

```

1 // gauss_jordan.cpp
2 #include <stdio.h>
3
4 const int dim = 3; // 連立方程式の未知数の数
5 typedef double Matrix[dim][dim+1]; // 係数行列の型を定義
6
7 // 行列を表示させる関数
8 void print_matrix(Matrix A) {
9     for (int i=0; i<dim; i++) {
10         for (int j=0; j<dim+1; j++) {
11             if (j == dim) printf(" |%.3f", A[i][j]);
12             else printf("%.3f", A[i][j]);
13         }
14         printf("\n");
15     }
16     printf("\n");
17 }
18
19 // ガウス・ジョルダン法
20 void gauss_jordan(Matrix A) {
21     for (int i=0; i<dim; i++) {
22         double pivot = A[i][i]; // ピボットを定義
23         for (int j=i; j<dim+1; j++) { // 各列をピボットで除算
24             A[i][j] /= pivot;
25         }
26         for (int k=0; k<dim; k++) { // 掃き出し計算
27             if (k != i) {
28                 double tmp = A[k][i];
29                 for (int j=i; j<dim+1; j++) {
30                     A[k][j] -= tmp * A[i][j];
31                 }
32             }
33         }
34         print_matrix(A);
35     }
36 }
37
38 // 解を表示させる関数
39 void print_solutions(Matrix A) {
40     printf("----- Solutions of Equations ----- \n");
41     for (int i=0; i<dim; i++) {
42         printf("x_%d = %.3f\n", i+1, A[i][dim]);
43     }
44 }
45
46 // メイン関数
47 int main(void) {
48     Matrix A = {
49         { 2, 3, 1, 4} ,
50         {4, 1, -3, -2},
51         {-1, 2, 2, 2}
52     };
53
54     print_matrix(A); // 行列を表示
55     gauss_jordan(A); // ガウス・ジョルダン法で連立方程式の解を計算
56     print_solutions(A); // 連立方程式の解を表示
57
58     return 0;
59 }
60

```