

Numerical Methods HW #03

1. 다음의 연립방정식을 Gauss 소거법을 사용하여 다음 물음과 같은 방법으로 풀어라.

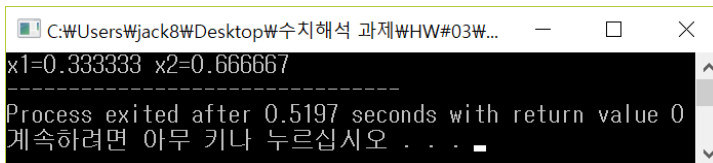
$$\begin{aligned} 0.00005x_1 + 2x_2 &= 1.33335 \\ 3x_1 + 18x_2 &= 13 \end{aligned}$$

(a) 부분피봇팅을 실시하지 않고 풀어라.

<Code>

```
#include<stdio.h>
int main()
{
    double A[2][3]={0.00005,2,1.33335},{3,18,13}};
    double B[2][3];
    int i;
    for(i=0;i<=2;i++)
    {
        B[0][i]=A[0][i]*A[1][0]/A[0][0];
    }
    for(i=0;i<=2;i++)
    {
        B[1][i]=B[0][i]-A[1][i];
    }
    double x1,x2;
    x2=B[1][2]/B[1][1];
    x1=(B[0][2]-x2*B[0][1])/B[0][0];
    printf("x1=%1f x2=%1f",x1,x2);
}
```

<Result>



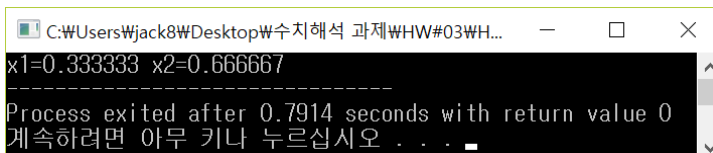
```
C:\Users\Wjack8\Desktop\수치해석 과제\WHW#03W...
x1=0.333333 x2=0.666667
-----
Process exited after 0.5197 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

(b) 부분피봇팅을 실시하여 풀어라.

<Code>

```
#include<stdio.h>
int main()
{
    double A[2][3]={3,18,13},{0.00005,2,1.33335}};
    int i;
    double B[2][3];
    for(i=0;i<=2;i++)
    {
        B[0][i]=A[0][i]*A[1][0]/A[0][0];
    }
    for(i=0;i<=2;i++)
    {
        B[1][i]=B[0][i]-A[1][i];
    }
    double x1,x2;
    x2=B[1][2]/B[1][1];
    x1=(B[0][2]-x2*B[0][1])/B[0][0];
    printf("x1=%1f x2=%1f",x1,x2);
}
```

<Result>



```
C:\Users\Wjack8\Desktop\수치해석 과제\WHW#03WH...
x1=0.333333 x2=0.666667
-----
Process exited after 0.7914 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

2. Crout법을 이용하여 다음 행렬의 역행렬을 구하라.

$$A = \begin{bmatrix} 3 & -0.1 & -0.2 \\ 0.1 & 7 & -0.3 \\ 0.3 & -0.2 & 10 \end{bmatrix}$$

