Name: Karanam Sirisha

Reg num: 20MIS0096

Karanam.sirisha2020@vitstudent.ac.in

M.Tech integrated-software engineering

ADS ASSIGNMENT 1

1. Assign your Name to variable name and Age to variable age. Make a Python program that prints your name and age.

```
[1]  Name="sirisha"
     age="20"

     print(Name)
     print(age)
```

```
sirisha
20
```

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

```
[3]  X= "Datascience is used to extract meaningful insight"
     split=X.split()
     print(split)
```

```
['Datascience', 'is', 'used', 'to', 'extract', 'meaningful', 'insight']
```

3. Make a function that gives multiplication of two numbers

```
[4]
def multiply(a,b):
  result= a*b
  return result
```

```
[5] n1=10
n2=12
mul_result= multiply(n1,n2)
print(mul_result)
```

```
120
```

4. Create a Dictionary of 5 States with their capitals. also print the keys and values.

```
print("country names:")
for country in countries_capitals:
  print(country)

print("\ncapital cities:")
for capital in countries_capitals.values():
  print(capital)

print("\ncountry-capital pairs")
for country, capital in countries_capitals.items():
  print(country, "-", capital)
```

```
country names:
india
united states
germany
japan

capital cities:
new delhi
washington, D.C
berlin
tokyo

country-capital pairs
india - new delhi
united states - washington, D.C
germany - berlin
japan - tokyo
```

5. Create an identity matrix of dimension 4 by 4

```
[10] def create_identity_matrix(n):
       matrix = [[0] * n for _ in range(n)]
       for i in range(n):
         matrix[i][i]=1
       return matrix

     identity_matrix= create_identity_matrix(4)

     for row in identity_matrix:
       print(row)
```

```
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
```

6. Create an identity matrix of dimension 4 by 4

```
[11] import numpy as np

     identity_matrix=np.eye(4)
     print(identity_matrix)
```

```
[[1. 0. 0. 0.]
 [0. 1. 0. 0.]
 [0. 0. 1. 0.]
 [0. 0. 0. 1.]]
```

7.Create a 3x3 matrix with values ranging from 1 to 9

```
[14] def create_matrix():
       matrix= [[0] * 3 for _ in range(3)]

       value=1
       for i in range(3):
         for j in range(3):
           matrix[i][j] = value
           value+=1
       return matrix

     matrix= create_matrix()

     for row in matrix:
       print(row)
```

```
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
```

8. Create 2 similar dimensional array and perform sum on them.

```
[16] def sum_array(array1, array2):
       rows= len(array1)
       cols=len(array1[0])

       result= [[0] * cols for _ in range(rows)]
       for i in range(rows):
         for j in range(cols):
           result[i][j]= array1[i][j] + array2[i][j]
       return result

     array1= [[1,2,3],[4,5,6],[7,8,9]]
     array2=[[9,8,7],[6,5,4],[3,2,1]]

     sum_result= sum_array(array1,array2)


     for row in sum_result:
       print(row)
```

```
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
```

