1. If

\[
L = \begin{pmatrix}
1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
3 & 4 & 1 & 0 \\
4 & 5 & 6 & 1
\end{pmatrix}, \quad b = \begin{pmatrix}
1 \\
2 \\
3 \\
4
\end{pmatrix}
\]

Write a program that finds the solution of

\[Lx = b.\]

\[
n := 4;
L := \text{array}(1..n, 1..n);
b := \text{array}(1..n);
x := \text{array}(1..n);
L := [[1,0,0,0],[2,1,0,0],[3,4,1,0],[4,5,6,1]];
b := [1,2,3,4];
for k from 1 to n do
x[k] := (b[k] - \text{add}(L[k,s]*x[s], s=1..k-1))/L[k,k];
od;
print(x);

OUTPUT: x = [1,0,0,0]

2. If

\[
U = \begin{pmatrix}
1 & 1 & 2 & 1 \\
0 & 2 & 1 & 2 \\
0 & 0 & 2 & 1 \\
0 & 0 & 0 & 1
\end{pmatrix}, \quad b = \begin{pmatrix}
1 \\
2 \\
2 \\
3
\end{pmatrix}
\]

Write a program that finds the solution of

\[Ux = b.\]

\[
n := 4;
U := \text{array}(1..n, 1..n);
b := \text{array}(1..n);
x := \text{array}(1..n);
U := [[1,1,2,1],[0,2,1,2],[0,0,2,1],[0,0,0,1]];
b := [1,2,3,4];
for k from 0 to (n-1) do
x[n-k] := (b[n-k] - \text{add}(U[n-k,s]*x[s], s=0..k-1))/U[n-k,n-k];
\]
3. Write a program to find the LU factorization of the matrix

\[
A = \begin{pmatrix}
1 & 1 & 1 & 1 \\
2 & 4 & 4 & 4 \\
1 & 5 & 8 & 8 \\
2 & 4 & 10 & 14
\end{pmatrix}
\]

assuming the lower triangular matrix \(L\) has 1's along its diagonal.

```plaintext
n := 4;  # all matrices are nxn=4x4
A := array(1..n,1..n);
L := array(1..n,1..n);
U := array(1..n,1..n);
A := [[1,1,1,1],[2,4,4,4],[1,5,8,8],[2,4,10,14]];

for k from 1 to n do  # calculate kth column of L and kth row of U
    for s from 1 to k-1 do
        L[s,k] := 0;  # so that L is lower triangular
        U[k,s] := 0;  # so that U is upper triangular
    od;
    L[k,k] := 1;     # by convention
    k1 := k-1;
    # calculate the kth element of kth row of U
    U[k,k] := A[k,k] - sum(L[k,j0]*U[j0,k],j0=1..k1);
    for t from k+1 to n do
        # calculate remaining elements in kth column of L
        L[t,k] := (A[t,k] - add(L[t,j1]*U[j1,k],j1=1..k1))/U[k,k];
        # calculate remaining elements in kth row of U
        U[k,t] := A[k,t] - add(L[k,j2]*U[j2,t],j2=1..k1);
    od;
    od;
print(L);
print(U);
```

OUTPUT:

\[
L = \begin{pmatrix}
1 & 0 & 0 & 0 \\
2 & 1 & 0 & 0 \\
1 & 2 & 1 & 0 \\
2 & 1 & 2 & 1
\end{pmatrix}, \quad U = \begin{pmatrix}
1 & 1 & 1 & 1 \\
0 & 2 & 2 & 2 \\
0 & 0 & 3 & 3 \\
0 & 0 & 0 & 4
\end{pmatrix}
\]
4. Write a program to find the LU factorization of the matrix

\[ A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 4 & 4 \\ 1 & 5 & 8 & 8 \\ 2 & 4 & 10 & 14 \end{pmatrix} \]

assuming the upper triangular matrix \( U \) has 1's along its diagonal.

