PROBLEM: Write a c program to find the solⁿ of the given system using Gauss-Seidal iterative method.

$$8x_1 + 2x_2 - 2x_3 = 8$$
$$x_1 - 8x_2 + 8x_3 = -9$$
$$2x_1 + x_2 + 9x_3 = 12$$

FORMULA AND THEORY DISCUSSION:

To illustrate the method we can write the system of eqⁿ : $x_1 = \frac{1}{a_{11}}(b_1 - a_{12}x_2 - a_{13}x_3 - \cdots a_{1n}x_n)$

$\cdots$

$$x_n = \frac{1}{a_{nn}}(b_n - a_{n1}x_1 - a_{n2}x_2 - \cdots a_{nn-1}x_{n-1})$$

provided $a_{ii} \neq 0$, $i = 1, 2, \ldots n$

To solve the eqⁿs $x_1^{(0)}, x_2^{(0)}, \ldots x_n^{(0)}$ be the initial approximation.

$\therefore x_1^{(1)} = (b_1 - a_{12}x_2^{(0)} - a_{13}x_3^{(0)} - \cdots - a_{1n}x_n^{(0)})/a_{11}$

$\therefore x_2^{(1)} = (b_2 - a_{21}x_1^{(1)} - a_{23}x_3^{(0)} - \cdots - a_{2n}x_n^{(0)})/a_{22}$

$\cdots$

$x_n^{(1)} = (b_n - a_{n1}x_1^{(1)} - a_{n2}x_2^{(1)} - \cdots - a_{nn-1}x_{n-1}^{(1)})/a_{nn}$

$\therefore x_n^{(k+1)} = (b_n - a_{n1}x_1^{(k+1)} - a_{n2}x_2^{(k+1)} - \cdots - a_{nn-1}x_{n-1}^{(k+1)})/a_{nn}$

☒ ALGORITHM:

Step-1: START

Step-2: Read the augmented matrix (a_{ij}), $i=1$ to n; $j=1$ to $(n+1)$

Step-3: Enter the initial approximation $x_i = 0$, $i=1$ to n

Step-4: for $i = 1(1)n$

Step-5: Set $S = a_{i,n+1}$

Step-6: for $j = 1(1)n$

Step-7: if $i \neq j$, set $s = s - a_{ij}x_j$

Step-8: Else, next j

Step-9: $x_i = s/a_{ij}$

Step-10: Print the value of x_i $(i=1,2,\ldots n)$

Step-11: Stop the program.

☒ SOURCE CODE:

```
#include<stdio.h>
#include<conio.h>
#include<math.h>
int main()
{
int n, i, j, k;
float a[100][100], x[100], s;
```

```
printf("\n Enter the order of the co-efficient matrix:");
scanf("%d", &n);
printf("\nEnter the elements of the augmented matrix:\n");
for(i=1; i<=n; i++)
{
for(j=1; j<=n+1; j++)
{
scanf("%f", &a[i][j]);
}
}
printf("\n Enter the initial approximation:\n");
for(i=1; i<=n; i++)
{
scanf("%f", &x[i]);
}
for(k=1; k<=15; k++)
{
for(i=1; i<=n; i++)
{
s=a[i][n+1];
for (j=1; j<=n; j++)
{
if(j!=i)
{
s=s-a[i][j]*x[i];
```

```
}
}
x[i]=s/a[i][j];
}
}
printf("\n The required solution is :\n");
for(i=1; i<=n; i++)
printf("x[%d] = %f\n", i, x[i]);
return 0;
}
```

OUTPUT:

```
Enter the order of the coefficient matrix: 3
Enter the elements of the augmented matrix:
8    2    -2    8
1    -8    8    -9
2    1    9    12
Enter the initial approximation:
0    0    0
The required solution is:
x[1] = 0.696970
x[2] = 2.151515
x[3] = 0.939394
```