

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define M 2 /* 行の要素数 */
5  #define N 3 /* 列の要素数 */
6
7  /* 行列の領域確保 */
8  double **dmatrix(int nr1, int nr2, int nl1, int nl2);
9  /* 行列の領域解放 */
10 void free_dmatrix(double **a, int nr1, int nr2, int nl1, int nl2);
11
12 int main(void)
13 {
14     double **a;
15     int i, j;
16
17     a = dmatrix(1, M, 1, N); /* メモリの確保 */
18
19     for(i = 1; i <= M; i++){ /* 要素の代入 */
20         for(j = 1; j <= N; j++){
21             a[i][j] = (i + j) / 2.0;
22             printf("a[%d][%d] = %f \n", i, j, a[i][j]);
23         }
24     }
25     free_dmatrix(a, 1, M, 1, N); /* メモリの解放 */
26
27     return 0;
28 }
29
30 double **dmatrix(int nr1, int nr2, int nl1, int nl2)
31 {
32     int i, nrow, ncol;
33     double **a;
34
35     nrow = nr2 - nr1 + 1; /* 行の数 */
36     ncol = nl2 - nl1 + 1; /* 列の数 */
37
38     /* 行の確保 */
39     if((a = malloc(nrow * sizeof(double *))) == NULL){
40         printf("メモリが確保できません (行列 a)\n");
41         exit(1);
42     }
43     a = a - nr1; /* 行をずらす */
44
45     /* 列の確保 */
46     for(i = nr1; i <= nr2; i++) a[i] = malloc(ncol * sizeof(double));
47     for(i = nr1; i <= nr2; i++) a[i] = a[i] - nl1; /* 列をずらす */
48
49     return (a);
50 }
51
52 void free_dmatrix(double **a, int nr1, int nr2, int nl1, int nl2)
53 {
54     int i;
55
56     /* メモリの解放 */
57     for(i = nr1; i <= nr2; i++) free((void *) (a[i] + nl1));
58     free((void *) (a + nr1));
59 }

```