

```

1 #include <stdio.h>
2 #include <stdlib.h> /* srand, rand 関数を使うために必要 */
3 #include <time.h> /* time 関数を使うために必要 */
4
5 #define ROW 3 /* 行の要素数 */
6 #define COLUMN 4 /* 列の要素数 */
7
8 /* 行列の領域確保 */
9 double **dmatrix(int nr1, int nr2, int n11, int n12);
10 /* 行列の領域解放 */
11 void free_dmatrix(double **a, int nr1, int nr2, int n11, int n12);
12 /* 行列の生成 a[nr1...nr2][n11...n12] */
13 double **produce_matrix( double **a, int nr1, int nr2, int n11, int n12);
14
15 int main(void)
16 {
17     double **a;
18     int i, j;
19
20     a = dmatrix(1, ROW, 1, COLUMN); /* 行列 a[1...ROW][1...COLUMN] */
21     a = produce_matrix( a, 1, ROW, 1, COLUMN); /* 行列の生成 */
22
23     /* 行列の表示 */
24     for ( i = 1 ; i <= ROW ; i++)
25     {
26         for ( j = 1 ; j <= COLUMN ; j++)
27         {
28             printf("%f\t",a[i][j]);
29         }
30         printf("\n");
31     }
32
33     free_dmatrix( a, 1, ROW, 1, COLUMN); /* 行列領域の解放 */
34
35     return 0;
36 }
37
38 /* 行列の生成 a[nr1...nr2][n11...n12] */
39 double **produce_matrix( double **a, int nr1, int nr2, int n11, int n12)
40 {
41     int i, j ;
42
43     srand( (unsigned)time(NULL)); /* シードを与える */
44     for ( i = nr1 ; i <= nr2 ; i++)
45     {
46         for ( j = n11 ; j <= n12 ; j++)
47         {
48             a[i][j] = (double)rand() / (double)RAND_MAX;
49         }
50     }
51     return a;
52 }
53
54 double **dmatrix(int nr1, int nr2, int n11, int n12)
55 {
56     int i, nrow, ncol;
57     double **a;
58
59     nrow = nr2 - nr1 + 1; /* 行の数 */
60     ncol = n12 - n11 + 1; /* 列の数 */
61
62     /* 行の確保 */
63     if ((a = malloc(nrow * sizeof(double *))) == NULL)
64     {
65         printf("メモリが確保できません (行列 a)\n");
66         exit(1);
67     }
68     a = a - nr1; /* 行をずらす */
69
70     /* 列の確保 */
71     for (i = nr1; i <= nr2; i++)
72         a[i] = malloc(ncol * sizeof(double));
73     for (i = nr1; i <= nr2; i++)
74         a[i] = a[i] - n11; /* 列をずらす */
75
76     return (a);
77 }
78
79 void free_dmatrix(double **a, int nr1, int nr2, int n11, int n12)
80 {
81     int i;
82
83     /* メモリの解放 */
84     for (i = nr1; i <= nr2; i++)
85         free((void *) (a[i] + n11));
86     free((void *) (a + nr1));
87 }

```