1. Write a program to solve system of n linear equations using Gauss Elimination method.
Algorithm:

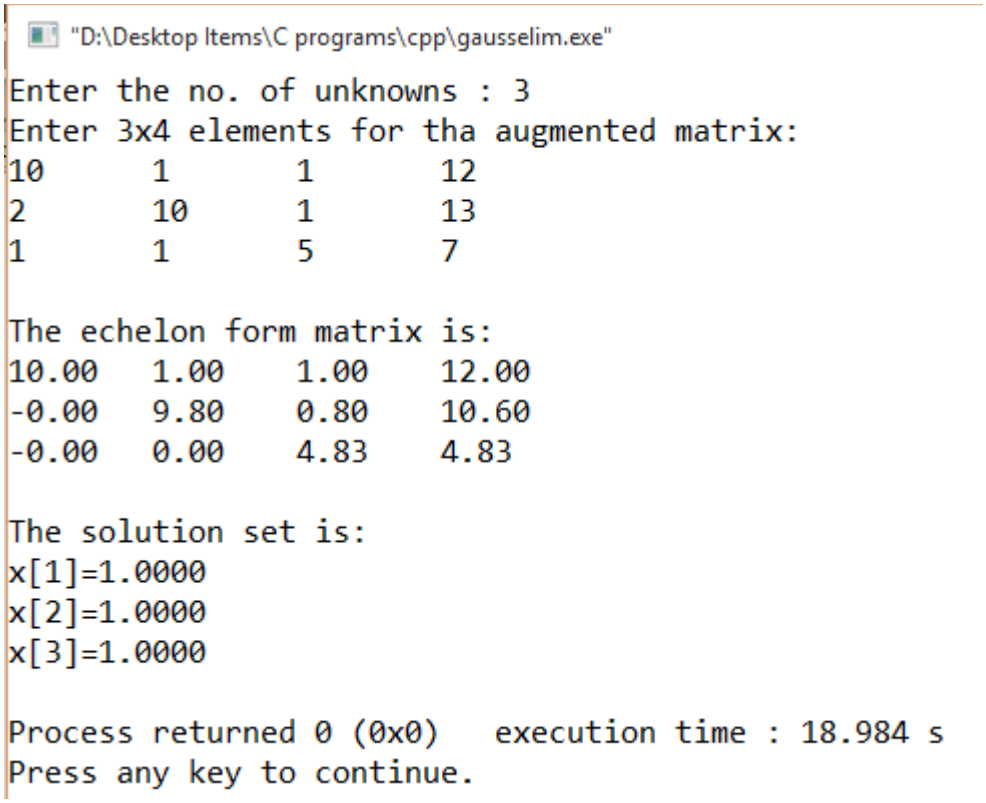
1. Enter the number of unknowns, n.
2. Enter system of n linear equations in augmented matrix $A_{n \times (n+1)}$
3. Perform forward elimination as,
   For k = 1 to n-1
   For i= k+1 to n
   For j=k to n+1
   $$A_{ij} \leftarrow A_{ij} - A_{kj} * \frac{A_{ik}}{A_{kk}}$$
4. Perform backward substitution as,
   $x_n \leftarrow A_{n,n+1}/A_{n,n}$
   $x_i \leftarrow (A_{i,n+1} - \sum_{j=i+1}^{n} x_j A_{i,j})/A_{i,i}$
5. Display the solution x_i where i = 1 to n
6. Stop

Source Code:

```
/*Gauss Elimination Method */
#include<stdio.h>
#include<math.h>
#define MAX 10

int main()
{
   int i,j,k,n;
   float a[MAX][MAX],x[MAX],c,sumx;
   printf("Enter the no. of unknowns : ");
   scanf("%d",&n);
   printf("Enter %dx%d elements for tha augmented matrix:\n",n,n+1);
   for(i=1;i<=n;i++)
      for(j=1;j<=n+1;j++)
         scanf("%f",&a[i][j]);

   /*Elimination process*/
   for(k=1;k<=n-1;k++)
   {
      for(i=k+1;i<=n;i++)
      {
         c=a[i][k]/a[k][k];
         for(j=k;j<=n+1;j++)
            a[i][j]=a[i][j]-c*a[k][j];
      }
   }

   for(i=1;i<=n;i++)
   {
      for(j=1;j<=n+1;j++)
         printf("%.2f\t",a[i][j]);
      printf("\n");
   }
```

```
   /*Backward Substituion*/

   x[n]=a[n][n+1]/a[n][n];
   for(i=n-1;i>=1;i--)
   {   sumx=0;
      for(j=i+1;j<=n;j++)
         sumx+=x[j]*a[i][j];
      x[i]=(a[i][n+1]-sumx)/a[i][i];
   }
   for(i=1;i<=n;i++)
      printf("x[%d]=%.4f\t",i,x[i]);
return 0;
}
```

