

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  /* 被積分関数の定義 */
5  double func(double x, double y);
6
7  /* y の積分区間 */
8  double phi(double x);
9  double psi(double x);
10
11 /* ベクトル領域の確保 */
12 double *dvector(int i, int j);
13 /* ベクトル領域の解放 */
14 void free_dvector(double *a, int i);
15
16 /* 重積分用の台形公式 */
17 double trapezoidal2( double a, double b, int m, int n, double (*p)(double),
18                     double (*q)(double), double (*f)(double,double) );
19
20 int main(void)
21 {
22     int n=20, m=20;
23
24     printf("8x^2+4y を x=[1,2], y=[2-x,x^2] で積分します \n");
25     printf("x の分割数は%d, y の分割数は%d, 結果は%15.10f\n", m, n,
26           trapezoidal2( 1.0, 2.0, m, n, phi, psi, func ) );
27
28     return 0;
29 }
30
31 /* 重積分の台形公式 */
32 double trapezoidal2( double a, double b, int m, int n, double (*p)(double),
33                     double (*q)(double), double (*f)(double,double) )
34 {
35     double T, h, k, *F, x, y1, y2;
36     int i, j;
37
38     F = dvector( 0, n );
39     h = ( b - a ) / n ;           /* 刻み幅の指定 (x 方向) */
40
41     /* F_i の計算 */
42     for ( i = 0; i <= n; i++ )
43     {
44         x = a + i*h;
45         y1 = (*p)(x); y2 = (*q)(x);
46         k = ( y2 - y1 )/m;        /* 刻み幅の指定 (y 方向) */
47         F[i] = ( (*f)(x, y1) + (*f)(x, y2) ) / 2.0;
48         for ( j = 1; j < m; j++ ) F[i] += (*f)(x, y1+j*k);
49         F[i] *= k;
50     }
51
52     /* 積分の計算 */
53     T = ( F[0] + F[n] ) / 2.0;
54     for ( i = 1; i < n; i++ ) T += F[i];
55     T *= h;
56
57     free_dvector( F, 0 );
58
59     return T;
60 }
61
62 /* 被積分関数の定義 */
63 double func(double x, double y)
64 {
65     return( 8.0*x*x + 4.0*y );
66 }
67
68 /* y の積分区間 */
69 double phi(double x)
70 {
71     return( 2.0-x );
72 }
73
74 double psi(double x)
75 {
76     return( x*x );
77 }
78
79 double *dvector(int i, int j) /* a[i]~a[j] の領域を確保 */
80 {
81     double *a;
82
83     if ((a = malloc(((j - i + 1) * sizeof(double)))) == NULL)
84     {
85         printf("メモリが確保できません (from dvector) \n");
86         exit(1);
87     }
88
89     return (a - i);
90 }
91
92 void free_dvector(double *a, int i)
93 {
94     free((void *) (a + i)); /* (void *) 型へのキャストが必要 */
95 }

```