<Code>

```
#include<stdio.h>
int main()
{
    double A[3][3]={3,-0.1,-0.2},{0.1,7,-0.3},{0.3,-0.2,10}};
    double L[3][3]={0}, U[3][3]={0};
    int i,j;
    for(i=0;i<=2;i++)
    {
        L[i][0]=A[i][0];
        U[0][i]=A[0][i]/L[0][0];
    }
    L[1][1]=A[1][1]-L[1][0]*U[0][1];
    L[2][1]=A[2][1]-L[2][0]*U[0][1];
    L[2][2]=A[2][2]-L[2][0]*U[0][2]-L[2][1]*U[1][2];
    U[1][1]=1;
    U[2][2]=1;
    U[1][2]=(A[1][2]-L[1][0]*U[0][2])/L[1][1];
    printf("Matrix L\n");
    for(j=0;j<=2;j++)
    {
        for(i=0;i<=2;i++)
        {
            printf("%.21f",L[j][i]);
        }
        printf("\n");
    }
    printf("Matrix U\n");
    for(j=0;j<=2;j++)
    {
        for(i=0;i<=2;i++)
        {
            printf("%.21f",U[j][i]);
        }
        printf("\n");
    }
    double invL[3][3]={1,0,0},{0,1,0},{0,0,1}};
    double invU[3][3]={1,0,0},{0,1,0},{0,0,1}};
    for(i=0;i<=2;i++)
    {
        invL[0][i]=invL[0][i]/L[0][0];
    }
    for(i=0;i<=2;i++)
    {
        invL[1][i]=invL[1][i]-invL[0][i]*L[1][0];
    }
    for(i=0;i<=2;i++)
    {
        invL[1][i]=invL[1][i]/L[1][1];
    }
    for(i=0;i<=2;i++)
    {
        invL[2][i]=invL[2][i]-invL[0][i]*L[2][0]-invL[1][i]*L[2][1];
    }
    for(i=0;i<=2;i++)
    {
        invL[2][i]=invL[2][i]/L[2][2];
    }
    for(i=0;i<=2;i++)
    {
        invU[1][i]=invU[1][i]-invU[2][i]*U[1][2];
    }
    for(i=0;i<=2;i++)
    {
        invU[0][i]=invU[0][i]-invU[1][i]*U[0][1]-invU[2][i]*U[0][2];
    }
}
```

```

printf("Inverse Matrix of L\n");
for(i=0;i<=2;i++)
{
    for(j=0;j<=2;j++)
    {
        printf("%.21f    ",invL[i][j]);
    }
    printf("\n");
}
printf("Inverse Matrix of U\n");
for(i=0;i<=2;i++)
{
    for(j=0;j<=2;j++)
    {
        printf("%.21f    ",invU[i][j]);
    }
    printf("\n");
}
double invA[3][3]={0};
int k;
printf("Inverse Matrix of A\n");
for(i=0;i<=2;i++)
{
    for(j=0;j<=2;j++)
    {
        for(k=0;k<=2;k++)
        {
            invA[i][j]=invA[i][j]+invU[i][k]*invL[k][j];
        }
        printf("%.21f    ",invA[i][j]);
    }
    printf("\n");
}
}

```

<Result>

```

C:\Users\Wjack8\Desktop\수치해석 과제\HW#03\W...
Matrix L
3.00  0.00  0.00
0.10  7.00  0.00
0.30 -0.19  10.02
Matrix U
1.00 -0.03 -0.07
0.00  1.00 -0.04
0.00  0.00  1.00
Inverse Matrix of L
0.33  0.00  0.00
-0.00  0.14  0.00
-0.01  0.00  0.10
Inverse Matrix of U
1.00  0.03  0.07
0.00  1.00  0.04
0.00  0.00  1.00
Inverse Matrix of A
0.33  0.00  0.01
-0.01  0.14  0.00
-0.01  0.00  0.10
-----
Process exited after 0.6802 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .

```

3. 다음 연립방정식을 TDMA로 풀어라.

$$\begin{bmatrix} 3 & -1 & 0 & 0 & 0 & 0 \\ -1 & 3 & -1 & 0 & 0 & 0 \\ 0 & -1 & 3 & -1 & 0 & 0 \\ 0 & 0 & -1 & 3 & -1 & 0 \\ 0 & 0 & 0 & -1 & 3 & -1 \\ 0 & 0 & 0 & 0 & -1 & 3 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \end{bmatrix} = \begin{bmatrix} 100 \\ 150 \\ 150 \\ 150 \\ 150 \\ 100 \end{bmatrix}$$

