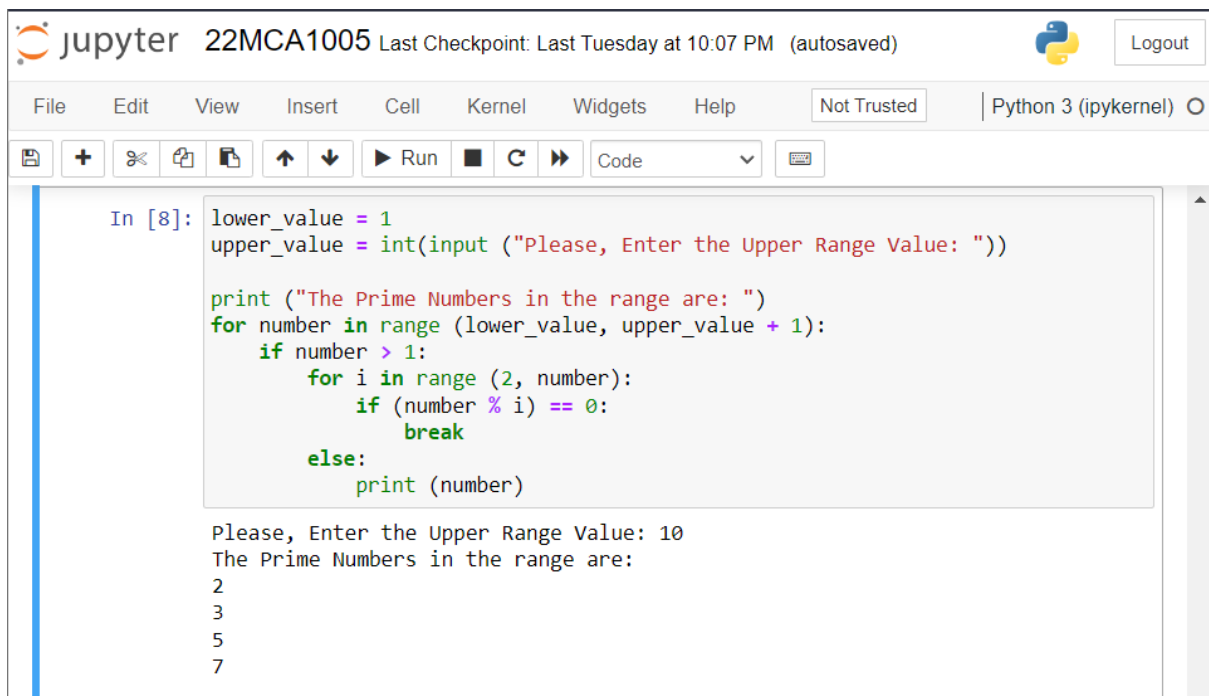


Jupyter 22MCA1005 Last Checkpoint: Last Tuesday at 10:07 PM (autosaved) Python 3 (ipykernel)

```
In [39]: list=[]
for i in range (10,51):
    if i%3==0:
        list.append(i)
print('Numbers multiple of 3 within the range 10-50 is',list)

Numbers multiple of 3 within the range 10-50 is [12, 15, 18, 21, 24, 27, 30, 33, 36, 39, 42, 45, 48]
```