9. Generate the series of dates from 1st Feb, 2023 to 1st March, 2023 (both inclusive)

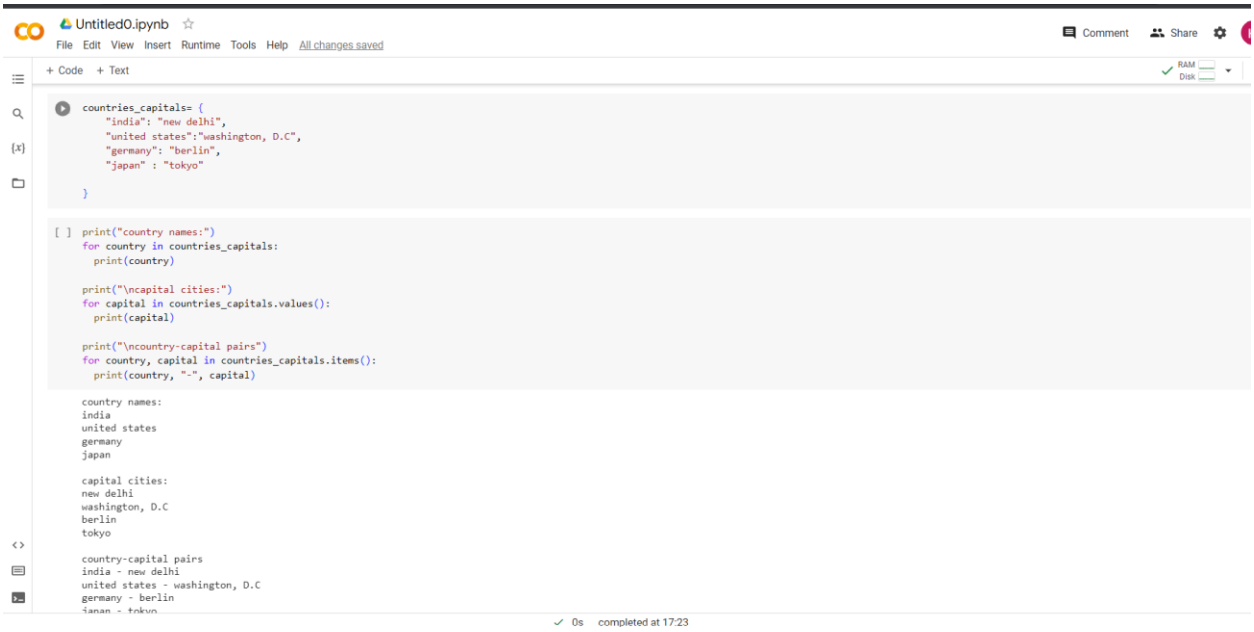
```
from datetime import datetime, timedelta

start_date = datetime(2023,2,1)
end_date = datetime(2023,3,1)

current_date = start_date
while current_date <= end_date:
  print(current_date.strftime("%Y-%m-%Dd"))
  current_date+= timedelta(days=1)
```

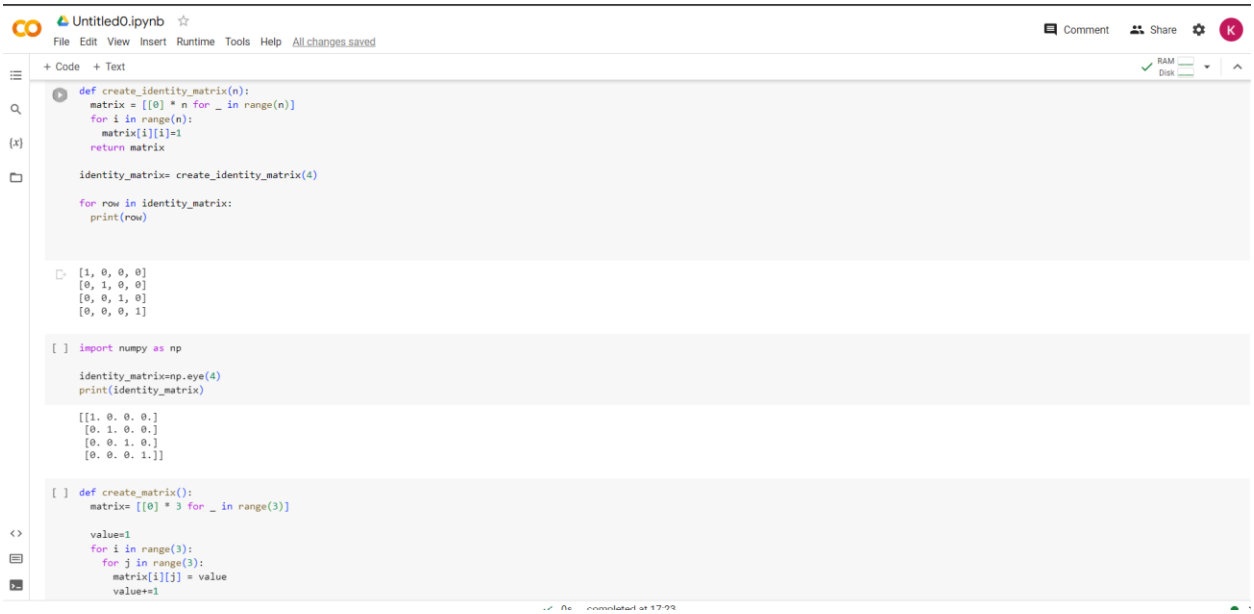
```
2023-02-02/01/23d
2023-02-02/02/23d
2023-02-02/03/23d
2023-02-02/04/23d
2023-02-02/05/23d
2023-02-02/06/23d
2023-02-02/07/23d
2023-02-02/08/23d
2023-02-02/09/23d
2023-02-02/10/23d
2023-02-02/11/23d
2023-02-02/12/23d
2023-02-02/13/23d
2023-02-02/14/23d
2023-02-02/15/23d
2023-02-02/16/23d
2023-02-02/17/23d
2023-02-02/18/23d
2023-02-02/19/23d
2023-02-02/20/23d
2023-02-02/21/23d
2023-02-02/22/23d
```

