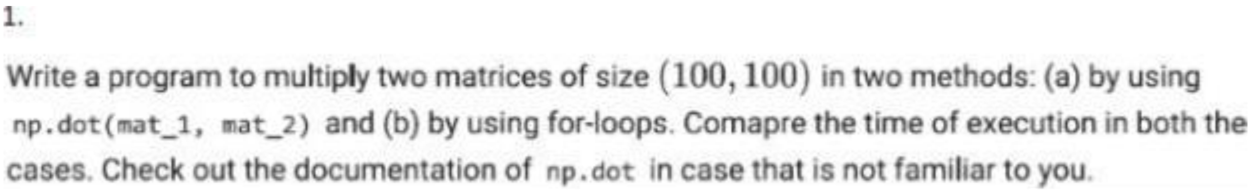
LAB ASSIGNMENT – 3

NAME : PRATHAPANI SATWIKA

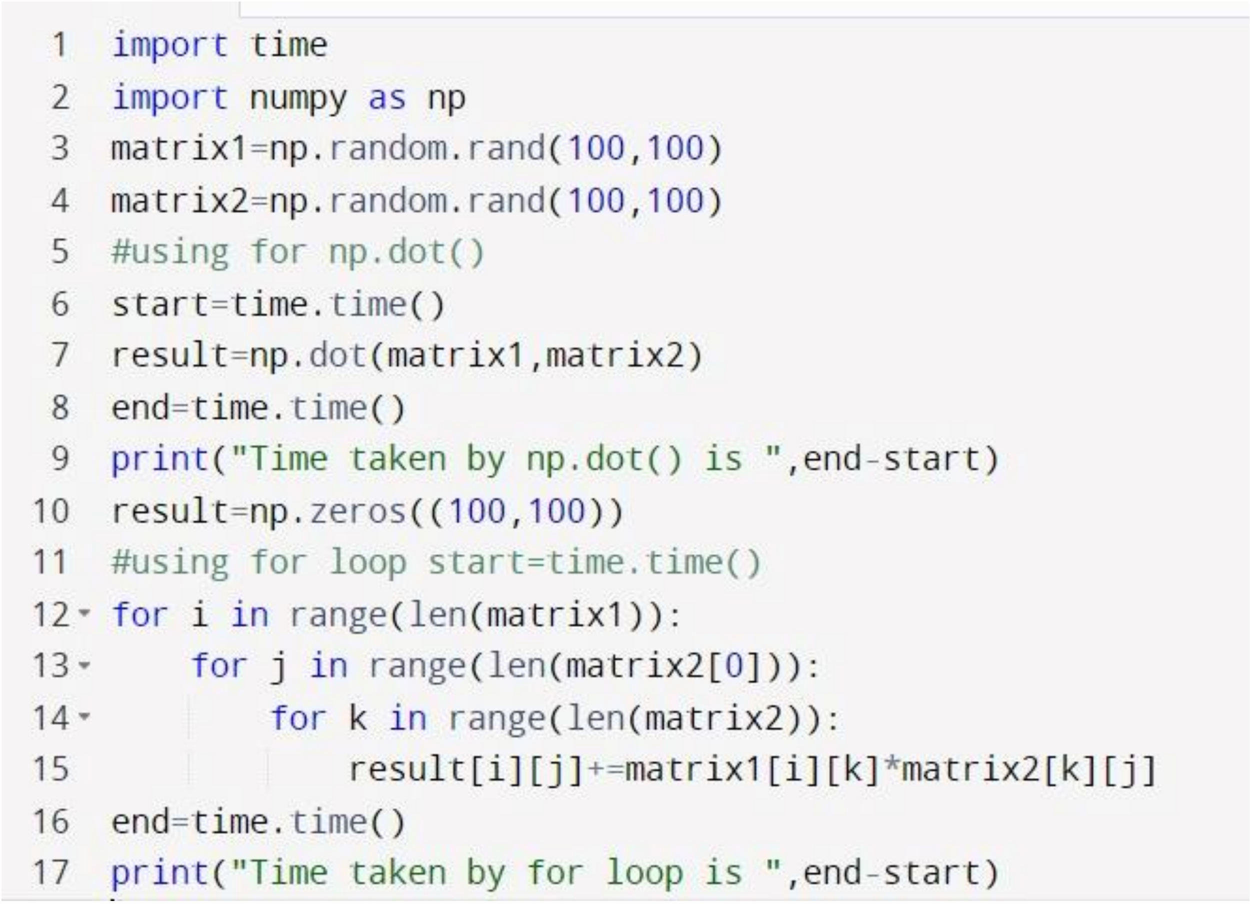
REG.NO. : 20BCD7160



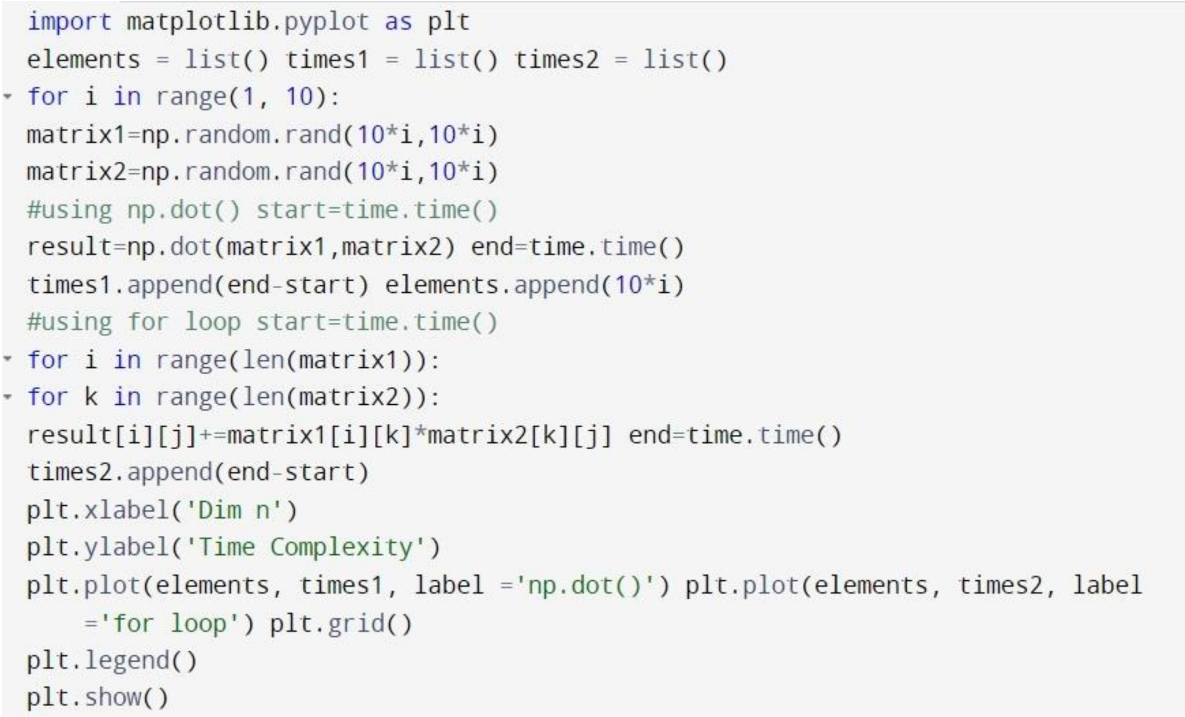
1.

Write a program to multiply two matrices of size $(100, 100)$ in two methods: (a) by using `np.dot(mat_1, mat_2)` and (b) by using for-loops. Comapre the time of execution in both the cases. Check out the documentation of `np.dot` in case that is not familiar to you.

