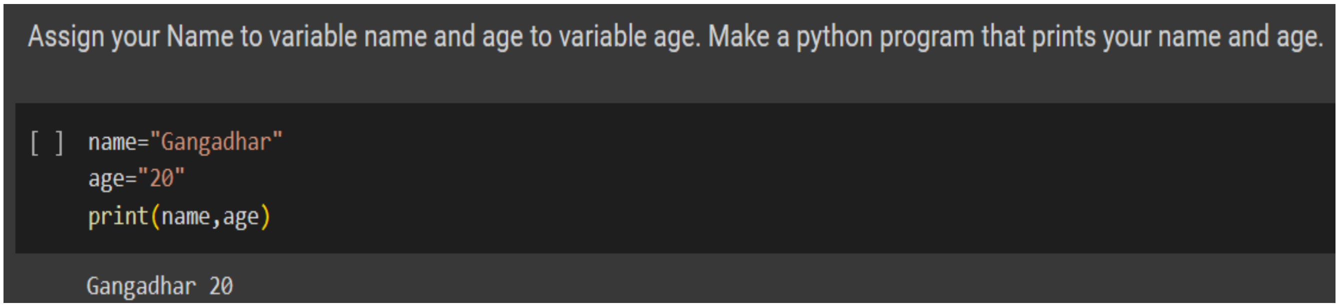
SmartBridge Applied DataScience

Assignment - 1

Name: N. Gangadhar

Reg. Num:20BCE7598

1)

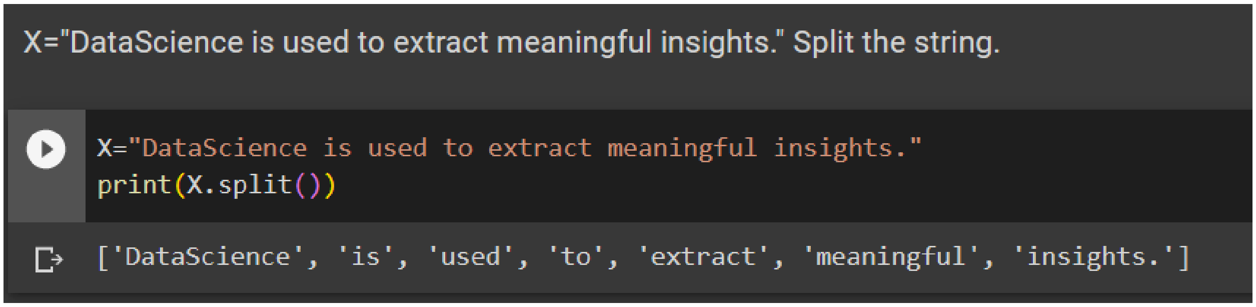


Assign your Name to variable name and age to variable age. Make a python program that prints your name and age.

```
[ ]  name="Gangadhar"
     age="20"
     print(name,age)
```

```
Gangadhar 20
```

2)