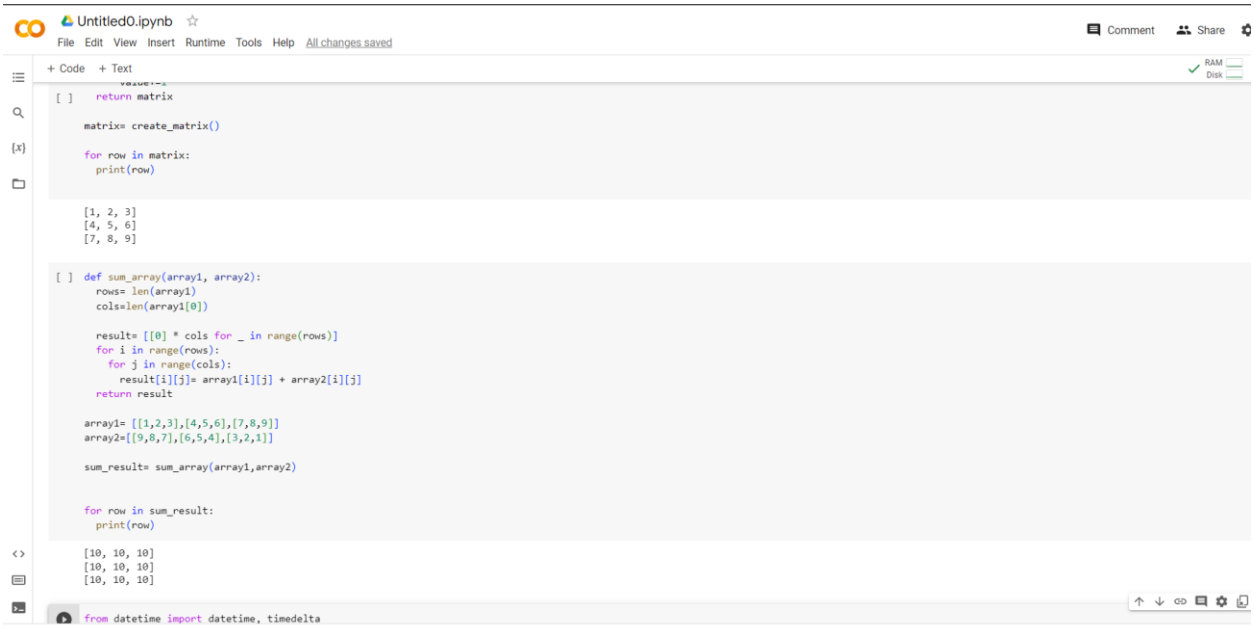
10.Given a dictionary, convert it into corresponding dataframe and display it dictionary = {'Brand': ['Maruti', 'Renault', 'Hyndai'], 'Sales' : [250, 200, 240]}

