

## Navie\_GauusElmi.cpp

```
1  #include<iostream>
2  #include<vector>
3  #include<fstream>
4  #include<iomanip>
5
6
7  using namespace std;
8
9  int main()
10 {
11     cout <<"Enter the values for number of Rows(M) and columns(N):";
12     cout <<"M:\n";
13     int m,n;
14     cin >> m;
15     cout << "N:\n";
16     cin >> n;
17
18     cout << "Enter the Matrix Elemnets:\n";
19     vector<vector<double> >Mat(m,vector<double>(n,0));
20
21     for(int i=0;i<m;i++)
22     {
23         for(int j=0; j<n; j++)
24         {
25             cin >> Mat[i][j];
26         }
27     }
28
29     cout << "Matrix you Entered:\n";
30
31     for(int i=0;i<m;i++)
32     {
33         for(int j=0; j<n; j++)
34         {
35             cout << Mat[i][j]<<" ";
36         }
37         cout << "\n";
38     }
39
40     cout <<"Upper Triangular Matrix:\n";
41
42     for(int i=0;i<m-1;i++)
43     {
44         for(int j=i+1;j<m;j++)
45         {
46             double factor = Mat[j][i]/Mat[i][i];
47             for(int k=0; k<n;k++)
48             {
49                 Mat[j][k] = Mat[j][k] - Mat[i][k]*factor;
50             }
51         }
```

```

52     }
53
54     for(int i=0;i<m;i++)
55     {
56         for(int j=0; j<n; j++)
57         {
58             cout << Mat[i][j]<<" ";
59         }
60         cout << "\n";
61     }
62
63     vector<double>y(m,0);
64     y[m-1] = Mat[m-1][n-1]/Mat[m-1][m-1];
65
66     for(int i=m-2;i>=0;i--)
67     {
68         y[i] = Mat[i][n-1];
69         for(int j=i+1;j<m;j++)
70         {
71             y[i] = y[i] - Mat[i][j]*y[j];
72         }
73         y[i] = y[i]/Mat[i][i];
74     }
75
76     ofstream outfile("Naive_GaussElimination.txt");
77     cout << "solution is _" << endl;
78     outfile << "Solution is _\n" << endl;
79     for(int i = 0; i<m;i++)
80     {
81         cout << "x" << i + 1 << " = " << y[i] << endl;
82         outfile << "x" << i + 1 << " = " << y[i] << endl;
83     }
84
85     outfile.close();
86
87     return 0;
88 }

```