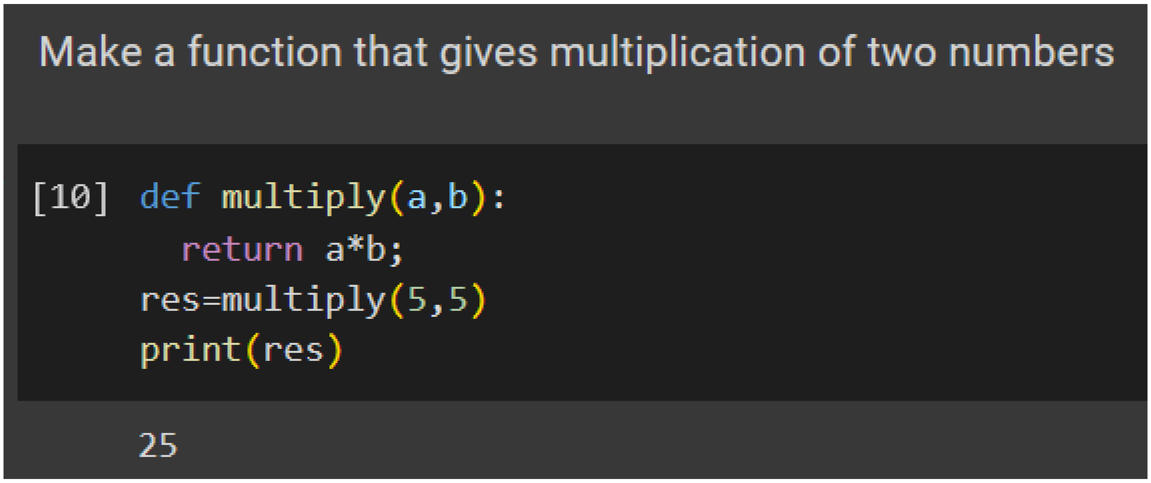


X="DataScience is used to extract meaningful insights." Split the string.

```
X="DataScience is used to extract meaningful insights."
print(X.split())
```

```
['DataScience', 'is', 'used', 'to', 'extract', 'meaningful', 'insights.']
```

3)

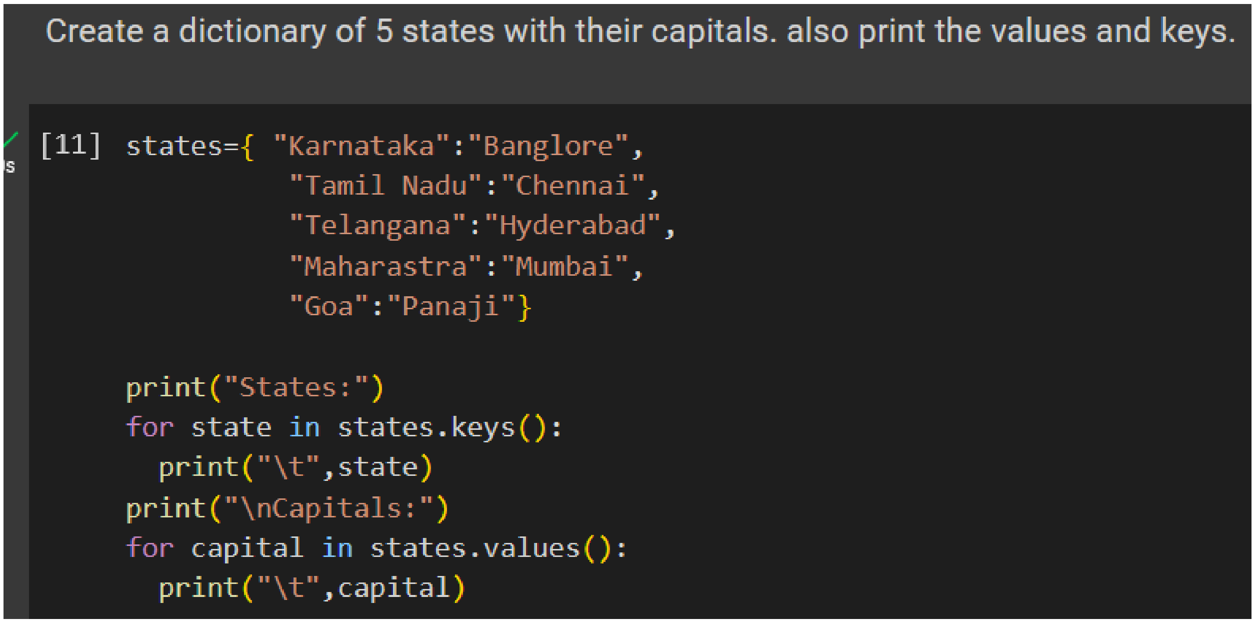


Make a function that gives multiplication of two numbers

```
[10]  def multiply(a,b):
        return a*b;
      res=multiply(5,5)
      print(res)
```

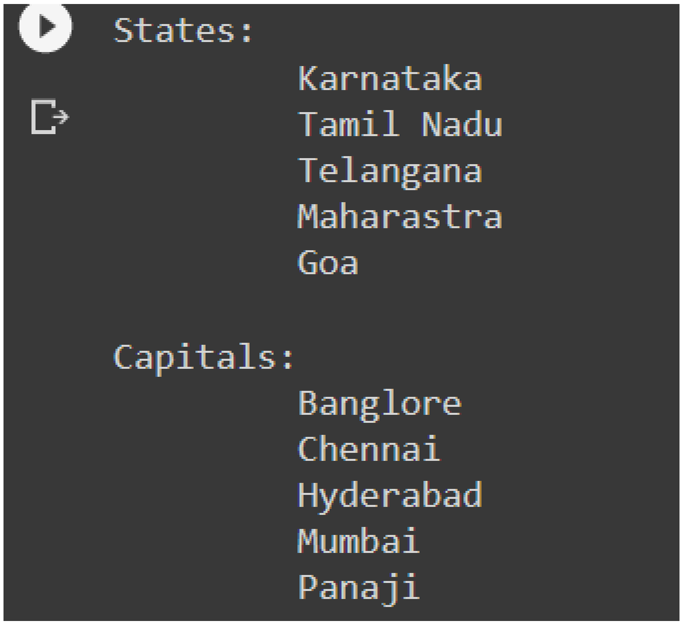
```
25
```

4)