CODE :

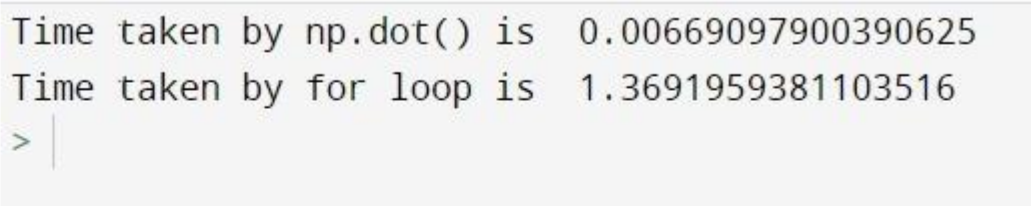


```
import time
import numpy as np
matrix1=np.random.rand(100,100)
matrix2=np.random.rand(100,100)
#using for np.dot()
start=time.time()
result=np.dot(matrix1,matrix2)
end=time.time()
print("Time taken by np.dot() is ",end-start)
result=np.zeros((100,100))
#using for loop start=time.time()
for i in range(len(matrix1)):
    for j in range(len(matrix2[0])):
        for k in range(len(matrix2)):
            result[i][j]+=matrix1[i][k]*matrix2[k][j]
end=time.time()
print("Time taken by for loop is ",end-start)
```

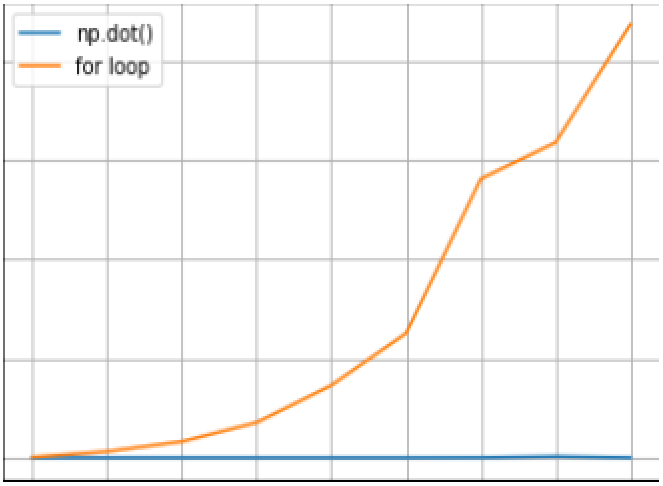


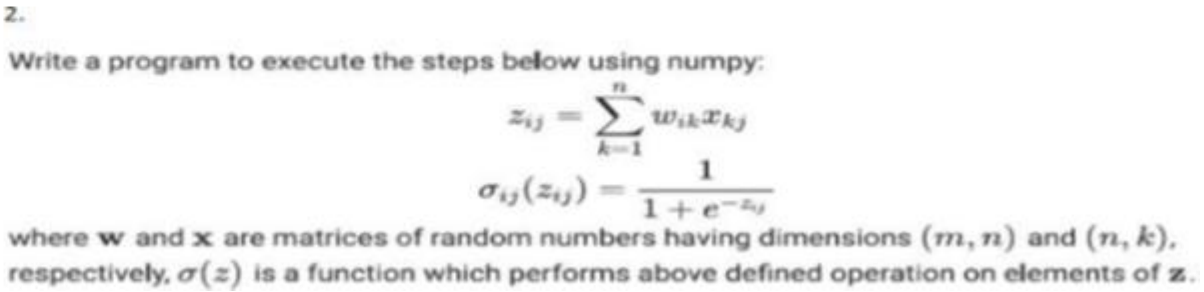
```
import matplotlib.pyplot as plt
elements = list() times1 = list() times2 = list()
for i in range(1, 10):
matrix1=np.random.rand(10*i,10*i)
matrix2=np.random.rand(10*i,10*i)
#using np.dot() start=time.time()
result=np.dot(matrix1,matrix2) end=time.time()
times1.append(end-start) elements.append(10*i)
#using for loop start=time.time()
for i in range(len(matrix1)):
for k in range(len(matrix2)):
result[i][j]+=matrix1[i][k]*matrix2[k][j] end=time.time()
times2.append(end-start)
plt.xlabel('Dim n')
plt.ylabel('Time Complexity')
plt.plot(elements, times1, label ='np.dot()') plt.plot(elements, times2, label
    ='for loop') plt.grid()
plt.legend()
plt.show()
```

OUTPUT :



```
Time taken by np.dot() is  0.00669097900390625
Time taken by for loop is  1.3691959381103516
>
```





2.