<Code>

```
#include<stdio.h>
int main()
{
    double a=-1,b=3,c=-1;
    double d[6]={100,150,150,150,150,100};
    double p[6]={0}, q[6]={0};
    p[0]=-c/b; q[0]=d[0]/b;
    int i;
    for(i=1;i<=5;i++)
    {
        q[i]=(d[i]-a*q[i-1])/(a*p[i-1]+b);
        if(i==5) break;
        p[i]=-c/(a*p[i-1]+b);
    }
    double x[6];
    x[5]=q[5];
    for(i=4;i>=0;i--)
    {
        x[i]=p[i]*x[i+1]+q[i];
    }
    printf("x= ");
    for(i=0;i<=5;i++)
    {
        printf("%1f",x[i]);
    }
    printf("\np= ");
    for(i=0;i<=5;i++)
    {
        printf("%1f",p[i]);
    }
    printf("\nq= ");
    for(i=0;i<=5;i++)
    {
        printf("%1f",q[i]);
    }
}
```

<Result>

```
C:\Users\Wjack8W\Desktop\수치해석 과제\WHW#03\WHW#03_3.exe
x= 73.076923    119.230769    134.615385    134.615385    119.230769    73.076923
p= 0.333333    0.375000    0.380952    0.381818    0.381944    0.000000
q= 33.333333    68.750000    83.333333    89.090909    91.319444    73.076923
-----
Process exited after 0.7134 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

4. 다음 연립방정식을 Jacobi 반복법과 Gauss-Seidel 법으로 각각 풀어라. 단, 방정식의 해는 유효숫자 5자리까지 정확히 구하여라.

$$\begin{aligned} 10x_1 - 3x_2 + 6x_3 &= 24.5 \\ x_1 + 8x_2 - 2x_3 &= -10 \\ -2x_1 + 4x_2 - 9x_3 &= -50 \end{aligned}$$

(a) Jacobi 반복법

<Code>

```
#include<stdio.h>
#include<math.h>
int main()
{
    double a[3][3]={10,-3,6},{1,8,-2},{-2,4,-9}};
    double b[3]={24.5,-10,-50};
    double x[3]={0};
    double X[3]={0};
    double error=0.5/pow(10,5);
    int i;
    for(i=1;i<=20;i++)
    {
        X[0]=(b[0]-a[0][1]*x[1]-a[0][2]*x[2])/a[0][0];
        X[1]=(b[1]-a[1][0]*x[0]-a[1][2]*x[2])/a[1][1];
        X[2]=(b[2]-a[2][0]*x[0]-a[2][1]*x[1])/a[2][2];
        printf("%d x1=%1f x2=%1f x3=%1f\n",i,X[0],X[1],X[2]);
        double e1,e2,e3;
        e1=fabs((X[0]-x[0])/X[0]);
        e2=fabs((X[1]-x[1])/X[1]);
        e3=fabs((X[2]-x[2])/X[2]);
        printf("error1=%1ferror2=%1f error3=%1f\n",e1,e2,e3);
        if(i!=1)
        {
            if(e1<error&&e2<error&&e3<error) break;
        }
        x[0]=X[0]; x[1]=X[1]; x[2]=X[2];
    }
}
```

(b) Gauss-Seidel 법

<Code>

```
#include<stdio.h>
#include<math.h>
int main()
{
    double a[3][3]={10,-3,6},{1,8,-2},{-2,4,-9}};
    double b[3]={24.5,-10,-50};
    double x[3]={0};
    double X[3]={0};
    double error=0.5/pow(10,5);
    int i;
    for(i=1;i<=20;i++)
    {
        X[0]=(b[0]-a[0][1]*x[1]-a[0][2]*x[2])/a[0][0];
        X[1]=(b[1]-a[1][0]*X[0]-a[1][2]*x[2])/a[1][1];
        X[2]=(b[2]-a[2][0]*X[0]-a[2][1]*X[1])/a[2][2];
        printf("%d x1=%1f x2=%1f x3=%1f\n",i,X[0],X[1],X[2]);
        double e1,e2,e3;
        e1=fabs((X[0]-x[0])/X[0]);
        e2=fabs((X[1]-x[1])/X[1]);
        e3=fabs((X[2]-x[2])/X[2]);
        printf("error1=%1ferror2=%1f error3=%1f\n",e1,e2,e3);
        if(i!=1)
        {
            if(e1<error&&e2<error&&e3<error) break;
        }
        x[0]=X[0]; x[1]=X[1]; x[2]=X[2];
    }
}
```

<Result>

(a) Jacobi 반복법