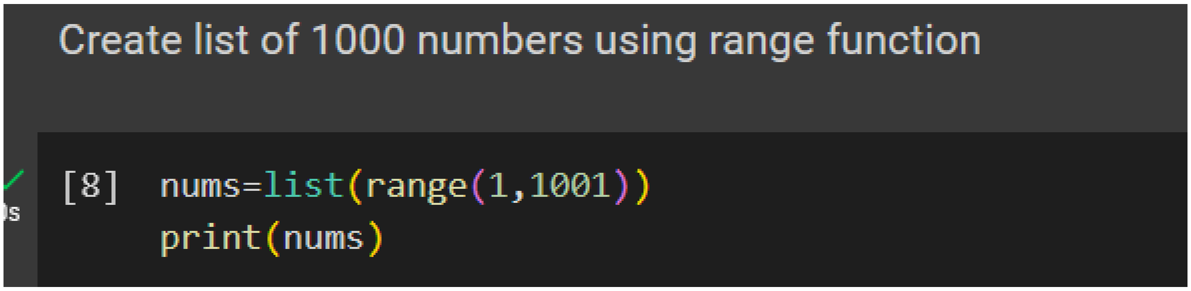
Create a dictionary of 5 states with their capitals. also print the values and keys.

```
[11] states={ "Karnataka":"Banglore",
              "Tamil Nadu":"Chennai",
              "Telangana":"Hyderabad",
              "Maharastra":"Mumbai",
              "Goa":"Panaji"}

     print("States:")
     for state in states.keys():
       print("\t",state)
     print("\nCapitals:")
     for capital in states.values():
       print("\t",capital)
```



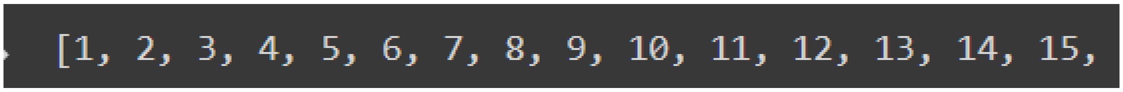
```
States:
             Karnataka
             Tamil Nadu
             Telangana
             Maharastra
             Goa

Capitals:
             Banglore
             Chennai
             Hyderabad
             Mumbai
             Panaji
```

5)

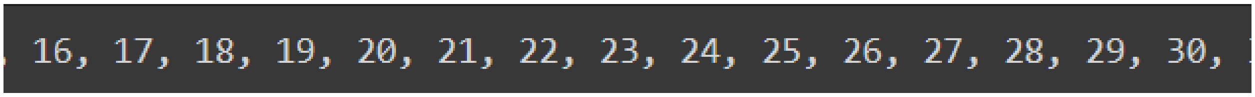


Create list of 1000 numbers using range function

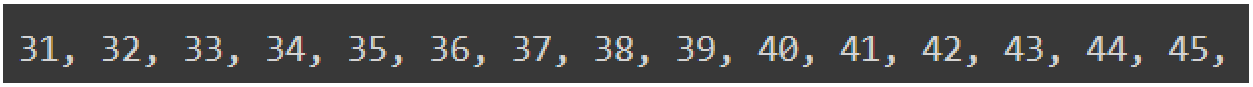
```
[8]  nums=list(range(1,1001))
     print(nums)
```



```
 [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15,
```

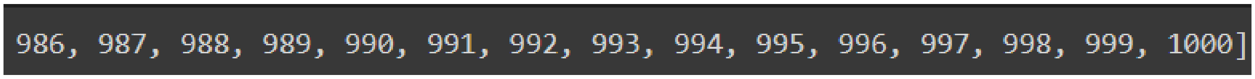


```
 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30,
```