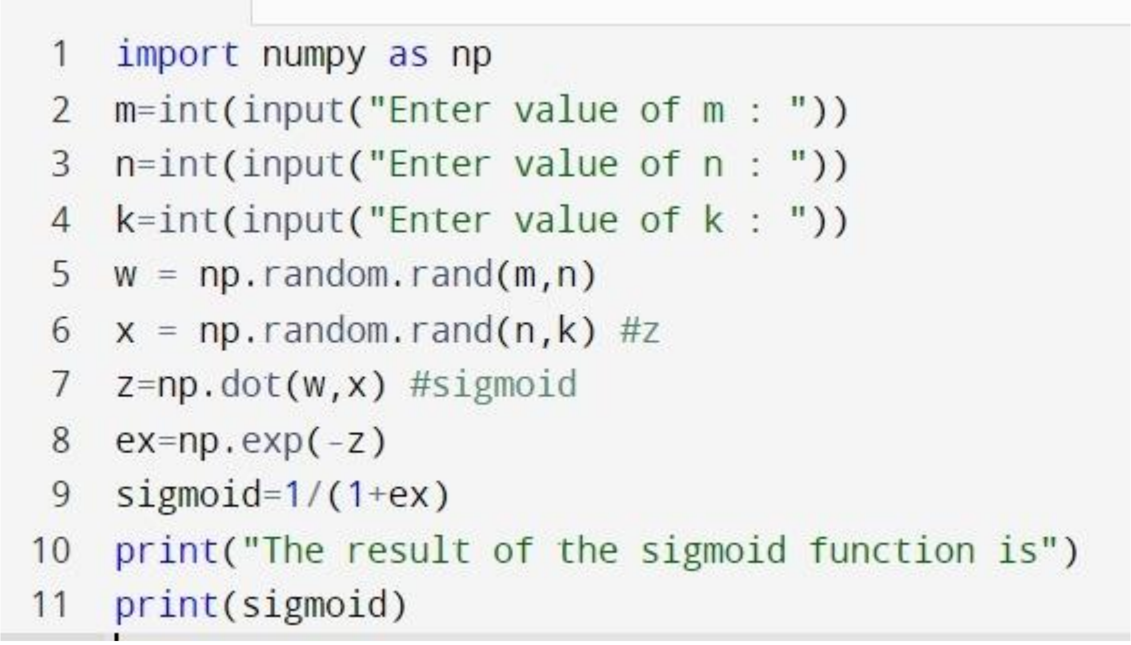
Write a program to execute the steps below using numpy:

$$z_{ij} = \sum_{k=1}^{n} w_{ik} x_{kj}$$

$$\sigma_{ij}(z_{ij}) = \frac{1}{1 + e^{-z_{ij}}}$$

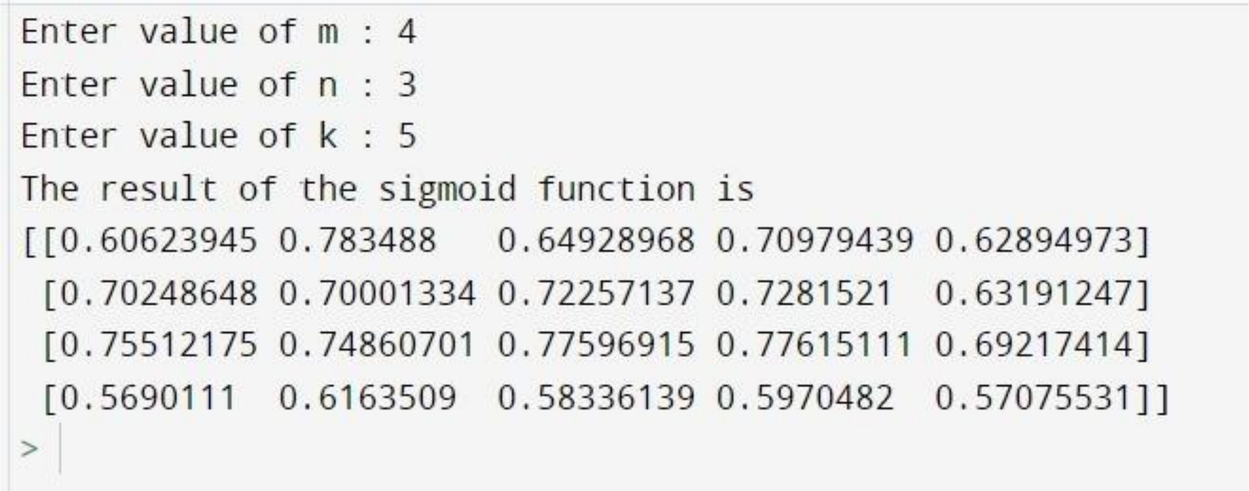
where **w** and **x** are matrices of random numbers having dimensions (m, n) and (n, k), respectively, $\sigma(z)$ is a function which performs above defined operation on elements of **z**.