```
C:\Users\Wjack8W\Desktop\수치해석 과제...
1 x1=2.450000 x2=-1.250000 x3=5.555556
error1=1.000000 error2=1.000000 error3=1.000000
2 x1=-1.258333 x2=-0.167361 x3=4.455556
error1=2.947020 error2=6.468880 error3=0.246883
3 x1=-0.273542 x2=0.021181 x3=5.760802
error1=3.600152 error2=8.901639 error3=0.226574
4 x1=-1.000127 x2=0.224393 x3=5.625756
error1=0.726493 error2=0.905610 error3=0.024005
5 x1=-0.858136 x2=0.281455 x3=5.877536
error1=0.165465 error2=0.202738 error3=0.042838
6 x1=-0.992085 x2=0.326651 x3=5.871343
error1=0.135018 error2=0.138362 error3=0.001055
7 x1=-0.974811 x2=0.341847 x3=5.921197
error1=0.017721 error2=0.044451 error3=0.008420
8 x1=-1.000164 x2=0.352151 x3=5.924112
error1=0.025349 error2=0.029261 error3=0.000492
9 x1=-0.998822 x2=0.356049 x3=5.934326
error1=0.001344 error2=0.010948 error3=0.001721
10 x1=-1.003781 x2=0.358434 x3=5.935760
error1=0.004940 error2=0.006656 error3=0.000242
11 x1=-1.003926 x2=0.359413 x3=5.937922
error1=0.000144 error2=0.002722 error3=0.000364
12 x1=-1.004929 x2=0.359971 x3=5.938389
error1=0.000999 error2=0.001552 error3=0.000079
13 x1=-1.005042 x2=0.360213 x3=5.938860
error1=0.000112 error2=0.000672 error3=0.000079
14 x1=-1.005252 x2=0.360345 x3=5.938993
error1=0.000209 error2=0.000366 error3=0.000022
15 x1=-1.005292 x2=0.360405 x3=5.939098
error1=0.000040 error2=0.000165 error3=0.000018
16 x1=-1.005338 x2=0.360436 x3=5.939134
error1=0.000045 error2=0.000087 error3=0.000006
17 x1=-1.005349 x2=0.360451 x3=5.939158
error1=0.000012 error2=0.000040 error3=0.000004
18 x1=-1.005359 x2=0.360458 x3=5.939167
error1=0.000010 error2=0.000021 error3=0.000002
19 x1=-1.005363 x2=0.360462 x3=5.939172
error1=0.000003 error2=0.000010 error3=0.000001
20 x1=-1.005365 x2=0.360463 x3=5.939175
error1=0.000002 error2=0.000005 error3=0.000000
```

(b) Gauss-Seidel 법

```
C:\Users\Wjack8W\Desktop\수치해석 과제\HW#03W...
1 x1=2.450000 x2=-1.556250 x3=4.319444
error1=1.000000 error2=1.000000 error3=1.000000
2 x1=-0.608542 x2=-0.094071 x3=5.648978
error1=5.02602 error2=15.54332 error3=0.23536
3 x1=-0.967608 x2=0.283195 x3=5.896444
error1=0.37109 error2=1.33218 error3=0.04197
4 x1=-1.002908 x2=0.349475 x3=5.933746
error1=0.03520 error2=0.18965 error3=0.00629
5 x1=-1.005405 x2=0.359112 x3=5.938584
error1=0.00248 error2=0.02684 error3=0.00081
6 x1=-1.005417 x2=0.360323 x3=5.939125
error1=0.00001 error2=0.00336 error3=0.00009
7 x1=-1.005378 x2=0.360454 x3=5.939175
error1=0.00004 error2=0.00036 error3=0.00001
8 x1=-1.005369 x2=0.360465 x3=5.939177
error1=0.00001 error2=0.00003 error3=0.00000
9 x1=-1.005367 x2=0.360465 x3=5.939177
error1=0.00000 error2=0.00000 error3=0.00000

