Программа на языке Си

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

// Определение функции f(x)
double f(double x) {
   return exp(-x*x);
}

// Функция для численного интегрирования методом центральных прямоугольников
double integrate(double a, double b, int N) {
   double h = (b - a) / N; // Шаг разбиения
   double integral = 0.0;

   for (int i = 0; i < N; i++) {
      double x_middle = a + h * (i + 0.5); // Точка в середине каждого подотрезка
      integral += f(x_middle);
   }

   integral *= h;
   return integral;
}

int main() {
   double a, b;
   int N;

   // Запрос у пользователя на ввод интервалов и количества разбиений
   printf("Введите значение a: ");
   scanf("%lf", &a);
   printf("Введите значение b: ");
   scanf("%lf", &b);
   printf("Введите количество разбиений N: ");
   scanf("%d", &N);

   // Вычисление интеграла
   double result = integrate(a, b, N);

   // Вывод результата
   printf("Численное значение интеграла: %.6f\n", result);

   return 0;
}
```

Объяснение кода

1. **Функция `f`:**

```
double f(double x) {
   return exp(-x*x);
}
```

Эта функция определяет произвольную функцию f(x), которую мы будем интегрировать. В данном случае, $f(x)=\exp(-x^2)$.

2. **Функция `integrate`:**

```
double integrate(double a, double b, int N) {
    double h = (b - a) / N; // Шаг разбиения
    double integral = 0.0;

    for (int i = 0; i < N; i++) {
        double x_middle = a + h * (i + 0.5); // Точка в середине каждого подотрезка
        integral += f(x_middle);
    }

    integral *= h;
    return integral;
}
```

Эта функция вычисляет численное значение интеграла методом центральных прямоугольников. Она принимает границы отрезка [a, b] и количество разбиений N.

3. **Функция `main`:**

```
int main() {
    double a, b;
    int N;

    // Запрос у пользователя на ввод интервалов и количества разбиений
    printf("Введите значение a: ");
    scanf("%lf", &a);
    printf("Введите значение b: ");
    scanf("%lf", &b);
    printf("Введите количество разбиений N: ");
    scanf("%d", &N);

    // Вычисление интеграла
    double result = integrate(a, b, N);

    // Вывод результата
    printf("Численное значение интеграла: %.6f\n", result);

    return 0;
}
```

Эта функция запрашивает у пользователя границы интервала [a, b] и количество разбиений N. Затем вызывает функцию `integrate` для вычисления интеграла и выводит результат.

Проверка работы программы

Для проверки работы программы можно ввести значения:
- a = 0
- b = 1
- N = 1000

Программа должна вывести численное значение интеграла функции $f(x)=\exp(-x^2)$на отрезке[0, 1]. = 0.746824

Метод центральных прямоугольников является одним из методов численного интегрирования, который используется для приближенного вычисления определенных интегралов. Он заключается в разбиении отрезка интегрирования на небольшие подотрезки и вычислении значения интеграла как суммы значений функции в серединах этих подотрезков, умноженных на ширину каждого подотрезка.

Связь кода с методом центральных прямоугольников

1. **Определение функции f(x):**

```
double f(double x) {
    return exp(-x*x);
}
```

Здесь мы определяем произвольную функцию f(x), которую будем интегрировать. В данном примере это $\mathbf{f(x)=\exp(-x^2)}$.

2. **Численное интегрирование методом центральных прямоугольников:**

```
double integrate(double a, double b, int N) {
    double h = (b - a) / N; // Шаг разбиения
    double integral = 0.0;

    for (int i = 0; i < N; i++) {
        double x_middle = a + h * (i + 0.5); // Точка в середине каждого подотрезка
        integral += f(x_middle);
    }

    integral *= h;
    return integral;
}
```

Шаги метода:

- Шаг разбиения (h):

```
double h = (b - a) / N;
```

Отрезок [a, b]делится на N равных частей с шириной h .

- Итерация по каждому подотрезку:

```
for (int i = 0; i < N; i++) {
    double x_middle = a + h * (i + 0.5); // Точка в середине каждого подотрезка
    integral += f(x_middle);
}
```

Мы проходим по каждому подотрезку, вычисляем значение функции f в середине каждого подотрезка и суммируем эти значения.

- **Умножение на ширину подотрезка:**

```
integral *= h;
```

После суммирования значений функции в серединах всех подотрезков, мы умножаем эту сумму на ширину подотрезка h, чтобы получить приближенное значение интеграла.