CODE :

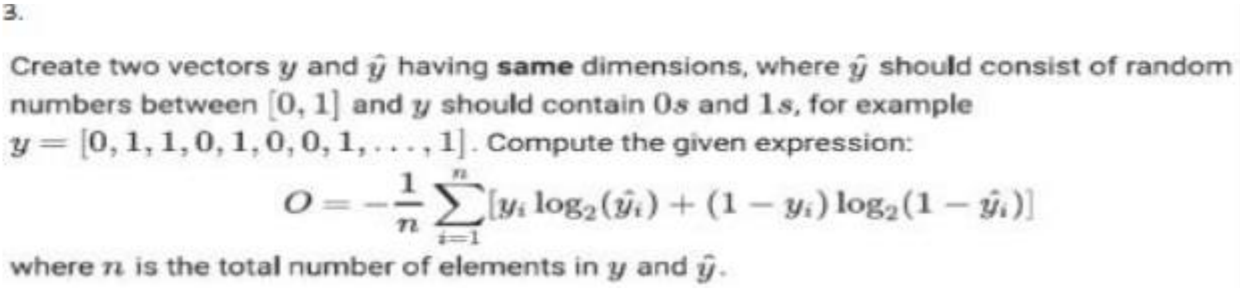


```
import numpy as np
m=int(input("Enter value of m : "))
n=int(input("Enter value of n : "))
k=int(input("Enter value of k : "))
w = np.random.rand(m,n)
x = np.random.rand(n,k) #z
z=np.dot(w,x) #sigmoid
ex=np.exp(-z)
sigmoid=1/(1+ex)
print("The result of the sigmoid function is")
print(sigmoid)
```

OUTPUT :



```
Enter value of m : 4
Enter value of n : 3
Enter value of k : 5
The result of the sigmoid function is
[[0.60623945 0.783488   0.64928968 0.70979439 0.62894973]
 [0.70248648 0.70001334 0.72257137 0.7281521  0.63191247]
 [0.75512175 0.74860701 0.77596915 0.77615111 0.69217414]
 [0.5690111  0.6163509  0.58336139 0.5970482  0.57075531]]
>
```

