

```

1 import os, sys
2 sys.path.append(os.path.join(os.path.dirname(__file__), '.././ch02'))
3
4 from program2_1 import Dvector
5 from program2_2 import Dmatrix
6 from program2_3 import input_vector, input_matrix
7
8 N = 4 # N次正方行列
9
10 def main():
11     global N
12     a = Dmatrix(1, N, 1, N) # 行列 a[1...N][1...N]
13     b = Dvector(1, N) # b[1...N]
14
15     # ファイルのオープン
16     with open("input.dat", "r") as fin:
17         with open("output.dat", "w") as fout:
18             input_matrix( a, 'A', fin, fout ) # 行列 A の入出力
19             input_vector( b, 'b', fin, fout ) # ベクトル b の入出力
20             b = simple_gauss( a, b ) # ガウス消去法
21
22             # 結果の出力
23             fout.write("Ax=b の解は次の通りです\n")
24             for i in range(1, N+1):
25                 fout.write("{:.6f}\n".format(b[i]))
26
27
28 # ガウス消去法
29 def simple_gauss(a: Dmatrix, b: Dvector):
30     # 前進消去
31     for k in range(1, N):
32         for i in range(k+1, N+1):
33             alpha = - a[i][k] / a[k][k]
34             for j in range(k+1, N+1):
35                 a[i][j] += alpha * a[k][j]
36             b[i] += alpha * b[k]
37
38     # 後退代入
39     b[N] /= a[N][N]
40     for k in range(N-1, 0, -1):
41         tmp = b[k]
42         for j in range(k+1, N+1):
43             tmp -= a[k][j] * b[j]
44         b[k] = tmp / a[k][k]
45
46     return b
47
48
49 if __name__ == "__main__":
50     main()

```