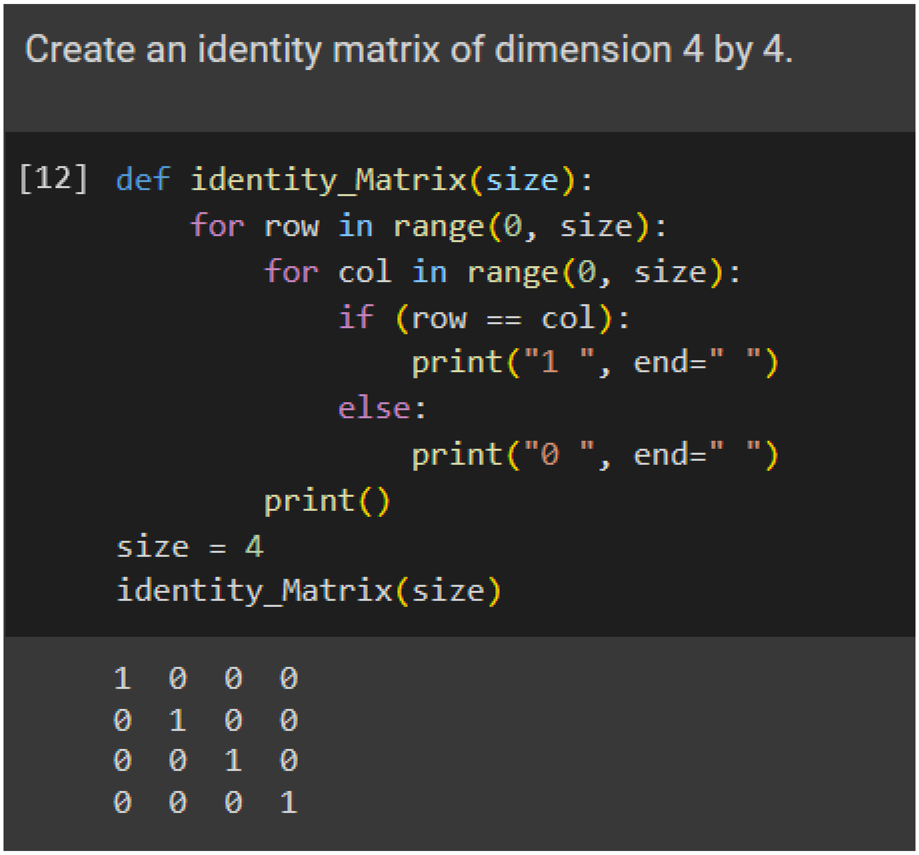
```
31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45,
```

.
.
.
.
.
.



```
986, 987, 988, 989, 990, 991, 992, 993, 994, 995, 996, 997, 998, 999, 1000]
```

6)