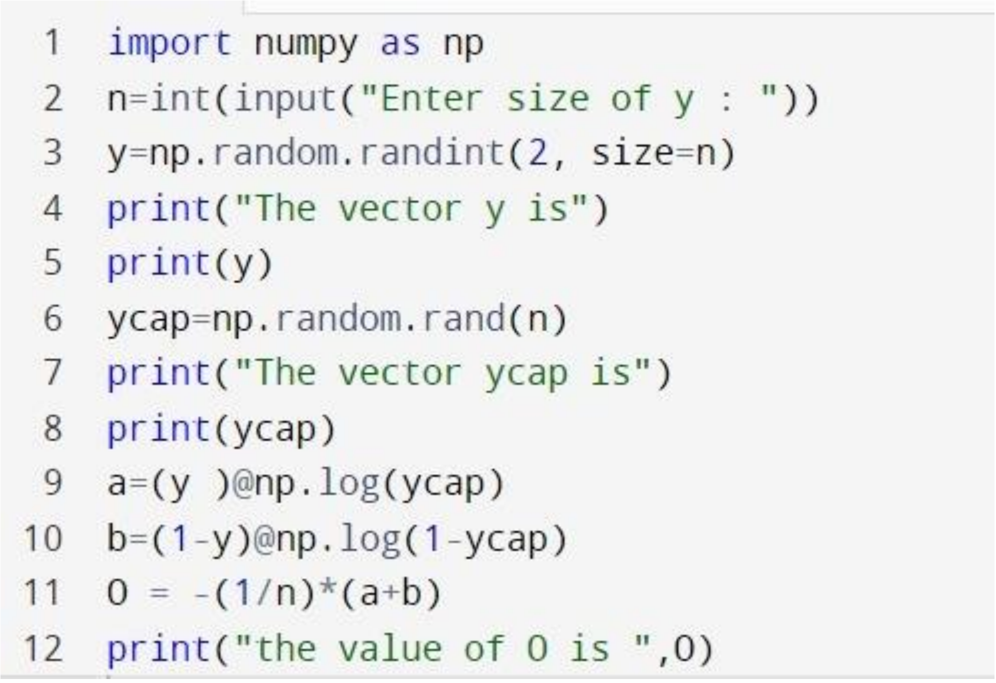


3.

Create two vectors y and $\hat{y}$ having **same** dimensions, where $\hat{y}$ should consist of random numbers between $[0, 1]$ and y should contain $0s$ and $1s$, for example $y = [0, 1, 1, 0, 1, 0, 0, 1, \ldots, 1]$. Compute the given expression:

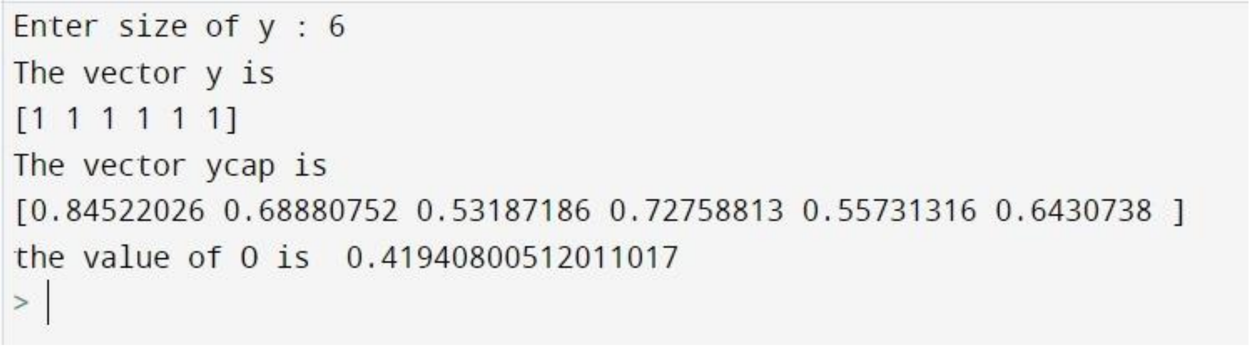
$$O = -\frac{1}{n}\sum_{i=1}^{n}[y_i \log_2(\hat{y_i}) + (1 - y_i)\log_2(1 - \hat{y_i})]$$

where n is the total number of elements in y and $\hat{y}$.



```python
import numpy as np
n=int(input("Enter size of y : "))
y=np.random.randint(2, size=n)
print("The vector y is")
print(y)
ycap=np.random.rand(n)
print("The vector ycap is")
print(ycap)
a=(y )@np.log(ycap)
b=(1-y)@np.log(1-ycap)
O = -(1/n)*(a+b)
print("the value of O is ",O)
```

OUTPUT :



```
Enter size of y : 6
The vector y is
[1 1 1 1 1 1]
The vector ycap is
[0.84522026 0.68880752 0.53187186 0.72758813 0.55731316 0.6430738 ]
the value of O is  0.41940800512011017
> 
```