

1. My Python Code:

```

1 # Hsuan-You Lin Module 10 Problem Set Question 1.
2 import numpy as np
3
4 def lower_triangular_system(Lower_Matrix, b_Matrix):
5     for i in range(len(Lower_Matrix)):
6         b_Matrix[i] = b_Matrix[i] / Lower_Matrix[i][i]
7         Lower_Matrix[i][i] = 1
8         for j in range(i + 1, len(Lower_Matrix)):
9             b_Matrix[j] -= Lower_Matrix[j][i] * b_Matrix[i]
10            Lower_Matrix[j][i] = 0
11            x_Matrix = b_Matrix
12    return x_Matrix
13
14 if __name__ == "__main__":
15     Lower_Matrix = np.array([[1, 0, 0],
16                             [4, 1, 0],
17                             [-6, 5, 1]])
18     b_Matrix = np.array([3, 14, -7])
19     print("Q1: Solve the lower triangular system: Lx = b.")
20     print("L = \n", Lower_Matrix)
21     print("b = ", b_Matrix)
22     ans = lower_triangular_system(Lower_Matrix, b_Matrix)
23     print("x = ", ans)
24

```

Module12 — -bash — 80x24

```

(base) pisces:Module12 pisces$ python Q1.py
Q1: Solve the lower triangular system: Lx = b.
L =
[[ 1  0  0]
 [ 4  1  0]
 [-6  5  1]]
b = [ 3 14 -7]
x = [ 3  2  1]
(base) pisces:Module12 pisces$

```

2. My Python Code:

```

1 # Hsuan-You Lin Module 10 Problem Set Question 2.
2 import numpy as np
3
4 def upper_triangular_system(Upper_Matrix, b_Matrix):
5     for i in range(len(Upper_Matrix)-1, -1, -1):
6         b_Matrix[i] = b_Matrix[i] / Upper_Matrix[i][i]
7         Upper_Matrix[i][i] = 1
8         for j in range(i):
9             b_Matrix[j] -= Upper_Matrix[j][i] * b_Matrix[i]
10            Upper_Matrix[j][i] = 0
11            x_Matrix = b_Matrix
12    return x_Matrix
13
14 if __name__ == "__main__":
15     Upper_Matrix = np.array([[1, 2, 1],
16                             [0, 1, 4],
17                             [0, 0, 3]])
18     b_Matrix = np.array([5, 5, 3])
19     print("Q2: Solve the upper triangular system: Ux = b.")
20     print("U = \n", Upper_Matrix)
21     print("b = ", b_Matrix)
22     ans = upper_triangular_system(Upper_Matrix, b_Matrix)
23     print("x = ", ans)
24

```

Module12 — -bash — 80x24

```

(base) pisces:Module12 pisces$ python Q2.py
Q2: Solve the upper triangular system: Ux = b.
U =
[[1 2 1]
 [0 1 4]
 [0 0 3]]
b = [ 5  5  3]
x = [ 2  1  1]
(base) pisces:Module12 pisces$

```

3. My Step:

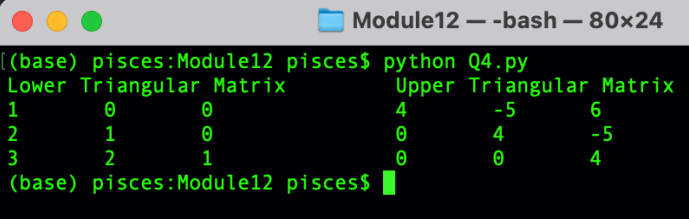
$$\begin{array}{l} L = \begin{bmatrix} 1, & 0, & 0 \\ 4, & 1, & 0 \\ -6, & 5, & 1 \end{bmatrix} \quad x = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad b = \begin{bmatrix} 3 \\ 14 \\ -7 \end{bmatrix} \quad \rightarrow \quad \text{Answer } x = \begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix} \end{array}$$

Each iteration for this lower triangular matrix will be:

Iteration 1:			Iteration 2:			Iteration 3:	
L:	b:		L:	b:		L:	b:
$\begin{bmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ 0, & 5, & 1 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 2 \\ 11 \end{bmatrix}$	\rightarrow	$\begin{bmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ 0, & 0, & 1 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$	\rightarrow	$\begin{bmatrix} 1, & 0, & 0 \\ 0, & 1, & 0 \\ 0, & 0, & 1 \end{bmatrix}$	$\begin{bmatrix} 3 \\ 2 \\ 1 \end{bmatrix}$