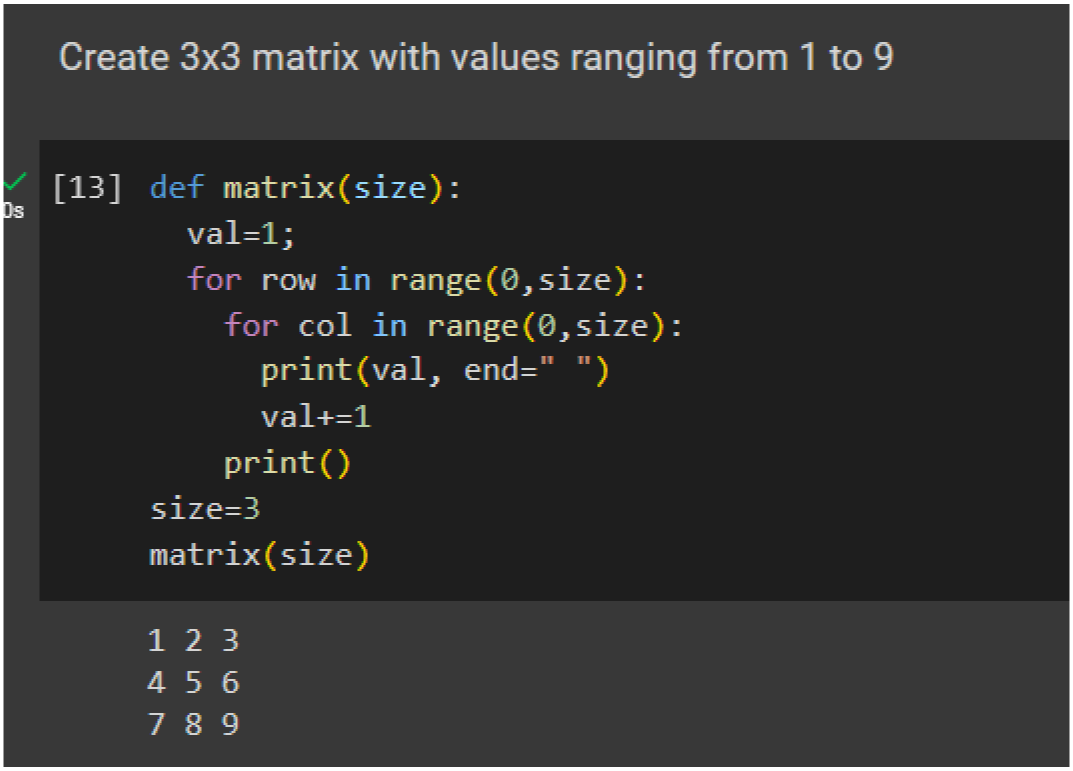


Create an identity matrix of dimension 4 by 4.

```
[12] def identity_Matrix(size):
         for row in range(0, size):
             for col in range(0, size):
                 if (row == col):
                     print("1 ", end=" ")
                 else:
                     print("0 ", end=" ")
             print()
     size = 4
     identity_Matrix(size)
```

```
1  0  0  0
0  1  0  0
0  0  1  0
0  0  0  1
```

7)

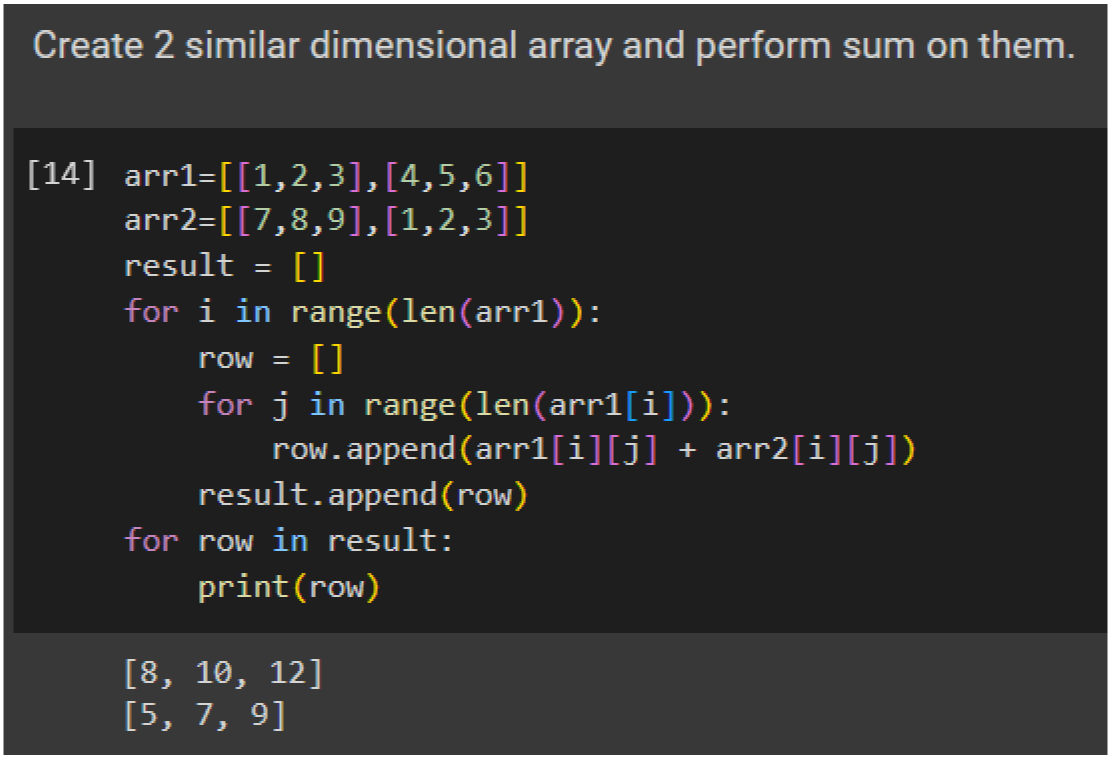


Create 3x3 matrix with values ranging from 1 to 9

```
[13] def matrix(size):
       val=1;
       for row in range(0,size):
         for col in range(0,size):
           print(val, end=" ")
           val+=1
         print()
     size=3
     matrix(size)
```

```
1 2 3
4 5 6
7 8 9
```

