

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 double *dvector(long i, long j);      /* ベクトル領域の確保 */
5 void free_dvector(double *a, long i); /* 領域の解放 */
6 double func(double x, double y);     /* 関数の定義 */
7 /* ルンゲ・クッタ法 */
8 double *rk4( double y0, double *y, double a, double b, int n,
9              double (*f)(double, double) );
10
11 int main(void)
12 {
13     double *y, h, a=0.0, b=1.0, y0=1.0 ;
14     int i, n;
15
16     printf("分割数を入力してください--->");
17     scanf("%d",&n);
18
19     y = dvector( 0, n );              /* 領域の確保 */
20     y = rk4( y0, y, a, b, n, func ); /* ルンゲ・クッタ法 */
21
22     /* 結果の表示 */
23     h = (b-a)/n ;                    /* 刻み幅 */
24     for ( i = 0 ; i <= n ; i++)
25     {
26         printf("x=%f \t y=%f \n", a+i*h, y[i] );
27     }
28
29     free_dvector( y, 0 ); /* 領域の解放 */
30     return 0;
31 }
32
33 /* ルンゲ・クッタ法 */
34 double *rk4( double y0, double *y, double a, double b, int n,
35              double (*f)(double, double) )
36 {
37     double k1, k2, k3, k4, h, x;
38     int i;
39
40     h = (b-a)/n;
41     /* 初期値の設定 */
42     y[0] = y0; x = a;
43
44     /* ルンゲ・クッタ法 */
45     for ( i = 0 ; i < n ; i++)
46     {
47         k1 = f(x,y[i]); k2 = f(x+h/2.0, y[i]+h*k1/2.0);
48         k3 = f(x+h/2.0, y[i]+h*k2/2.0);
49         k4 = f(x+h, y[i]+h*k3);
50         y[i+1] = y[i] + h/6.0 * ( k1 + 2.0*k2 + 2.0*k3 + k4 );
51         x += h;
52     }
53
54     return y;
55 }
56
57 /* 関数の定義 */
58 double func(double x, double y)
59 {
60     return( x + y );
61 }
62
63 double *dvector(long i, long j) /* a[i]~a[j] の領域を確保 */
64 {
65     double *a;
66
67     if((a = malloc( ((j - i + 1) * sizeof(double))) ) == NULL)
68     {
69         printf("メモリが確保できません (from dvector) \n");
70         exit(1);
71     }
72
73     return (a - i);
74 }
75
76 void free_dvector(double *a, long i)
77 {
78     free( (void *) (a + i) ); /* (void *) 型へのキャストが必要 */
79 }

```