Output:



2. Write a program to solve the system of n linear equations using Gauss Jordan Method.
Algorithm:

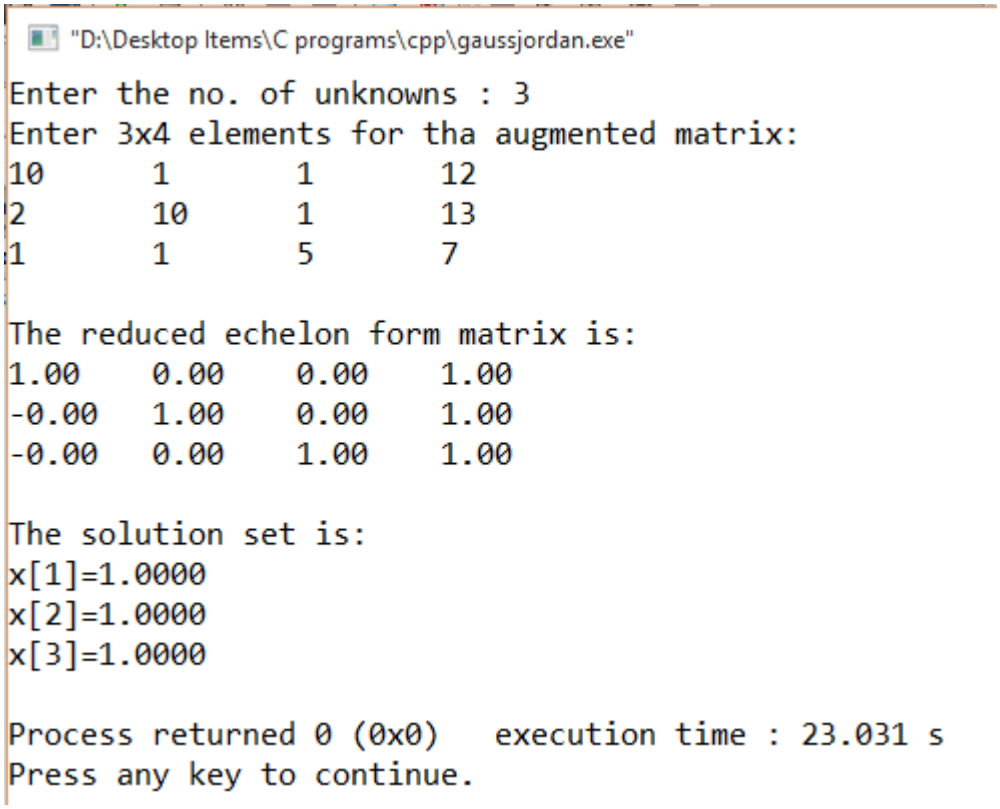
1. Enter the number of unknowns, n.
2. Enter system of n linear equations in augmented matrix $A_{n \times (n+1)}$
3. Perform forward and backward elimination as,
   For k = 1 to n
   For i= 1 to n when i ≠ k do
   For j=k to n+1
   $$A_{ij} \leftarrow A_{ij} - A_{kj} * \frac{A_{ik}}{A_{kk}}$$
4. Make the diagonal element unity as,
   For i = 1 to n do
   $A_{n+1,i} = A_{n+1,i} / A_{ii}$
   $A_{ii} = A_{ii} / A_{ii}$
   $x_i = A_{n+1,i}$
5. Display the solution x_i where i = 1 to n
6. Stop