8)

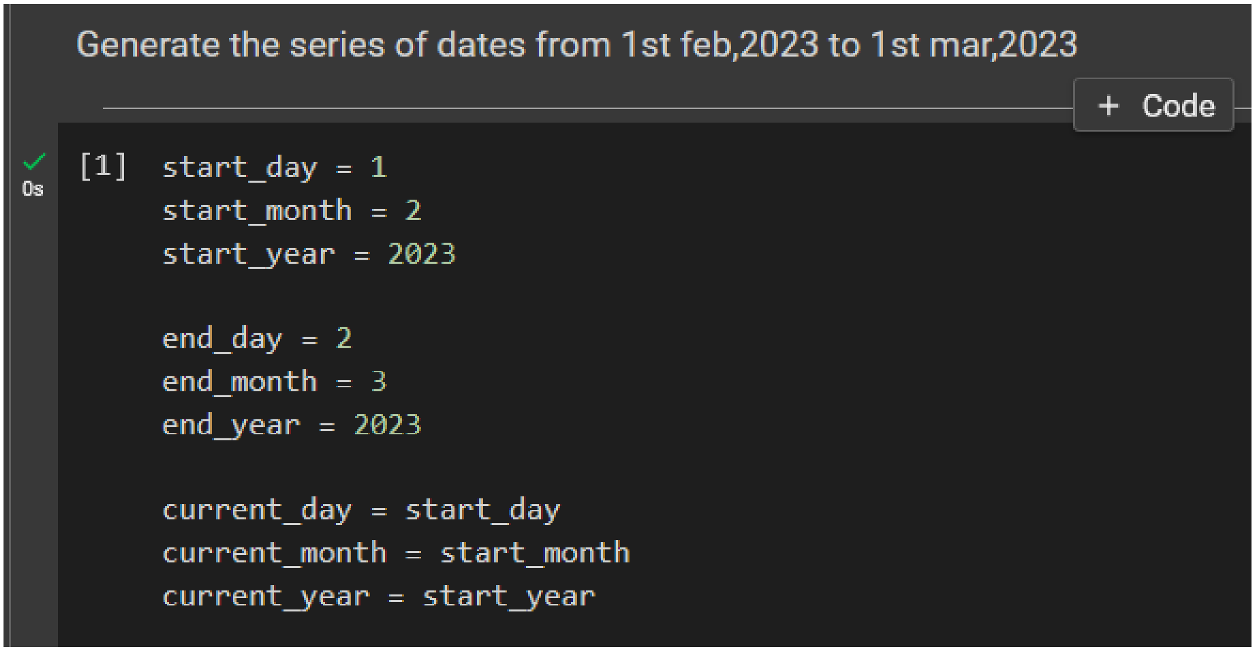


Create 2 similar dimensional array and perform sum on them.

```
[14] arr1=[[1,2,3],[4,5,6]]
     arr2=[[7,8,9],[1,2,3]]
     result = []
     for i in range(len(arr1)):
         row = []
         for j in range(len(arr1[i])):
             row.append(arr1[i][j] + arr2[i][j])
         result.append(row)
     for row in result:
         print(row)
```