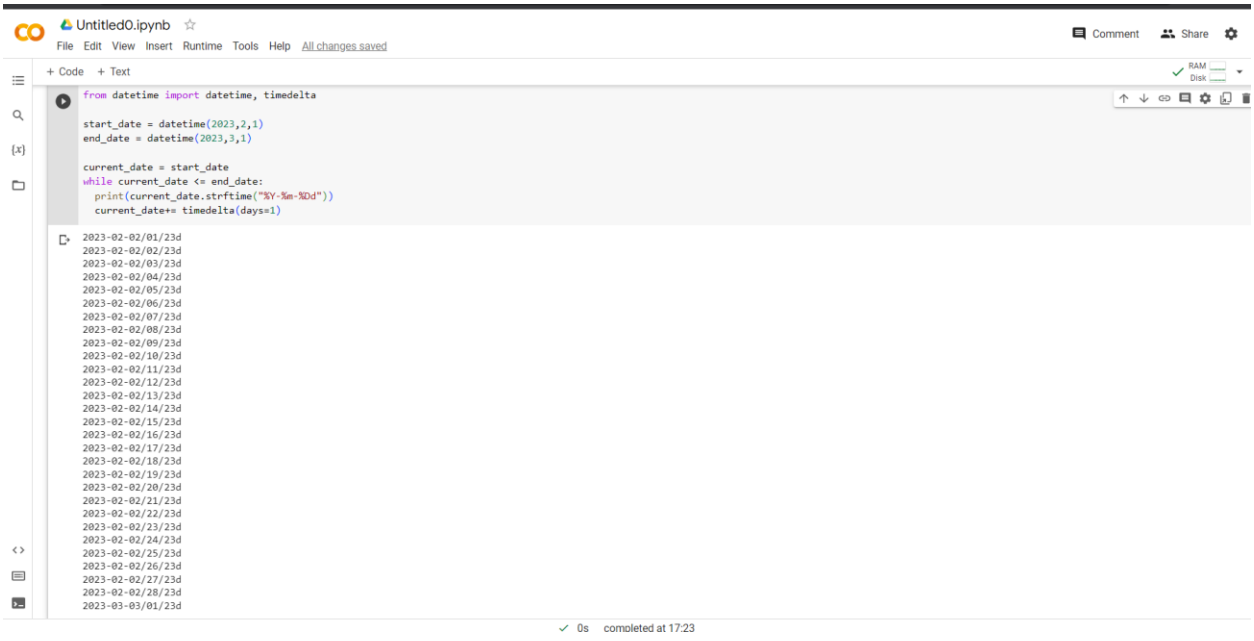
```
[1] import pandas as pd
    dictionary= {'brand': ['maruti','renault', 'hyundai'],'sales':[250,200,240]}
    df= pd.DataFrame(dictionary)
    print(df)
```

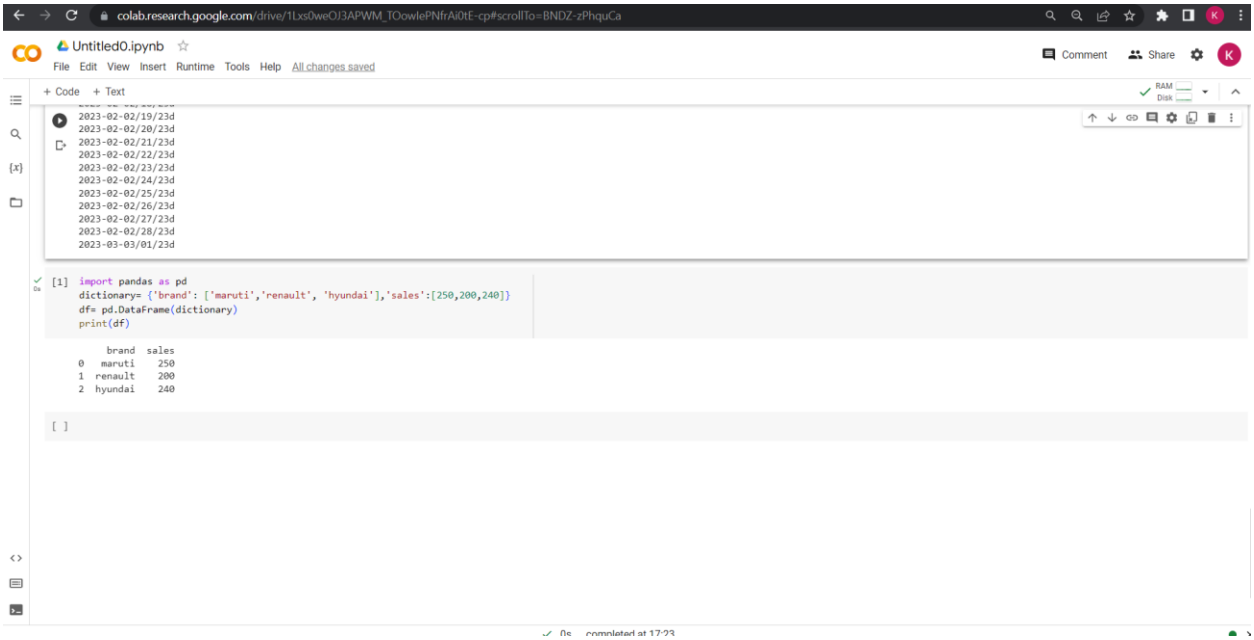
```
     brand  sales
0   maruti    250
1  renault    200
2  hyundai    240
```


Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
countries_capitals= {
"india": "new delhi",
"united states":"washington, D.C",
"germany": "berlin",
"japan" : "tokyo"
}
[ ] print("country names:")
for country in countries_capitals:
print(country)
print("\ncapital cities:")
for capital in countries_capitals.values():
print(capital)
print("\ncountry-capital pairs")
for country, capital in countries_capitals.items():
print(country, "-", capital)
country names:
india
united states
germany
japan
capital cities:
new delhi
washington, D.C
berlin
tokyo
country-capital pairs
india - new delhi
united states - washington, D.C
germany - berlin
0s completed at 17:23


Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
K
+ Code + Text
RAM
Disk
def create_identity_matrix(n):
matrix = [[0] * n for _ in range(n)]
for i in range(n):
matrix[i][i]=1
return matrix
identity_matrix= create_identity_matrix(4)
for row in identity_matrix:
print(row)
[1, 0, 0, 0]
[0, 1, 0, 0]
[0, 0, 1, 0]
[0, 0, 0, 1]
[ ] import numpy as np
identity_matrix=np.eye(4)
print(identity_matrix)
[[1. 0. 0. 0.]
[0. 1. 0. 0.]
[0. 0. 1. 0.]
[0. 0. 0. 1.]]
[ ] def create_matrix():
matrix= [[0] * 3 for _ in range(3)]
value=1
for i in range(3):
for j in range(3):
matrix[i][j] = value
value+=1


Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
return matrix
matrix= create_matrix()
for row in matrix:
print(row)
[1, 2, 3]
[4, 5, 6]
[7, 8, 9]
def sum_array(array1, array2):
rows= len(array1)
cols=len(array1[0])
result= [[0] * cols for _ in range(rows)]
for i in range(rows):
for j in range(cols):
result[i][j]= array1[i][j] + array2[i][j]
return result
array1= [[1,2,3],[4,5,6],[7,8,9]]
array2=[[9,8,7],[6,5,4],[3,2,1]]
sum_result= sum_array(array1,array2)
for row in sum_result:
print(row)
[10, 10, 10]
[10, 10, 10]
[10, 10, 10]
from datetime import datetime, timedelta
0s completed at 17:23


Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
from datetime import datetime, timedelta
start_date = datetime(2023,2,1)
end_date = datetime(2023,3,1)
current_date = start_date
while current_date <= end_date:
print(current_date.strftime("%Y-%m-%Dd"))
current_date+= timedelta(days=1)
2023-02-02/01/23d
2023-02-02/02/23d
2023-02-02/03/23d
2023-02-02/04/23d
2023-02-02/05/23d
2023-02-02/06/23d
2023-02-02/07/23d
2023-02-02/08/23d
2023-02-02/09/23d
2023-02-02/10/23d
2023-02-02/11/23d
2023-02-02/12/23d
2023-02-02/13/23d
2023-02-02/14/23d
2023-02-02/15/23d
2023-02-02/16/23d
2023-02-02/17/23d
2023-02-02/18/23d
2023-02-02/19/23d
2023-02-02/20/23d
2023-02-02/21/23d
2023-02-02/22/23d
2023-02-02/23/23d
2023-02-02/24/23d
2023-02-02/25/23d
2023-02-02/26/23d
2023-02-02/27/23d
2023-02-02/28/23d
2023-03-03/01/23d
0s completed at 17:23


colab.research.google.com/drive/1Lxs0weOJ3APWM_TOowIePNfrAi0tE-cp#scrollTo=BNDZ-zPhquCa
Untitled0.ipynb
File Edit View Insert Runtime Tools Help All changes saved
Comment
Share
+ Code + Text
RAM
Disk
2023-02-02/19/23d
2023-02-02/20/23d
2023-02-02/21/23d
2023-02-02/22/23d
2023-02-02/23/23d
2023-02-02/24/23d
2023-02-02/25/23d
2023-02-02/26/23d
2023-02-02/27/23d
2023-02-02/28/23d
2023-03-03/01/23d
[1] import pandas as pd
dictionary= {'brand': ['maruti','renault', 'hyundai'],'sales':[250,200,240]}
df= pd.DataFrame(dictionary)
print(df)
brand sales
0 maruti 250
1 renault 200
2 hyundai 240
[ ]