-----
Process exited after 0.5438 seconds with return value 0
계속하려면 아무 키나 누르십시오 . . .
```

5. 다음 연립방정식을 주어진 이완계수 w 을 갖고 SOR 또는 SUR로 풀어라. 그리고, Gauss-Seidel 법으로 풀어 반복횟수를 비교하여라. 단, 방정식의 해는 유효숫자 5자리 이내에 있도록 하라.

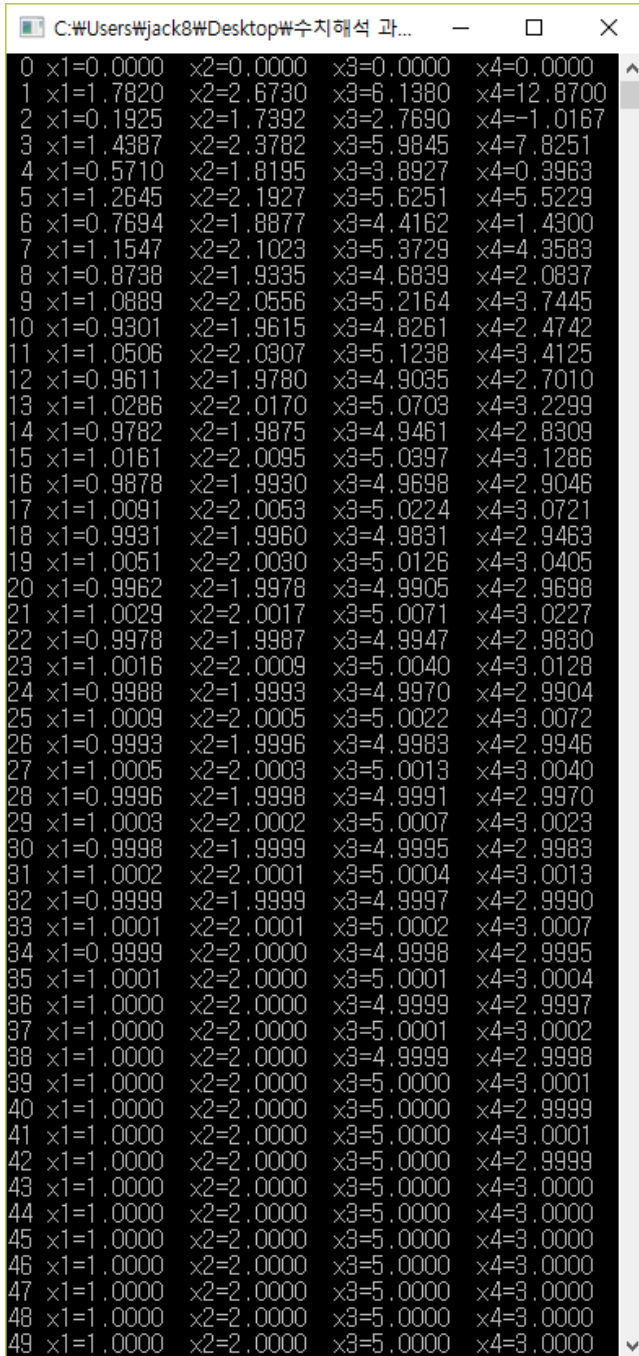
$$\begin{aligned} 2x_1 + 10x_2 + x_3 &= 27 \\ x_1 + x_2 + 5x_3 + x_4 &= 31 \\ 3x_1 + x_2 + x_3 + x_4 &= 13 \\ 15x_1 + 2x_2 + x_3 + x_4 &= 27 \\ w &= 0.99 \end{aligned}$$

<Code>

