```c
#include <stdio.h>
#include <math.h>
#include <stdlib.h>

double *dvector(long i, long j);       /* ベクトル領域の確保 */
void free_dvector(double *a, long i); /* 領域の解放 */
double func(double x, double y);       /* 関数の定義 */
/* ルンゲ・クッタ法 */
double *rk4( double y0, double *y, double a, double b, int n,
                                    double (*f)(double, double) );
/* アダムス法 */
double *adams( double y0, double *y, double a, double b, int n,
               int N, double eps, double (*f)(double, double) );

int main(void)
{
  double *y, h, a=0.0, b=1.0, y0=1.0, eps=pow(10.0,-8.0) ;
  int i, n, N=10;   /* 最大反復回数 N */

  printf("分割数を入力してください--->");
  scanf("%d",&n);

  y = dvector( 0, n );       /* 領域の確保 */

  /* アダムス法 */
  y = adams( y0, y, a, b, n, N, eps, func );

  /* 結果の表示 */
  h = (b-a)/n ;              /* 刻み幅 */
  for ( i = 0 ; i <= n ; i++)
  {
    printf("x=%f \t y=%f \n", a+i*h, y[i] );
  }

  free_dvector( y, 0 );
  return 0;
}

/* アダムス法 */
double *adams( double y0, double *y, double a, double b, int n,
               int N, double eps, double (*f)(double, double) )
{
  double yp, h, *F, x;
  int i, j;

  y = dvector( 0, n );   /* y[0,1,...n] の確保 */
  F = dvector( 0, 4 );   /* F[0,1,...4] の確保 */
  h = (b-a)/n;           /* 刻み幅の設定 */

  /* スタータ */
  y = rk4( y0, y, a, b, n, f);
  x = a;
  for ( i = 0; i <= 3; i++)
  {
    F[i] = f(x,y[i]); x += h;
  }

  /* 反復計算 */
  for ( i = 3; i <= n-1; i++)
  {
    /* アダムス・バッシュフォース法 */
    F[3] = f(x-h, y[i]);
    yp = y[i] + h*(55.0*F[3]-59.0*F[2]+37.0*F[1]-9.0*F[0])/24.0;
    for ( j = 1; j <= N; j++ )
    {
      /* アダムス・ムルトン法 */
      F[4] = f(x,yp);
      y[i+1] = y[i] + h*(9.0*F[4]+19.0*F[3]-5.0*F[2]+F[1])/24.0;
      if ( fabs(y[i+1]-yp) < eps ) break;
      yp = y[i+1];
    }
    for ( j = 1; j <= 4; j++) F[j-1] = F[j];  /* F[i] の更新 */
    x += h;
  }

  free_dvector( F, 0 ); /* 領域の解放 */
  return y;

}

/* ルンゲ・クッタ法 */
double *rk4(double y0, double *y, double a, double b, int n,
            double (*f)(double, double))
{
  double k1, k2, k3, k4, h, x;
  int i;

  h = (b - a) / n;
  /* 初期値の設定 */
  y[0] = y0;
  x = a;

  /* ルンゲ・クッタ法 */
  for (i = 0; i < n; i++)
  {
    k1 = f(x, y[i]);
    k2 = f(x + h / 2.0, y[i] + h * k1 / 2.0);
    k3 = f(x + h / 2.0, y[i] + h * k2 / 2.0);
    k4 = f(x + h, y[i] + h * k3);
    y[i + 1] = y[i] + h / 6.0 * (k1 + 2.0 * k2 + 2.0 * k3 + k4);
    x += h;
  }

  return y;
}

/* 関数の定義 */
double func(double x, double y)
{
  return( x + y );
}

double *dvector(long i, long j) /* a[i]～a[j] の領域を確保 */
{
  double *a;

  if ((a = malloc(((j - i + 1) * sizeof(double)))) == NULL)
  {
    printf("メモリが確保できません (from dvector) \n");
    exit(1);
  }

  return (a - i);
}

void free_dvector(double *a, long i)
{
  free((void *)(a + i)); /* (void *) 型へのキャストが必要 */
}
```