```
#include<stdio.h>
#include<conio.h>

void main()
{
    int i,j,k,m,n;
    static float coeff[20][20],solu_vect[10],rhs_vect[20];
    float sum=0.0;
    clrscr();
    printf("\n*************This program implements Gauss Seidal Method**********");
    l:
    printf("\n\t\tEnter number of equations :: ");
    scanf("%d",&m);
    printf("\n\t\tEnter number of variables :: ");
    scanf("%d",&n);
    if(m != n)
    {
        printf("\nGauss elemination method works when number of equation and
variables are same.\n");
        goto l;
    }

    /*input coefficients*/
    printf("\nEnter the elements of coefficient matrix:\n");
    for(i=0; i<n; i++)
    {
        printf("\n\tEnter coefficients for equation %d --> ",i+1);
        for(j=0; j<n; j++)
        {
            printf("\n\t\tEnter coefficient for %dth variable --> ",j+1);
            scanf("%f",&coeff[i][j]);
        }
    }

    /*input right hand side vector*/
    printf("\nEnter the elements of right hand side vector:\n");
    for(i=0; i<n; i++)
    {
        printf("\n\tEnter rhs value for equation %d --> ",i+1);
        scanf("%f",&rhs_vect[i]);
    }

    /*create augmented matrix*/
    for(i=0; i<n; i++)
    {
        for(j=0; j<n; j++)  aug[i][j] = coeff[i][j];
        aug[i][n] = rhs_vect[i];
    }

    /*Making the system of equation diagonally dominant*/
```

```
    for(i=0; i<n; i++)
    {
        sum = 0;
        for(j=0; j<n; j++)
                if(i!=j) sum = sum + aug[i][j];
        max = aug[i][0];
        for(j=0; j<n; j++)
                if(max < aug[i][j])
                {
                        max = aug[i][j];
                        loc = j;
                }
        if(max >= sum)
                if(loc != i)
                        swap(aug[i],aug[loc])
    }


    /*Forward elemination process for the generation of upper triangular matrix*/
    for(i=0; i<n-1; i++)
    {
        for(j=i+1; j<n; j++)
        {
                pivot = aug[j][i]/aug[i][i];
                for(k=i; k<=n; k++)    aug[j][k] = aug[j][k] - pivot*aug[i][k];
        }
    }

    /*Back substitution process to generate solution vector*/
    solu_vect[n-1] = aug[n-1][n]/aug[n-1][n-1];
                /*loop for rest of the solutions*/
    for(i=n-2; i>=0; i--)
    {
        sum=0;
        for(j=i+1; j<n; j++)  sum = sum + aug[i][j]*solu_vect[j];
        solu_vect[i] = (aug[i][n] - sum)/aug[i][i];
    }

    /*printing solutions*/
    printf("\nThe solutions are as follows: \n\n");
    for(i=0; i<n; i++)   printf("\n\tx%d=%f\t",i+1,solu_vect[i]);
    getch();
    return;
}
```