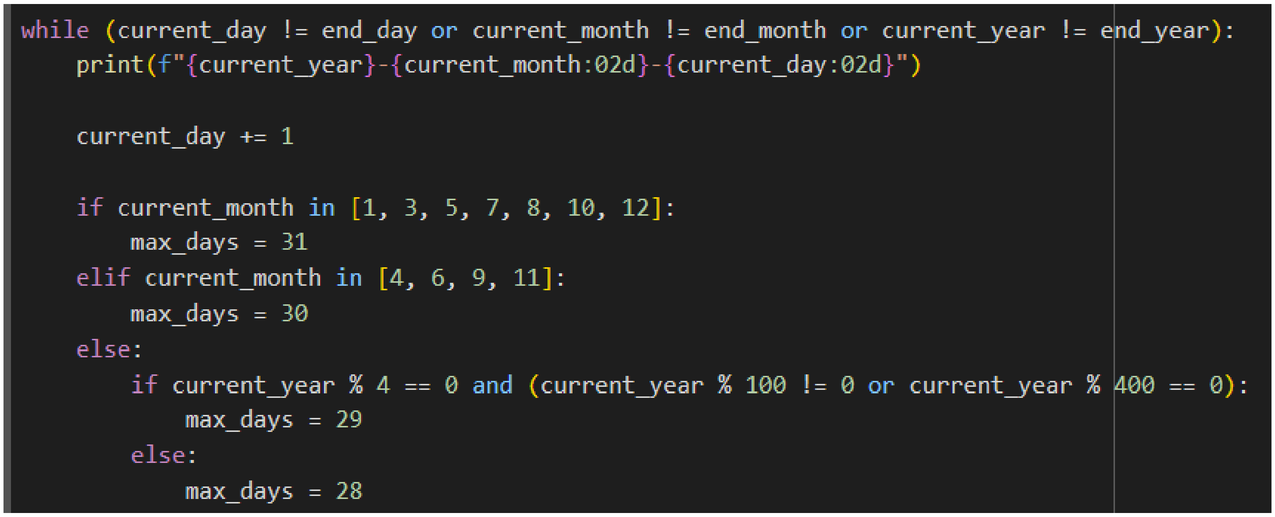
```
[8, 10, 12]
[5, 7, 9]
```

9)

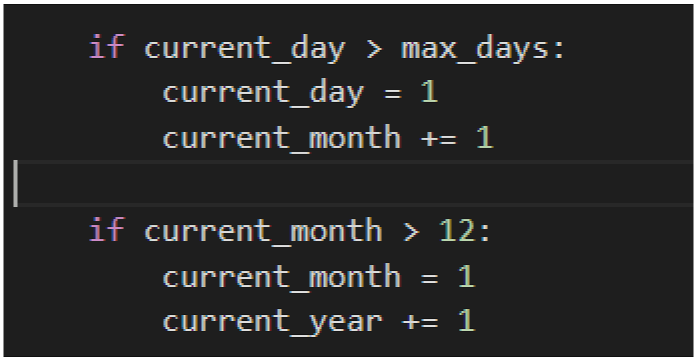


Generate the series of dates from 1st feb,2023 to 1st mar,2023

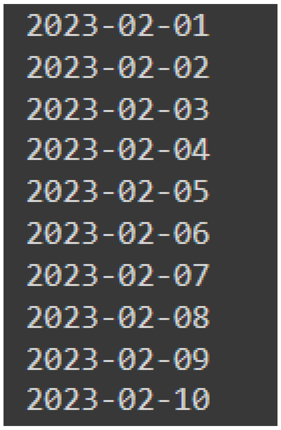
```
start_day = 1
start_month = 2
start_year = 2023

end_day = 2
end_month = 3
end_year = 2023

current_day = start_day
current_month = start_month
current_year = start_year
```



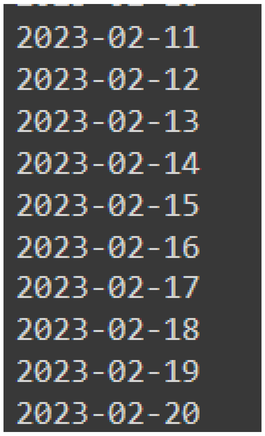
```
while (current_day != end_day or current_month != end_month or current_year != end_year):
    print(f"{current_year}-{current_month:02d}-{current_day:02d}")

    current_day += 1

    if current_month in [1, 3, 5, 7, 8, 10, 12]:
        max_days = 31
    elif current_month in [4, 6, 9, 11]:
        max_days = 30
    else:
        if current_year % 4 == 0 and (current_year % 100 != 0 or current_year % 400 == 0):
            max_days = 29
        else:
            max_days = 28
```



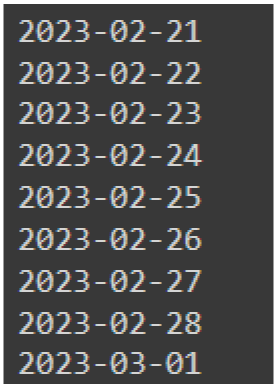
```
    if current_day > max_days:
        current_day = 1
        current_month += 1

    if current_month > 12:
        current_month = 1
        current_year += 1
```



```
2023-02-01
2023-02-02
2023-02-03
2023-02-04
2023-02-05
2023-02-06
2023-02-07
2023-02-08
2023-02-09
2023-02-10
```

