Question **1**

Correct

Mark 1.00 out of 1.00

Write the algorithm for QR decomposition using the Gram-Schmidt method.

For example:

Input	Result
([[1, 1, 0], [1,0,1], [0, 1, 1]])	The Q Matrix is [[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] The R Matrix is [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]

Answer: (penalty regime: 0. %)

Reset answer

Ace editor not ready. Perhaps reload page?
Falling back to raw text area.

```
'''
Program to QR decomposition using the Gram-Schmidt method
Developed by: your name
RegisterNumber:
'''
import numpy as np
def QR_Decomposition(A):
    n,m=A.shape
    Q=np.empty((n,m))
    u=np.empty((n,m))
    R=np.zeros((n,m))
    u[:,0]=A[:,0]
    Q[:,0]=u[:,0]/np.linalg.norm(u[:,0])
    for i in range(1,n):
        u[:,i]=A[:,i]
        for j in range(n):
            u[:,i]-=(A[:,i]@Q[:,j])*Q[:,j]
        Q[:,i]=u[:,i]/np.linalg.norm(u[:,i])
```

	Input	Expected	Got	
✓	([[1, 1, 0], [1,0,1], [0, 1, 1]])	The Q Matrix is [[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] The R Matrix is [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]	The Q Matrix is [[0.70710678 0.40824829 -0.57735027] [0.70710678 -0.40824829 0.57735027] [0. 0.81649658 0.57735027]] The R Matrix is [[1.41421356 0.70710678 0.70710678] [0. 1.22474487 0.40824829] [0. 0. 1.15470054]]	✓

	Input	Expected	Got	
✓	([[12, -51, 4], [6, 167, -68], [-4, 24, -41]])	The Q Matrix is [[0.85714286 -0.39428571 -0.33142857] [0.42857143 0.90285714 0.03428571] [-0.28571429 0.17142857 -0.94285714]] The R Matrix is [[14. 21. -14.] [0. 175. -70.] [0. 0. 35.]]	The Q Matrix is [[0.85714286 -0.39428571 -0.33142857] [0.42857143 0.90285714 0.03428571] [-0.28571429 0.17142857 -0.94285714]] The R Matrix is [[14. 21. -14.] [0. 175. -70.] [0. 0. 35.]]	✓

Passed all tests! ✓

▸ Show/hide question author's solution (Python3)

Correct

Marks for this submission: 1.00/1.00.