//Gauss Elimination
#include<iostream>
#include<iomanip>
using namespace std;
int main()
{
    int n,i,j,k;
    cout.precision(4);  //set precision
    cout.setf(ios::fixed);
    cout<<"\nEnter the no. of equations\n";
    cin>>n;          //input the no. of equations
    float a[n][n+1],x[n];    //declare an array to store the elements of augmented-matrix
    cout<<"\nEnter the elements of the augmented-matrix row-wise:\n";
    for (i=0;i<n;i++)         //Pivotisation
        for (k=i+1;k<n;k++)
            if (abs(a[i][i])<abs(a[k][i]))
                for (j=0;j<=n;j++)
                    {
                        double temp=a[i][j];
                        a[i][j]=a[k][j];
                        a[k][j]=temp;
                    }
    cout<<"\nThe matrix after Pivotisation is:\n";
    for (i=0;i<n;i++)                     //print the new matrix
    {
        for (j=0;j<=n;j++)
            cout<<a[i][j]<<setw(16);
        cout<<"\n";
    }
    for (i=0;i<n-1;i++)       //loop to perform the gauss elimination
        for (k=i+1;k<n;k++)
            {
                double t=a[k][i]/a[i][i];
                for (j=0;j<=n;j++)
                    a[k][j]=a[k][j]-t*a[i][j];    //make the elements below the pivot
                elements equal to zero or eliminate the variables
            }
    cout<<"\n\nThe matrix after gauss-elimination is as follows:\n";
    for (i=0;i<n;i++)                     //print the new matrix
    {
        for (j=0;j<=n;j++)
            cout<<a[i][j]<<setw(16);
        cout<<"\n";
    }
    for (i=n-1;i>=0;i--)                     //back-substitution
    {
        //x is an array whose values correspond to the values of
        x,y,z.
    }
}
C++ Program for Gauss-Elimination for solving a System of Linear Equations

```cpp
x[i]=a[i][n];                //make the variable to be calculated equal to the
rhs of the last equation
for (j=i+1;j<n;j++)
    if (j!=i)            //then subtract all the lhs values except the
        coefficient of the variable whose value is being
        calculated
        x[i]=x[i]-a[i][j]*x[j];
        x[i]=x[i]/a[i][i];            //now finally divide the rhs by the coefficient
        of the variable to be calculated
}
cout<"The values of the variables are as follows:\n";
for (i=0;i<n;i++)
    cout<<x[i]<<endl;            // Print the values of x, y,z,. . .
return 0;
```

**Sample 1**

The matrix after Pivotisation is:

```
6.0000 1.0000 -6.0000 -5.0000 6.0000
0.0000 2.0000 0.0000 1.0000 0.0000
2.0000 2.0000 3.0000 2.0000 -2.0000
4.0000 -3.0000 0.0000 1.0000 -7.0000
```

The matrix after gauss-elimination is as follows:

```
6.0000 1.0000 -6.0000 -5.0000 6.0000
0.0000 2.0000 0.0000 1.0000 0.0000
0.0000 0.0000 5.0000 2.8333 -4.0000
0.0000 0.0000 -0.0000 3.9000 -7.8060
```

The values of the variables are as follows:

-0.5000
1.0000
0.3333
-2.0000
The matrix after Pivotisation is:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0000</td>
<td>-2.0000</td>
<td>1.0000</td>
<td>15.0000</td>
</tr>
<tr>
<td>-3.0000</td>
<td>-1.0000</td>
<td>4.0000</td>
<td>8.0000</td>
</tr>
<tr>
<td>1.0000</td>
<td>-1.0000</td>
<td>3.0000</td>
<td>13.0000</td>
</tr>
</tbody>
</table>

The matrix after gauss-elimination is as follows:

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4.0000</td>
<td>-2.0000</td>
<td>1.0000</td>
<td>15.0000</td>
</tr>
<tr>
<td>0.0000</td>
<td>-2.5000</td>
<td>4.7500</td>
<td>19.2500</td>
</tr>
<tr>
<td>0.0000</td>
<td>0.0000</td>
<td>1.8000</td>
<td>5.4000</td>
</tr>
</tbody>
</table>

The values of the variables are as follows:

<p>| |</p>
<table>
<thead>
<tr>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>2.0000</td>
</tr>
<tr>
<td>-2.0000</td>
</tr>
<tr>
<td>3.0000</td>
</tr>
</tbody>
</table>

Sample 2

Tutorial Video:

Manas Sharma

PhD researcher at Friedrich-Schiller University Jena, Germany. I'm a physicist specializing in theoretical, computational and experimental condensed matter physics. I like to develop Physics related apps and softwares from time to time. Can code in most of the popular languages. Like to share my knowledge in Physics and applications using this Blog and a YouTube channel.