**9) Write a program to generate the prime numbers from 1 to N**



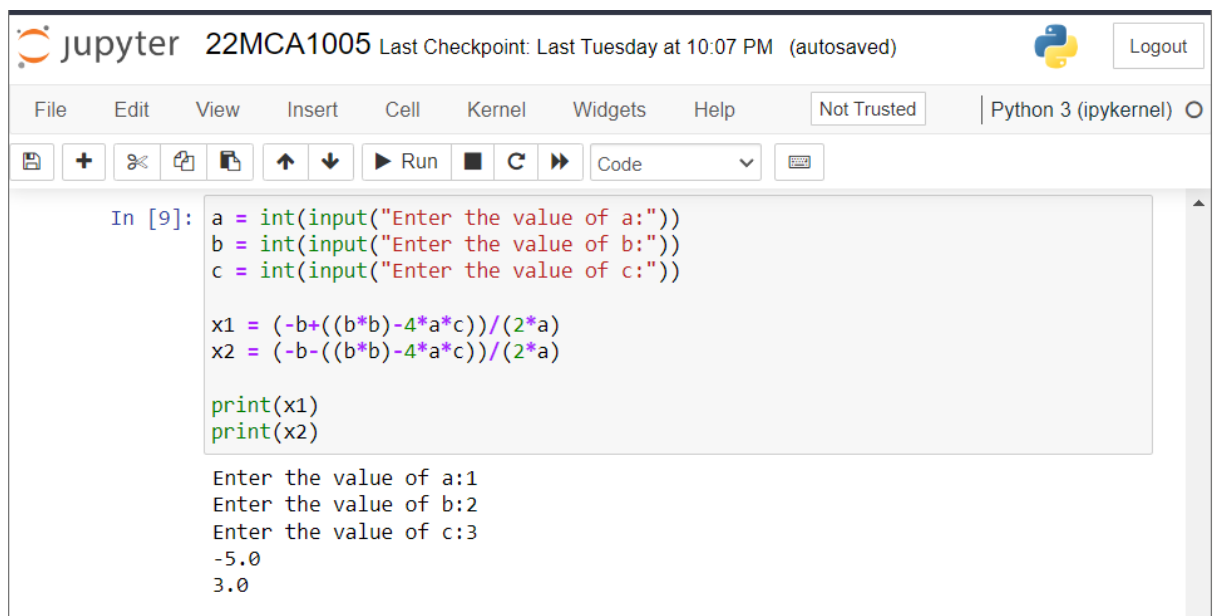
The image shows a Jupyter Notebook interface with the title "22MCA1005". The top bar includes a "Logout" button and a status "Not Trusted". The menu bar contains "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The toolbar has icons for saving, adding cells, undo, redo, and running code. The code cell, labeled "In [8]:", contains a Python program that prompts the user for an upper range value and prints prime numbers in that range. The output shows the user entered 10, and the program printed the prime numbers 2, 3, 5, and 7.

```
In [8]: lower_value = 1
upper_value = int(input("Please, Enter the Upper Range Value: "))

print("The Prime Numbers in the range are: ")
for number in range(lower_value, upper_value + 1):
    if number > 1:
        for i in range(2, number):
            if (number % i) == 0:
                break
        else:
            print(number)

Please, Enter the Upper Range Value: 10
The Prime Numbers in the range are:
2
3
5
7
```

10) Write a program to find the roots of a quadratic equation.



The image shows a Jupyter Notebook interface with the title "22MCA1005". The top bar includes a "Logout" button and a status "Not Trusted". The menu bar contains "File", "Edit", "View", "Insert", "Cell", "Kernel", "Widgets", and "Help". The toolbar has icons for saving, adding cells, undo, redo, and running code. The code cell, labeled "In [9]:", contains a Python program that prompts the user for the coefficients a, b, and c of a quadratic equation, calculates the roots using the quadratic formula, and prints the results. The output shows the user entered a=1, b=2, and c=3, and the program printed the roots -5.0 and 3.0.

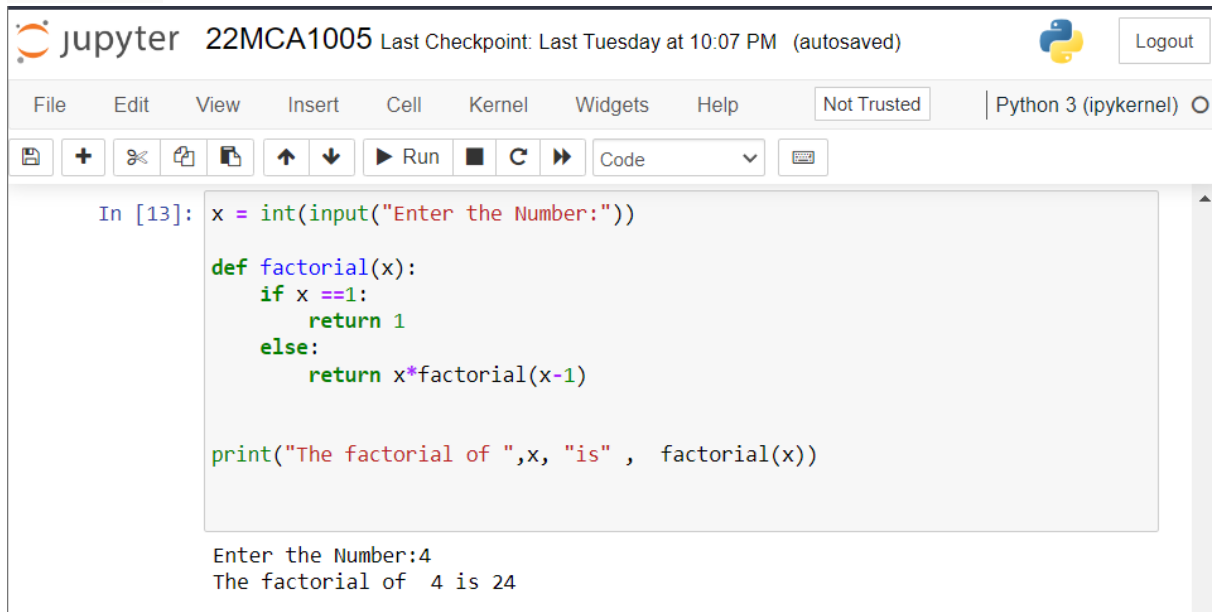
```
In [9]: a = int(input("Enter the value of a:"))
b = int(input("Enter the value of b:"))
c = int(input("Enter the value of c:"))

x1 = (-b+((b*b)-4*a*c))/(2*a)
x2 = (-b-((b*b)-4*a*c))/(2*a)

print(x1)
print(x2)

Enter the value of a:1
Enter the value of b:2
Enter the value of c:3
-5.0
3.0
```