```
#include<stdio.h>
#include<math.h>
int main()
{
    double a[4][4]={{15,2,1,1},{2,10,1,0},{1,1,5,1},{3,1,1,1}};
    double b[4]={27,27,31,13};
    double w=0.99; //or 1
    double x[4]={0,0,0,0};
    double X[4]={0,};
    double error=0.5/pow(10,5);
    int i,j,k;
    for(i=1;i<=200;i++)
    {
        printf("%2d x1=%.41f x2=%.41f x3=%.41f x4=%.41f\n",i-1,x[0],x[1],x[2],x[3]);
        for(j=0;j<=3;j++)
        {
            X[j]=x[j]+w*b[j]/a[j][j];
            for(k=0;k<=j-1;k++)
            {
                X[j]=X[j]-w*a[j][k]*x[k]/a[j][j];
            }
            for(k=j;k<=3;k++)
            {
                X[j]=X[j]-w*a[j][k]*x[k]/a[j][j];
            }
        }
        double e1,e2,e3,e4;
        e1=fabs((X[0]-x[0])/X[0]);
        e2=fabs((X[1]-x[1])/X[1]);
        e3=fabs((X[2]-x[2])/X[2]);
        e4=fabs((X[3]-x[3])/X[3]);
        if(i!=1)
        {
            if(e1<error&&e2<error&&e3<error&&e4<error) break;
        }
        x[0]=X[0]; x[1]=X[1]; x[2]=X[2]; x[3]=X[3];
    }
}
```

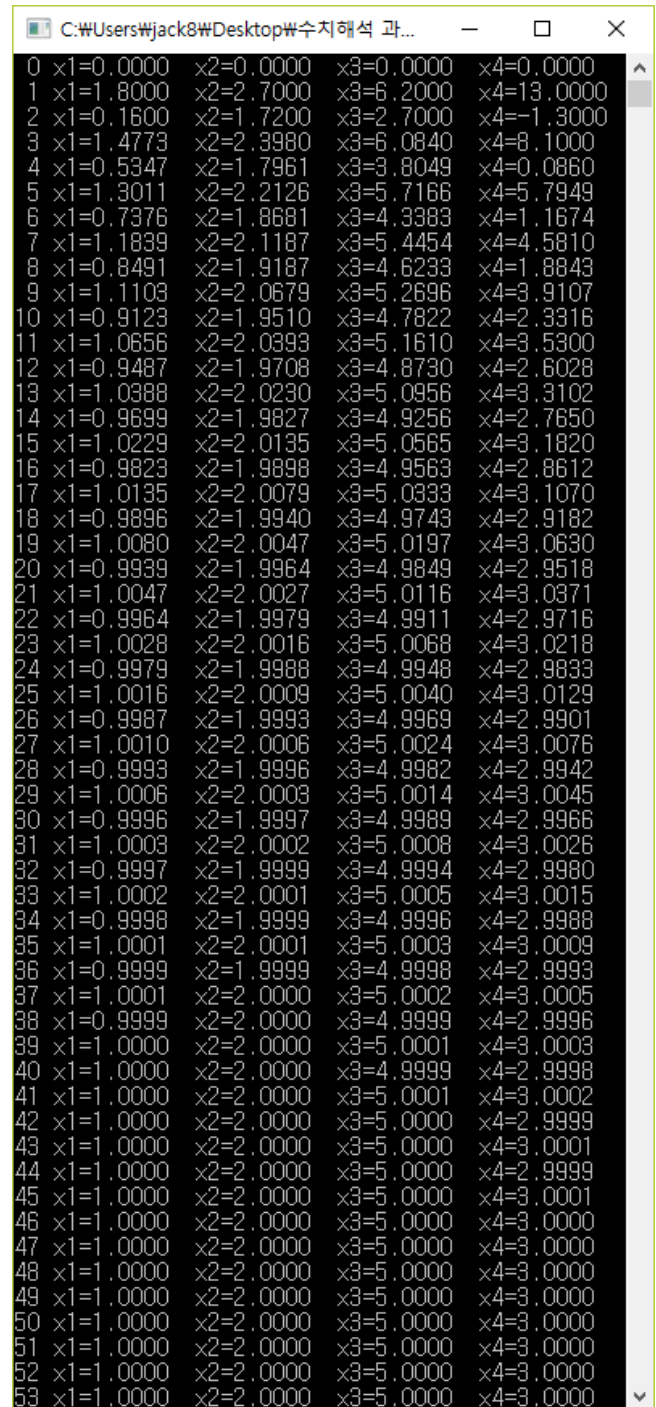
<Result>

(a) SUR($w=0.99 < 1$)



0	x1=0.0000	x2=0.0000	x3=0.0000	x4=0.0000
1	x1=1.7820	x2=2.6730	x3=6.1380	x4=12.8700
2	x1=0.1925	x2=1.7392	x3=2.7690	x4=-1.0167
3	x1=1.4387	x2=2.3782	x3=5.9845	x4=7.8251
4	x1=0.5710	x2=1.8195	x3=3.8927	x4=0.3963
5	x1=1.2645	x2=2.1927	x3=5.6251	x4=5.5229
6	x1=0.7694	x2=1.8877	x3=4.4162	x4=1.4300
7	x1=1.1547	x2=2.1023	x3=5.3729	x4=4.3583
8	x1=0.8738	x2=1.9335	x3=4.6839	x4=2.0837
9	x1=1.0889	x2=2.0556	x3=5.2164	x4=3.7445
10	x1=0.9301	x2=1.9615	x3=4.8261	x4=2.4742
11	x1=1.0506	x2=2.0307	x3=5.1238	x4=3.4125
12	x1=0.9611	x2=1.9780	x3=4.9035	x4=2.7010
13	x1=1.0286	x2=2.0170	x3=5.0703	x4=3.2299
14	x1=0.9782	x2=1.9875	x3=4.9461	x4=2.8309
15	x1=1.0161	x2=2.0095	x3=5.0397	x4=3.1286
16	x1=0.9878	x2=1.9930	x3=4.9698	x4=2.9046
17	x1=1.0091	x2=2.0053	x3=5.0224	x4=3.0721
18	x1=0.9931	x2=1.9960	x3=4.9831	x4=2.9463
19	x1=1.0051	x2=2.0030	x3=5.0126	x4=3.0405
