5.9.13

Vaishnavi - EE25BTECH11059

October 3, 2025

Question

A shopkeeper has 3 varieties of pens A, B and C. Meenu purchased 1 pen of each variety for a total of Rs 21. Jeevan purchased 4 pens of A variety, 3 pens of B variety and 2 pens of C variety for Rs 60. While Shikha purchased 6 pens of A variety, 2 pens of B variety and 3 pens of C variety for Rs 70. Using matrix method, find the cost of each variety of pen.

Solution

Variable	**Value**
a	cost of pen A
b	cost of pen B
c	cost of pen C

Table: Variables Used

Solution

Let unit cost matrix X be

$$X = \begin{pmatrix} a \\ b \\ c \end{pmatrix} \tag{1}$$

$$\begin{pmatrix} 1 & 1 & 1 \\ 4 & 3 & 2 \\ 6 & 2 & 3 \end{pmatrix} X = \begin{pmatrix} 21 \\ 60 \\ 70 \end{pmatrix} \tag{2}$$

solution

Solving it using a Augmented Matrix

$$\left(\begin{array}{ccc|c} 1 & 1 & 1 & 21 \\ 4 & 3 & 2 & 60 \\ 6 & 2 & 3 & 70 \end{array}\right) \xrightarrow{R_2 \leftarrow R_2 - 4R_1} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 21 \\ 0 & -1 & -2 & -24 \\ 6 & 2 & 3 & 70 \end{array}\right) \tag{3}$$

$$\xrightarrow{R_3 \leftarrow R_3 - 6R_1} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 21 \\ 0 & -1 & -2 & -24 \\ 0 & -4 & -3 & -56 \end{array}\right) \tag{4}$$

Solution

$$\xrightarrow{R_2 \leftarrow -1 \cdot R_2} \left(\begin{array}{ccc|c} 1 & 1 & 1 & 21 \\ 0 & 1 & 2 & 24 \\ 0 & -4 & -3 & -56 \end{array}\right) \tag{5}$$

$$\xrightarrow{R_1 \leftarrow R_1 - R_2} \left(\begin{array}{ccc|c} 1 & 0 & -1 & -3 \\ 0 & 1 & 2 & 24 \\ 0 & -4 & -3 & -56 \end{array}\right) \tag{6}$$

$$\xrightarrow{R_3 \leftarrow R_3 + 4R_2} \left(\begin{array}{ccc|c} 1 & 0 & -1 & -3 \\ 0 & 1 & 2 & 24 \\ 0 & 0 & 5 & 40 \end{array}\right) \tag{7}$$

$$\xrightarrow{R_3 \leftarrow \frac{1}{5} R_3} \left(\begin{array}{ccc|c} 1 & 0 & -1 & -3 \\ 0 & 1 & 2 & 24 \\ 0 & 0 & 1 & 8 \end{array}\right) \tag{8}$$

Solutions

$$\xrightarrow{R_1 \leftarrow R_1 + R_3} \left(\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 2 & 24 \\ 0 & 0 & 1 & 8 \end{array}\right) \tag{9}$$

$$\xrightarrow{R_2 \leftarrow R_2 - 2R_3} \left(\begin{array}{ccc|c} 1 & 0 & 0 & 5 \\ 0 & 1 & 0 & 8 \\ 0 & 0 & 1 & 8 \end{array}\right) \tag{10}$$

Solution

$$\mathbf{X} = \begin{pmatrix} 5 \\ 8 \\ 8 \end{pmatrix} \tag{11}$$

Therefore,
cost of pen A $=$ Rs 5
cost of pen B $=$ Rs 8
cost of pen C $=$ Rs 8

Python Code

```python
def gaussian_elimination(mat):
    n = len(mat)

    for i in range(n):
        # Make the diagonal element 1
        factor = mat[i][i]
        mat[i] = [val / factor for val in mat[i]]

        # Eliminate below and above
        for j in range(n):
            if j != i:
                row_factor = mat[j][i]
                mat[j] = [mat[j][k] - row_factor * mat
                    [i][k] for k in range(n + 1)]

        print(f After making row {i+1} pivot and
            eliminating others: )
        print_matrix(mat)
```

Python Code

```
# Extract solution
    return [row[-1] for row in mat]


def main():
    # Augmented matrix: [A | B]
    matrix = [
        [1, 1, 1, 21],
        [4, 3, 2, 60],
        [6, 2, 3, 70]
    ]

    print( Initial Augmented Matrix: )
    print_matrix(matrix)

    solution = gaussian_elimination(matrix)
```

Python Code

```
variables = ['x', 'y', 'z']
    print( Solution: )
    for var, val in zip(variables, solution):
        print(f {var} = {val:.2f} )


if __name__ ==  __main__ :
    main()
```

C Code

```c
#include <stdio.h>

#define N 3

void printMatrix(double mat[N][N + 1]) {
    for (int i = 0; i < N; i++) {
        for (int j = 0; j < N + 1; j++) {
            printf( %8.3f  , mat[i][j]);
        }
        printf( \n );
    }
    printf( \n );
}
```

C Code

```c
void gaussianElimination(double mat[N][N + 1],
   double result[N]) {
  for (int i = 0; i < N; i++) {
      // Make the diagonal element 1
      double diag = mat[i][i];
      for (int j = 0; j <= N; j++) {
          mat[i][j] /= diag;
      }

      // Eliminate other rows
      for (int k = 0; k < N; k++) {
          if (k != i) {
              double factor = mat[k][i];
              for (int j = 0; j <= N; j++) {
                  mat[k][j] -= factor * mat[i][j];
              }
          }
      }
  }
```

C Code

```
 // Extract solution
    for (int i = 0; i < N; i++) {
        result[i] = mat[i][N];
    }
}

void solve() {
    double mat[N][N + 1] = {
        {1, 1, 1, 21},
        {4, 3, 2, 60},
        {6, 2, 3, 70}
    };

    double result[N];

    gaussianElimination(mat, result);

    printf( Solution:\n );
    printf( x = %.2f\n , result[0]);
```

Python and C Code

```python
import ctypes

# Load the shared object file
lib = ctypes.CDLL('./libcode.so')

# Call the solve function
lib.solve()
```