

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  #define N 5 /* 要素数の指定 */
5
6  double *dvector(int i, int j); /* ベクトル領域の確保 */
7  void free_dvector(double *a, int i); /* 領域の解放 */
8
9  /* ベクトル a[m...n] と b[m...n] の内積を計算する */
10 double inner_product(int m, int n, double *a, double *b);
11
12 int main(void)
13 {
14     double *a, *b;
15     int i;
16
17     /* ベクトルの定義, 配列 a,b の添字は 1~N */
18     a = dvector(1,N); b = dvector(1,N);
19     for(i = 1; i <= N; i++){
20         a[i] = (double)i / 20.0;
21         b[i] = (double)i / 10.0;
22     }
23
24     printf("aとbの内積は%f です\n", inner_product(1, N, a, b));
25     free_dvector(a,1); free_dvector(b,1);
26
27     return 0;
28 }
29
30 /* ベクトル a[m...n] と b[m...n] の内積を計算する */
31 double inner_product(int m, int n, double *a, double *b)
32 {
33     int i;
34     double s = 0.0;
35
36     for (i = m; i <= n; i++)
37         s += a[i] * b[i];
38
39     return s;
40 }
41
42 double *dvector(int i, int j) /* a[i]~a[j] の領域を確保 */
43 {
44     double *a;
45
46     if((a = malloc( ((j - i + 1) * sizeof(double))) ) == NULL)
47     {
48         printf("メモリが確保できません (from dvector) \n");
49         exit(1);
50     }
51
52     return (a - i);
53 }
54
55 void free_dvector(double *a, int i)
56 {
57     free( (void *) (a + i) ); /* (void *) 型へのキャストが必要 */
58 }

```