20	x1=0.9962	x2=1.9978	x3=4.9905	x4=2.9698
21	x1=1.0029	x2=2.0017	x3=5.0071	x4=3.0227
22	x1=0.9978	x2=1.9987	x3=4.9947	x4=2.9830
23	x1=1.0016	x2=2.0009	x3=5.0040	x4=3.0128
24	x1=0.9988	x2=1.9993	x3=4.9970	x4=2.9904
25	x1=1.0009	x2=2.0005	x3=5.0022	x4=3.0072
26	x1=0.9993	x2=1.9996	x3=4.9983	x4=2.9946
27	x1=1.0005	x2=2.0003	x3=5.0013	x4=3.0040
28	x1=0.9996	x2=1.9998	x3=4.9991	x4=2.9970
29	x1=1.0003	x2=2.0002	x3=5.0007	x4=3.0023
30	x1=0.9998	x2=1.9999	x3=4.9995	x4=2.9983
31	x1=1.0002	x2=2.0001	x3=5.0004	x4=3.0013
32	x1=0.9999	x2=1.9999	x3=4.9997	x4=2.9990
33	x1=1.0001	x2=2.0001	x3=5.0002	x4=3.0007
34	x1=0.9999	x2=2.0000	x3=4.9998	x4=2.9995
35	x1=1.0001	x2=2.0000	x3=5.0001	x4=3.0004
36	x1=1.0000	x2=2.0000	x3=4.9999	x4=2.9997
37	x1=1.0000	x2=2.0000	x3=5.0001	x4=3.0002
38	x1=1.0000	x2=2.0000	x3=4.9999	x4=2.9998
39	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0001
40	x1=1.0000	x2=2.0000	x3=5.0000	x4=2.9999
41	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0001
42	x1=1.0000	x2=2.0000	x3=5.0000	x4=2.9999
43	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
44	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
45	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
46	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
47	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
48	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
49	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000

(b) Gauss-Seidel 법($w=1$)



0	x1=0.0000	x2=0.0000	x3=0.0000	x4=0.0000
1	x1=1.8000	x2=2.7000	x3=6.2000	x4=13.0000
2	x1=0.1600	x2=1.7200	x3=2.7000	x4=-1.3000
3	x1=1.4773	x2=2.3980	x3=6.0840	x4=8.1000
4	x1=0.5347	x2=1.7961	x3=3.8049	x4=0.0860
5	x1=1.3011	x2=2.2126	x3=5.7166	x4=5.7949
6	x1=0.7376	x2=1.8681	x3=4.3383	x4=1.1674
7	x1=1.1839	x2=2.1187	x3=5.4454	x4=4.5810
8	x1=0.8491	x2=1.9187	x3=4.6233	x4=1.8843
9	x1=1.1103	x2=2.0679	x3=5.2696	x4=3.9107
10	x1=0.9123	x2=1.9510	x3=4.7822	x4=2.3316
11	x1=1.0656	x2=2.0393	x3=5.1610	x4=3.5300