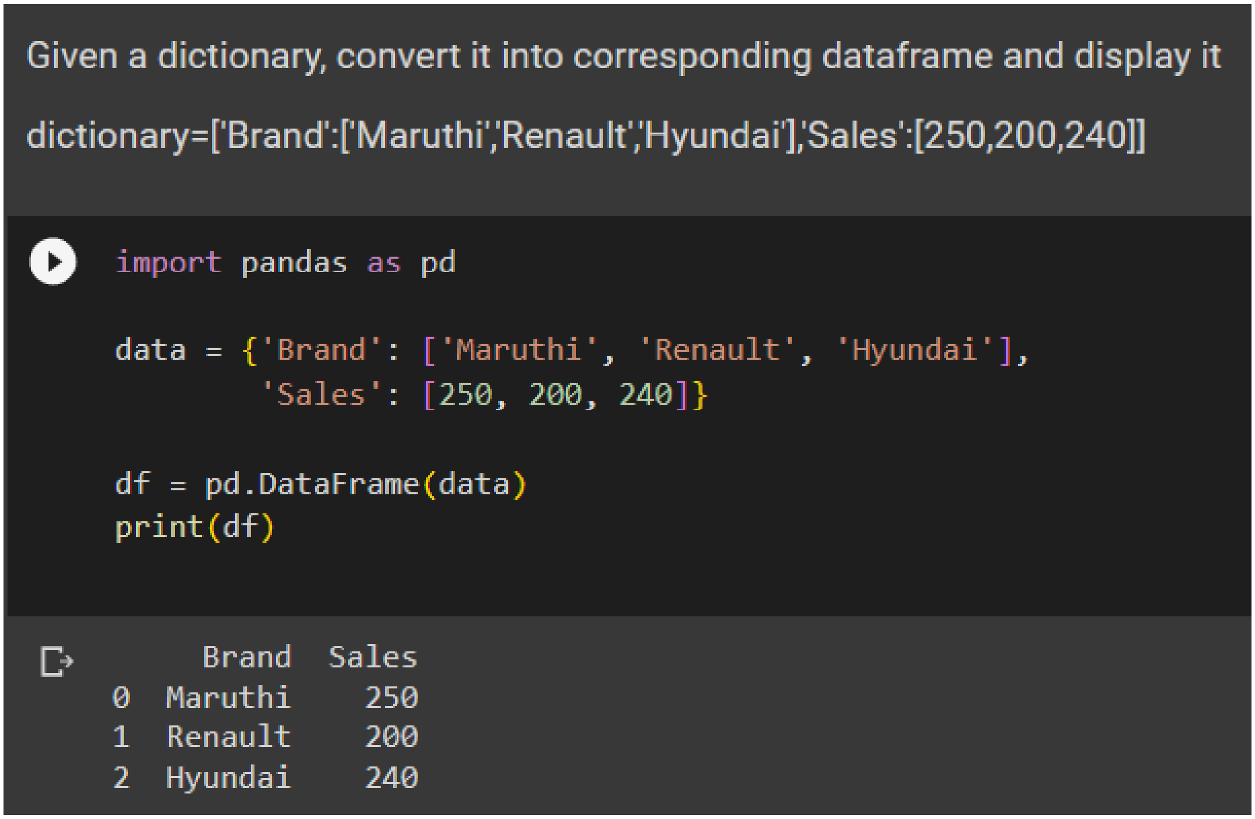


```
2023-02-11
2023-02-12
2023-02-13
2023-02-14
2023-02-15
2023-02-16
2023-02-17
2023-02-18
2023-02-19
2023-02-20
```



```
2023-02-21
2023-02-22
2023-02-23
2023-02-24
2023-02-25
2023-02-26
2023-02-27
2023-02-28
2023-03-01
```

10)



Given a dictionary, convert it into corresponding dataframe and display it

dictionary=['Brand':['Maruthi','Renault','Hyundai'],'Sales':[250,200,240]]

```
import pandas as pd

data = {'Brand': ['Maruthi', 'Renault', 'Hyundai'],
        'Sales': [250, 200, 240]}

df = pd.DataFrame(data)
print(df)
```

```
     Brand  Sales
0  Maruthi    250
1  Renault    200
2  Hyundai    240
```

GoogleColab Link:
https://colab.research.google.com/drive/11Is4NykARSrMqp3K_HFPkII0hO9_7V0D#scrollTo=JPxaP25x4CZJ