\[
\begin{align*}
\text{n} & := 4; \quad \text{# all matrices are nxn=4x4} \\
A & := \text{array}(1..n,1..n); \\
L & := \text{array}(1..n,1..n); \\
U & := \text{array}(1..n,1..n); \\
A & := [[1,1,1,1],[2,4,4,4],[1,5,8,8],[2,4,10,14]]; \end{align*}
\]

```plaintext
for k from 1 to n do  # calculate kth column of L and kth row of U
   for s from 1 to k-1 do
      L[s,k] := 0;  # so that L is lower triangular
      U[k,s] := 0;  # so that U is upper triangular
   od;
   U[k,k] := 1;  # by convention
   k1 := k-1;
   # calculate the kth element of kth row of L
   L[k,k] := A[k,k] - sum(L[k,j0]*U[j0,k],j0=1..k1);
   for t from k+1 to n do
      # calculate remaining elements in kth column of L
      L[t,k] := (A[t,k] - add(L[t,j1]*U[j1,k],j1=1..k1));
      # calculate remaining elements in kth row of U
      U[k,t] := A[k,t] - add(L[k,j2]*U[j2,t],j2=1..k1)/L[k,k];
   od;
print(L);
print(U);
```

**OUTPUT:**

\[
L = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 2 & 2 & 0 & 0 \\ 1 & 4 & -5 & 0 \\ 2 & 2 & 2 & -76/5 \end{pmatrix}, \quad U = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & 1 & 3 & 3 \\ 0 & 0 & 53/5 \\ 0 & 0 & 0 & 1 \end{pmatrix}
\]

5. Find the \( 1 \times 4 \) matrix \( x \) that solves

\[
\begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 4 & 4 & 4 \\ 1 & 5 & 8 & 8 \\ 2 & 4 & 10 & 14 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix}
\]

```plaintext
n := 4; \quad \text{# all matrices are nxn=4x4} \\
A := \text{array}(1..n,1..n); \\
L := \text{array}(1..n,1..n); \\
```
U := array(1..n,1..n);
A := [[1,1,1,1],[2,4,4,4],[1,5,8,8],[2,4,10,14]];
for k from 1 to n do  # calculate kth column of L and kth row of U
  for s from 1 to k-1 do
    L[s,k] := 0;  # so that L is lower triangular
    U[k,s] := 0;  # so that U is upper triangular
  od:
  L[k,k] := 1;  # by convention
  k1 := k-1;
  # calculate the kth element of kth row of U
  U[k,k] := A[k,k] - add(L[k,j]*U[j,k],j=0..k1);
  for t from k1 to n do  # calculate remaining elements in kth column of L
    L[t,k] := (A[t,k] - add(L[t,j]*U[j,k],j=1..k1))/U[k,k];
    # calculate remaining elements in kth row of U
    U[k,t] := A[k,t] - add(L[k,j]*U[j,t],j=1..k1);
  od:
od:
print('L = ', L);
print('U = ', U);

# now we solve Lz = b for z
b := array(1..n);
z := array(1..n);
b := [1,2,3,4];
for k from 1 to n do
  k1 := k-1;
  z[k] := (b[k] - add(L[k,s]*z[s],s=1..k1))/L[k,k];
od:
print('z = ', z);

# now we solve Ux = z for x
x := array(1..n);
for k from 0 to (n-1) do
  k1 := k-1;
  x[n-k] := (z[n-k] - add(U[n-k,n-s]*x[n-s],s=0..k1))/U[n-k,n-k];
od:
print('x = ', x);

OUTPUT:

L = \begin{pmatrix}
  1 & 0 & 0 & 0 \\
  2 & 1 & 0 & 0 \\
  1 & 2 & 1 & 0 \\
  2 & 1 & 2 & 1 \\
\end{pmatrix}, \quad U = \begin{pmatrix}
  1 & 1 & 1 & 1 \\
  0 & 2 & 2 & 2 \\
  0 & 0 & 3 & 3 \\
  0 & 0 & 0 & 4 \\
\end{pmatrix}

z = [1,0,2,-2]

x = [1,-2/3,7/6,-1/2]