12	x1=0.9487	x2=1.9708	x3=4.8730	x4=2.6028
13	x1=1.0388	x2=2.0230	x3=5.0956	x4=3.3102
14	x1=0.9699	x2=1.9827	x3=4.9256	x4=2.7650
15	x1=1.0229	x2=2.0135	x3=5.0565	x4=3.1820
16	x1=0.9823	x2=1.9898	x3=4.9563	x4=2.8612
17	x1=1.0135	x2=2.0079	x3=5.0333	x4=3.1070
18	x1=0.9896	x2=1.9940	x3=4.9743	x4=2.9182
19	x1=1.0080	x2=2.0047	x3=5.0197	x4=3.0630
20	x1=0.9939	x2=1.9964	x3=4.9849	x4=2.9518
21	x1=1.0047	x2=2.0027	x3=5.0116	x4=3.0371
22	x1=0.9964	x2=1.9979	x3=4.9911	x4=2.9716
23	x1=1.0028	x2=2.0016	x3=5.0068	x4=3.0218
24	x1=0.9979	x2=1.9988	x3=4.9948	x4=2.9833
25	x1=1.0016	x2=2.0009	x3=5.0040	x4=3.0129
26	x1=0.9987	x2=1.9993	x3=4.9969	x4=2.9901
27	x1=1.0010	x2=2.0006	x3=5.0024	x4=3.0076
28	x1=0.9993	x2=1.9996	x3=4.9982	x4=2.9942
29	x1=1.0006	x2=2.0003	x3=5.0014	x4=3.0045
30	x1=0.9996	x2=1.9997	x3=4.9989	x4=2.9966
31	x1=1.0003	x2=2.0002	x3=5.0008	x4=3.0026
32	x1=0.9997	x2=1.9999	x3=4.9994	x4=2.9980
33	x1=1.0002	x2=2.0001	x3=5.0005	x4=3.0015
34	x1=0.9998	x2=1.9999	x3=4.9996	x4=2.9988
35	x1=1.0001	x2=2.0001	x3=5.0003	x4=3.0009
36	x1=0.9999	x2=1.9999	x3=4.9998	x4=2.9993
37	x1=1.0001	x2=2.0000	x3=5.0002	x4=3.0005
38	x1=1.0000	x2=2.0000	x3=4.9999	x4=2.9996
39	x1=1.0000	x2=2.0000	x3=5.0001	x4=3.0003
40	x1=1.0000	x2=2.0000	x3=4.9999	x4=2.9998
41	x1=1.0000	x2=2.0000	x3=5.0001	x4=3.0002
42	x1=1.0000	x2=2.0000	x3=5.0000	x4=2.9999
43	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0001
44	x1=1.0000	x2=2.0000	x3=5.0000	x4=2.9999
45	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0001
46	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
47	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
48	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
49	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
50	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
51	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
52	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000
53	x1=1.0000	x2=2.0000	x3=5.0000	x4=3.0000