4. My Python code:

```
1 # Hsuan-You Lin Module 10 Problem Set Question 4.
2 import numpy as np
3
4 def LU_decomposition(A_Matrix, n):
5     Lower_Matrix = [[0 for x in range(n)]
6                     for y in range(n)]
7     Upper_Matrix = [[0 for x in range(n)]
8                     for y in range(n)]
9
10    for i in range(n):
11        for k in range(i, n):
12            LU = 0
13            for j in range(i):
14                LU += (Lower_Matrix[i][j] * Upper_Matrix[j][k])
15            Upper_Matrix[i][k] = A_Matrix[i][k] - LU
16        for k in range(i, n):
17            if (i == k):
18                Lower_Matrix[i][i] = 1
19            else:
20                LU = 0
21                for j in range(i):
22                    LU += (Lower_Matrix[k][j] * Upper_Matrix[j][i])
23
24                Lower_Matrix[k][i] = int((A_Matrix[k][i] - LU) /
25                                         Upper_Matrix[i][i])
26    print("Lower Triangular Matrix \tUpper Triangular Matrix")
27    for i in range(n):
28        for j in range(n):
29            print(Lower_Matrix[i][j], end="\t")
30        print("", end="\t")
31        for j in range(n):
32            print(Upper_Matrix[i][j], end="\t")
33        print("")
34    return Lower_Matrix, Upper_Matrix
35
36 if __name__ == "__main__":
37     A_Matrix = np.array([[4, -5, 6],
38                         [2, 4, -5],
39                         [3, 8, -6]])
40     LU_decomposition(A_Matrix, 3)
```

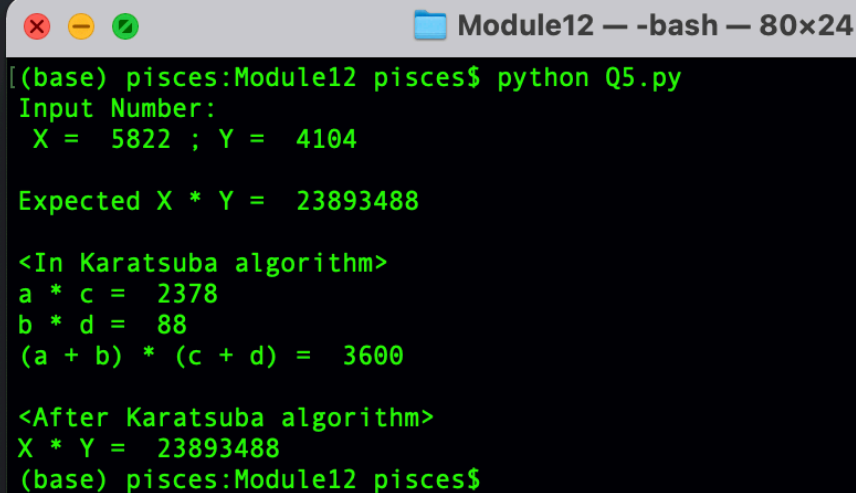


Each row operation in the LU decomposition will be:

Iteration 1:			Iteration 2:			Iteration 3:	
$\begin{bmatrix} 4, & -5, & 6 \\ 2, & 4, & -5 \\ 3, & 8, & -6 \end{bmatrix}$	\rightarrow	$\begin{bmatrix} 4, & -5, & 6 \\ 2, & 4, & -5 \\ 3, & 2, & 4 \end{bmatrix}$	\rightarrow	$\begin{bmatrix} 4, & -5, & 6 \\ 2, & 4, & -5 \\ 3, & 2, & 4 \end{bmatrix}$			

5. My Python code:

```
1 # Hsuan-You Lin Module 10 Problem Set Question 5.
2 def karatsuba(x, y):
3     a, b = x // 100, x % 100
4     c, d = y // 100, y % 100
5
6     ac = a * c
7     print("\n<In Karatsuba algorithm> \na * c = ", ac)
8     bd = b * d
9     print("b * d = ", bd)
10    abcd = (a+b) * (c+d)
11    print("(a + b) * (c + d) = ", abcd)
12    result = ac * 10000 + (abcd - ac - bd) * 100 + bd
13
14    return result
15
16 if __name__ == "__main__" :
17     x = 5822
18     y = 4104
19     print("Input Number: \n X = ", x, "; Y = ", y)
20     print("\nExpected X * Y = ", x*y)
21     ans = karatsuba(x, y)
22     print("\n<After Karatsuba algorithm> \nX * Y = ", ans)
23
```



The terminal window shows the execution of the Python code. The title bar reads "Module12 — -bash — 80x24". The output is as follows:

```
[(base) pisces:Module12 pisces$ python Q5.py
Input Number:
X = 5822 ; Y = 4104

Expected X * Y = 23893488

<In Karatsuba algorithm>
a * c = 2378
b * d = 88
(a + b) * (c + d) = 3600

<After Karatsuba algorithm>
X * Y = 23893488
(base) pisces:Module12 pisces$
```

From above python code, we can see that $(a * c) = 2378$, $(b * d) = 88$, $(a + b) * (c + d) = 3600$, and we can calculate $[(a + b) * (c + d) - (a * c) - (b * d)] = 1134$, Finally we could get $5822 * 4104 = 23893488$ using Karatsuba algorithm.