```
/*****************************************************************************
                    Solving linear system using
                Gaussian Elimination with Partial Pivoting
*****************************************************************************/
```

1

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10

int gauss2(int n, float a[MAX][ MAX], float b[MAX], float x[MAX]){
   clim(n, a, b);
   bsub(n, a, b, x);
   return 0;
}

int clim(int n, float a[MAX][ MAX], float b[MAX]){
   int i, j, k;
   float factor;
   for(k=1;k<=n-1;k++){
      pivot(n, a, b, k);
      for(i=k+1;i<=n;i++){
         factor=a[i][k]/a[k][k];
         for(j=k+1;j<=n;j++){
            a[i][j]=a[i][j]-factor*a[k][j];
         }
         b[i]=b[i]-factor*b[k];
      }
   }
   return 0;
}

int pivot(int n, float a[MAX][ MAX], float b[MAX], int k){
   int p, i, j;
   float large, temp;
   p=k;
   large=fabs(a[k][k]);
   for(i=k+1;i<=n;i++){
      if(fabs(a[i][k])>large){
         large=fabs(a[i][k]);
         p=i;
      }
   }
   if(p!=k){
      for(j=k;j<=n;j++){
         temp=a[p][j];
         a[p][j]=a[k][j];
         a[k][j]=temp;
      }
      temp=b[p];
      b[p]=b[k];
      b[k]=temp;
   }
   return 0;
}
```

```
int bsub(int n, float a[MAX][ MAX], float b[MAX], float x[MAX]){
   int i, j, k;
   float sum;
   x[n]=b[n]/a[n][n];
   for(k=n-1;k>=1;k--){
      sum=0.0;
      for(j=k+1;j<=n;j++)
         sum=sum+a[k][j]*x[j];
      x[k]=(b[k]-sum)/a[k][k];
   }
   return 0;
}

int main()
{
   int i, j, n;
   float a[MAX][ MAX], b[MAX], x[MAX];
   printf("\nInput the number of variables: ");
   scanf("%d", &n);
   printf("\nInput coefficients a(i,j), row-wise (one row on each line): ");
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
         scanf("%f", &a[i][j]);
   printf("\nEnter vector b: ");
   for(i=1;i<=n;i++)
      scanf("%f", &b[i]);
   gauss2(n, a, b, x);
   printf("\nSolution vector x:\n");
   for(i=1;i<=n;i++)
      printf ("\t%f", x[i]);
   getch();
   return 0;
}
```

```
/*************************************************************************
                    Solving linear system using
                Gauss-Jordan Method with Partial Pivoting
*************************************************************************/
```

3.

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10

int main()
{
   int i, j, k, n, pivrow;
   float a[MAX][MAX], b[MAX], large, temp, factor, pivot;
   printf("\nInput the number of variables: ");
   scanf("%d", &n);
   printf("\nInput coefficients a(i,j) row-wise (one row on each line): ");
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
         scanf("%f", &a[i][j]);
   printf("\nEnter vector b: ");
   for(i=1;i<=n;i++)
      scanf("%f", &b[i]);
   for(i=1;i<=n;i++){
      pivrow=i;
      large=a[i][i];
      for(k=i+1;k<=n;k++){
         if (fabs(a[k][i])>large) {
            large=a[k][i];
            pivrow=k;
         }
      }
      if(pivrow!=i){
         for(j=i;j<=n;j++){
            temp=a[pivrow][j];
            a[pivrow][j]=a[i][j];
            a[i][j]=temp;
         }
         temp=b[pivrow];
         b[pivrow]=b[i];
         b[i]=temp;
      }
      pivot=a[i][i];
      for(j=1;j<=n;j++)
         a[i][j]=a[i][j]/pivot;
      b[i]=b[i]/pivot;
      for(j=1;j<=n;j++){
         if (j!=i) {
            factor=a[j][i];
            for(k=i;k<=n;k++)
               a[j][k]=a[j][k]-factor*a[i][k];
            b[j]=b[j]-factor*b[i];
         }
      }
   }
```

```
    printf("\nSolution vector x:\n");
    for(i=1;i<=n;i++)
       printf ("\t%f", b[i]);
    getch();
    return 0;
}
```

```
/*****************************************************************************
                    Solving linear system using Jacobi Iterative Method
*****************************************************************************/
```