Source Code:

```
/*Gauss Jordan Method */
#include<stdio.h>
#include<math.h>
#define MAX 10

int main()
{
   int i,j,k,n;
   float a[MAX][MAX],c;
   printf("Enter the no. of unknowns : ");
   scanf("%d",&n);
   printf("Enter %dx%d elements for tha augmented matrix:\n",n,n+1);
   for(i=1;i<=n;i++)
      for(j=1;j<=n+1;j++)
         scanf("%f",&a[i][j]);

   /*forward and backward Elimination process*/

   for(k=1;k<=n;k++)
   {
      for(i=1;i<=n;i++)
      {   if(i!=k){
         c=a[i][k]/a[k][k];
         for(j=k;j<=n+1;j++)
            a[i][j]=a[i][j]-c*a[k][j];}
      }
   }

   printf("\nThe reduced echelon form matrix is:\n");
   for(i=1;i<=n;i++)
   {   a[i][n+1]/=a[i][i];
      a[i][i]/=a[i][i];/*making diagonal element identity*/
```

```
        for(j=1;j<=n+1;j++)
            printf("%.2f\t",a[i][j]);
        printf("\n");
    }
    printf("\nThe solution set is:\n");
    for(i=1;i<=n;i++)
        printf("x[%d]=%.4f\n",i,a[i][n+1]);
    return 0;
}
```

Output:



3. Write a program to solve the system of n linear equations using Gauss Seidel Method.
Algorithm:

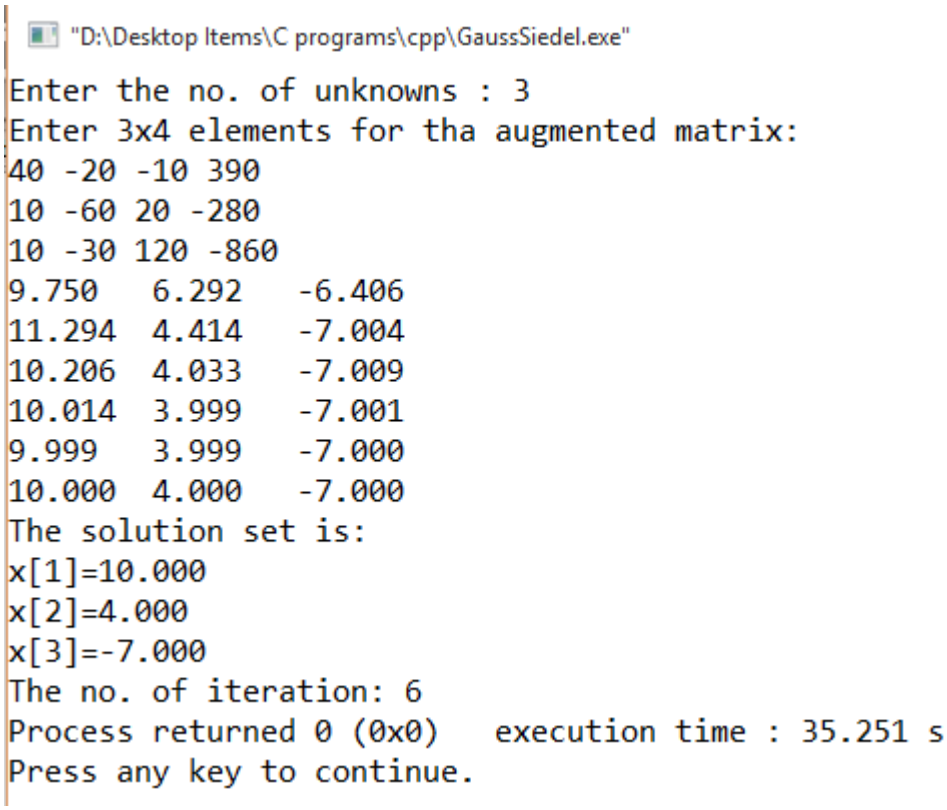
1. Enter the number of unknowns, n.
2. Enter system of n linear equations in augmented matrix $\mathbf{A_{n\ x\ (n+1)}}$
3. Perform iterative step as,
   For k = 1 to maxIteration
      For i= 1 to n do
         $\mathbf{x_j} = \frac{1}{A_{ii}} [-\sum_{j=1}^{i-1} A_{ij}x_j - \sum_{j=1+1}^{n} A_{ij}x0_j + A_{i,n+1}]$
      If $\|x - x0\| < maxError$ **then** $OUTPUT\ (x1, x2, x3, \dots, xn)$ **and STOP**
      For i = 1 to n
         Set $\mathbf{x0_i = x_i}$
4. OUTPUT "Maximum number of iteration exceeded."