**11) Write a program to find the factorial of a given number using recursion.**



The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar includes a 'Logout' button and the text 'Last Checkpoint: Last Tuesday at 10:07 PM (autosaved)'. The menu bar contains 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The status bar shows 'Not Trusted' and 'Python 3 (ipykernel)'. The toolbar includes icons for saving, adding, undo, redo, and running code. The code cell contains the following Python code:

```
In [13]: x = int(input("Enter the Number:"))

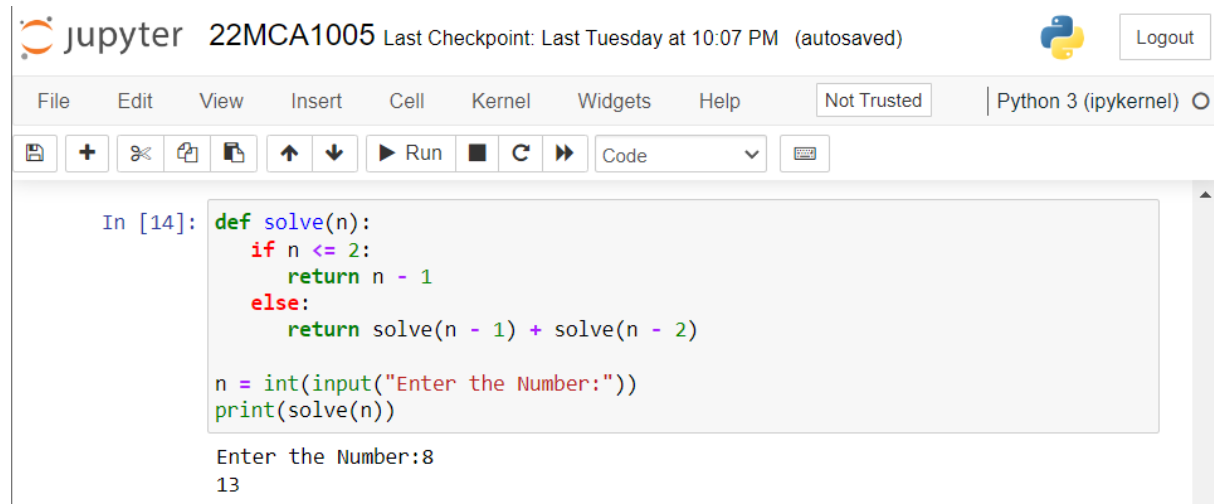
def factorial(x):
    if x ==1:
        return 1
    else:
        return x*factorial(x-1)

print("The factorial of ",x, "is" , factorial(x))
```

The output of the code is:

```
Enter the Number:4
The factorial of  4 is 24
```

**12) Write a program to find the Nth term in a Fibonacci series using recursion.**



The image shows a Jupyter Notebook interface with the title '22MCA1005'. The top bar includes a 'Logout' button and the text 'Last Checkpoint: Last Tuesday at 10:07 PM (autosaved)'. The menu bar contains 'File', 'Edit', 'View', 'Insert', 'Cell', 'Kernel', 'Widgets', and 'Help'. The status bar shows 'Not Trusted' and 'Python 3 (ipykernel)'. The toolbar includes icons for saving, adding, undo, redo, and running code. The code cell contains the following Python code:

```
In [14]: def solve(n):
        if n <= 2:
            return n - 1
        else:
            return solve(n - 1) + solve(n - 2)

        n = int(input("Enter the Number:"))
        print(solve(n))
```

The output of the code is:

```
Enter the Number:8
13
```