5.

```
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10
#define EPS 0.0001

int main()
{
   int i, j, n, key, count=0;
   float a[MAX][MAX], b[MAX], x[MAX], xold[MAX], sum;
   printf("\nInput the number of variables: ");
   scanf("%d", &n);
   printf("\nInput coefficients a(i,j) row-wise (one row on each line): ");
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
         scanf("%f", &a[i][j]);
   printf("\nEnter vector b: ");
   for(i=1;i<=n;i++)
      scanf("%f", &b[i]);
   for(i=1;i<=n;i++)
      xold[i]=0;
   do {
      key=0;
      for(i=1;i<=n;i++){
         sum=b[i];
         for(j=1;j<=n;j++){
            if (j!=i) sum=sum-(a[i][j]*xold[j]);
         }
         x[i]=sum/a[i][i];
         printf("%f\t", x[i]);
         if ((key==0)&&(fabs((x[i]-xold[i])/x[i])>EPS))
               key=1;
      }
      printf("\n");
      for(i=1;i<=n;i++)
         xold[i]=x[i];
      count++;
   }
   while (key==1);
   printf("\nSolution vector x:\n");
   for(i=1;i<=n;i++)
      printf ("\t%f", x[i]);
   printf("\nNumber of iterations: %d", count);
   getch();
   return 0;
}
```

```
/*****************************************************************************
                Solving linear system using Gauss-Seidel Iterative Method
*****************************************************************************/
                                    6
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10
#define EPS 0.0001

int main()
{
   int i, j, n, key, count=0;
   float a[MAX][MAX], b[MAX], x[MAX], sum, dummy;
   printf("\nInput the number of variables: ");
   scanf("%d", &n);
   printf("\nInput coefficients a(i,j) row-wise (one row on each line): ");
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
         scanf("%f", &a[i][j]);
   printf("\nEnter vector b: ");
   for(i=1;i<=n;i++)
      scanf("%f", &b[i]);
   for(i=1;i<=n;i++)
      x[i]=0;
   do {
      key=0;
      for(i=1;i<=n;i++){
         sum=b[i];
         for(j=1;j<=n;j++){
            if (j!=i) sum=sum-(a[i][j]*x[j]);
         }
         dummy=sum/a[i][i];
         printf("%f\t", dummy);
         if ((key==0)&&(fabs((dummy-x[i])/dummy)>EPS))
               key=1;
         x[i]=dummy;
      }
      printf("\n");
      count++;
   }
   while (key==1);
   printf("\nSolution vector x:\n");
   for(i=1;i<=n;i++)
      printf ("\t%f", x[i]);
   printf("\nNumber of iterations: %d", count);
   getch();
   return 0;
}
```

```
/*************************************************************************
                Finding eigenvalue and eigenvector using Power Method
*************************************************************************/
                                    f.
#include <stdio.h>
#include <math.h>
#include <conio.h>
#define MAX 10
#define EPS 0.0001

int main()
{
   int i, j, k, n, count=0;
   float ev=1.0, ev_temp, temp_max, a[MAX][MAX], x[MAX], y[MAX];
   printf("\nInput the size of matrix: ");
   scanf("%d", &n);
   printf("\nInput elements a(i,j) row-wise (one row on each line): ");
   for(i=1;i<=n;i++)
      for(j=1;j<=n;j++)
         scanf("%f", &a[i][j]);
      for(i=1;i<=n;i++)
         x[i]=1;
   do {
      for(i=1;i<=n;i++)
         y[i]=0;
      ev_temp=ev;
      for(i=1;i<=n;i++)
         for(j=1;j<=n;j++)
            y[i]=y[i]+a[i][j]*x[j];
      for(i=1;i<=n;i++)
         x[i]=y[i];
      temp_max=fabs(x[1]);
      k=1;
      for(j=2;j<=n;j++){
         if (fabs(x[j])>temp_max){
            temp_max=fabs(x[j]);
            k=j;
         }
      }
      ev=x[k];
      for(i=1;i<=n;i++)
         x[i]=x[i]/ev;
      count++;
      printf("\nEigenvalue: %f\t", ev);
      printf("Eigenvector: ");
      for(i=1;i<=n;i++)
         printf ("%f\t", x[i]);
   } while(fabs((ev_temp-ev)/ev)>EPS);
   printf("\nApproximate eigenvalue is %f: ", ev);
   printf("\nCorresponding eigenvector is:\n");
   for(i=1;i<=n;i++)
      printf ("\t%f", x[i]);
   printf("\nNumber of iterations: %d", count);
   getch();
   return 0;
}
```