Source Code:

```
/*Gauss Seidel Method*/
#include<stdio.h>
#include<math.h>
#define MAX 10
#define E 0.0001
#define N 100

int main()
{
   int i,j,k,n;
   float a[MAX][MAX],x[MAX],x0[MAX],sum1,sum2;
   printf("Enter the no. of unknowns : ");
   scanf("%d",&n);
   printf("Enter %dx%d elements for tha augmented matrix:\n",n,n+1);
   for(i=1;i<=n;i++)
      for(j=1;j<=n+1;j++)
         scanf("%f",&a[i][j]);
   for(i=1;i<=n;i++)
      x0[i]=0;
   for(k=1;k<=N;k++)
   {
      for(i=1;i<=n;i++)
      {   sum1=sum2=0;
         for(j=1;j<=i-1;j++)
            sum1+=a[i][j]*x[j];
         for(j=i+1;j<=n;j++)
            sum2+=a[i][j]*x0[j];
         x[i]=(-sum1-sum2+a[i][n+1])/a[i][i];
         printf("%.3f\t",x[i]);
      }
      printf("\n");
      if((fabs(x[1]-x0[1])/x[1])<E)
         break;

      for(i=1;i<=n;i++)
      {
```

```
            x0[i]=x[i];
        }
    }
    if(k==N+1)
        printf("Maximum number of iterations exceeded.");
    else{
    printf("The solution set is:\n");
    for(i=1;i<=n;i++)
        printf("x[%d]=%.3f\n",i,x[i]);
    }
    printf("The no. of iteration: %d",k);
    return 0;
}
```

Output:



4. Write a program to solve the system of n linear equations using Jacobi Method.
Algorithm:

1. Enter the number of unknowns, n.
2. Enter system of n linear equations in augmented matrix $\mathbf{A_{n\,,\,(n+1)}}$
3. Perform iterative step as,
   For k = 1 to maxIteration
      For i= 1 to n do
         $\mathbf{x_j} = \frac{1}{A_{ii}} [-\sum_{j=1, i\neq j}^{n} A_{ij} x0_j + A_{i,n+1}]$
      If $\|x - x0\| < maxError$ **then** $OUTPUT\ (x1, x2, x3, \ldots, xn)$ **and STOP**
      For i = 1 to n
         Set $x0_i = x_i$
4. OUTPUT "Maximum number of iteration exceeded."

Source Code:

```
/*Jacobi Method*/
#include<stdio.h>
#include<math.h>
#define MAX 10
#define E 0.0001
#define N 100

int main()
{
   int i,j,k,n;
   float a[MAX][MAX],x[MAX],x0[MAX],sum;
   printf("Enter the no. of unknowns : ");
   scanf("%d",&n);
   printf("Enter %dx%d elements for tha augmented matrix:\n",n,n+1);
   for(i=1;i<=n;i++)
      for(j=1;j<=n+1;j++)
         scanf("%f",&a[i][j]);
   for(i=1;i<=n;i++)
      x0[i]=0;
   for(k=1;k<=N;k++)
   {
      for(i=1;i<=n;i++)
      {   sum=0;
         for(j=1;j<=n;j++)
            if(i!=j) sum+=a[i][j]*x0[j];
         x[i]=(-sum+a[i][n+1])/a[i][i];
         printf("%.3f\t",x[i]);
      }
      printf("\n");
      if((fabs(x[1]-x0[1])/x[1])<E)
         break;

      for(i=1;i<=n;i++)
      {
         x0[i]=x[i];
      }
```

```
    }
    if(k==N+1)
       printf("Maximum number of iterations exceeded.");
    else{
    printf("The solution set is:\n");
    for(i=1;i<=n;i++)
       printf("x[%d]=%.3f\n",i,x[i]);
    }
    printf("\nThe no. of iteration: %d",k);
    return 0;
}
```

Output:

```
"D:\Desktop Items\C programs\cpp\jacobi.exe"
Enter the no. of unknowns : 3
Enter 3x4 elements for tha augmented matrix:
40 -20 -10 390
10 -60 20 -280
10 -30 120 -860
9.750   4.667   -7.167
10.292  3.903   -6.813
9.998   4.111   -7.049
10.043  3.984   -6.972
9.999   4.017   -7.008
10.006  3.997   -6.996
10.000  4.002   -7.001
10.001  4.000   -6.999
10.000  4.000   -7.000
The solution set is:
x[1]=10.000
x[2]=4.000
x[3]=-7.000

The no. of iteration: 9
Process returned 0 (0x0)   execution time : 29.297 s
Press any key to continue.
```