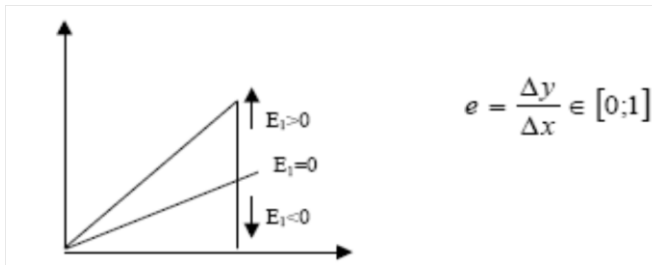


## 7. Алгоритм растровой развертки отрезков, прямых, окружностей.

Существует два вида реализации этого алгоритма: **вещественный** и **целочисленный**.

**Вещественный** служит базой для построения целочисленного алгоритма. Работа вещественного алгоритма базируется на расчете дополнительной переменной – оценки отложения точки аппроксимации от истинного направления (обозначим эту оценку как  $e$ ).



Рассмотрим работу алгоритма в 1 квадранте. Для удобства работы величину  $e$  изменяют таким образом, чтобы она в крайних точках имела противоположные значения:

$$e_1 = \frac{\Delta y}{\Delta x} - \frac{1}{2} \quad e_1 \in \left[-\frac{1}{2}; +\frac{1}{2}\right]$$

Недостатки: наличие операций деления; вещественный характер вычислений.

Для того, чтобы работать в одной области определения с растровыми данными, осуществляется **переход к целочисленным значениям оценки  $e$** :

$$\bar{e} = 2\Delta x e_1 = 2\Delta y - \Delta x \quad \bar{e} \in [-1; +1]$$

Алгоритм основанный на данном вычислении оценки, позволяет эффективно реализовать растровые разложения отрезка как на аппаратном, так и на программном уровнях.

### Растровая развертка окружностей

Построения окружностей и эллипсов можно осуществлять двумя способами:

- 1) используя уравнения тригонометрии и аналитической геометрии;
- 2) с использованием численных методов.

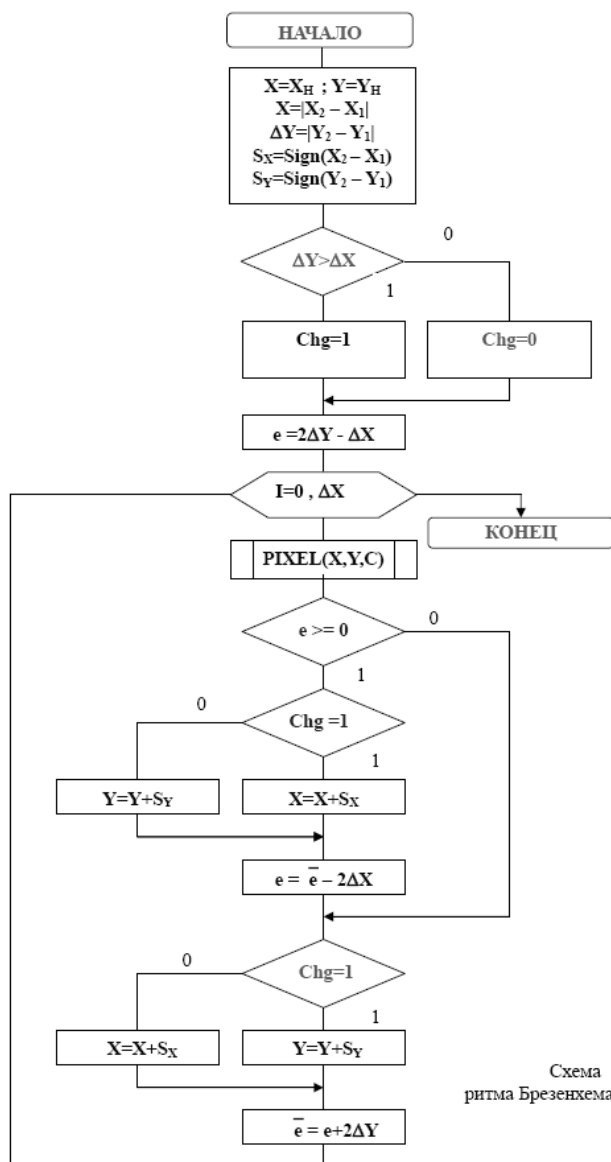
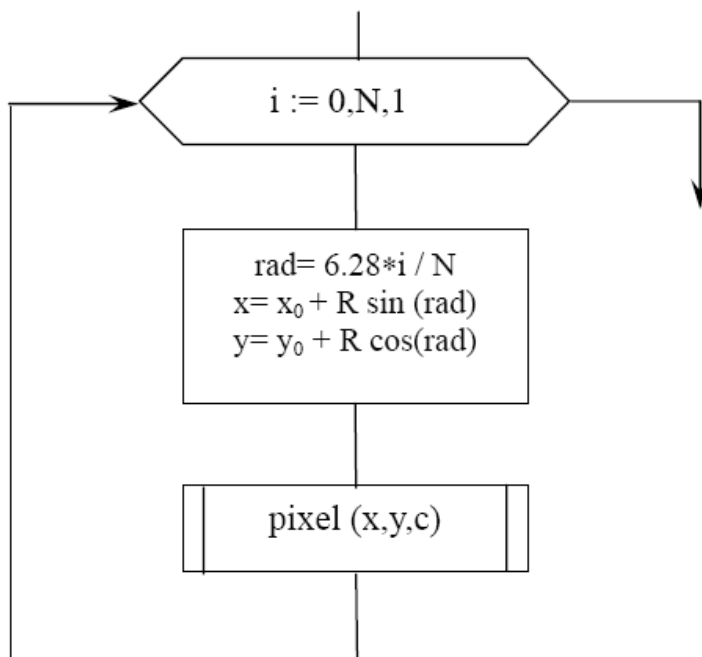


Схема  
ритма Брезенхема



**Построение окружности с использованием аналитических выражений.** В простейшем случае отображение окружности в растр можно осуществить при помощи аналитической зависимости между координатами X и Y. Алгоритм будет иметь вид:

Данный алгоритм легко модернизировать для случая построения эллипса. Необходимо заменить радиус  $R$  в уравнениях для вычисления координат на полуоси  $R_x$  и  $R_y$ . Аналогичным образом можно учесть другое соотношение координат окружности вида:

$$x^2 + y^2 = R^2$$

Отсюда можно вычислить соответствующее значение  $y$ :

$$y = \sqrt{R^2 - x^2}$$

И в том и в другом случае алгоритм растровой развертки окружности достаточно прост для программирования, однако, его вычислительная сложность слишком велика для реализации в составе ядра базовой графической системы. Это объясняется наличием тригонометрических функций в первом случае и степенных – во втором. Поэтому использовать подобные процедуры или функции в составе базовой графической системы не целесообразно. Необходимо разработать такие алгоритмы, которые бы максимально эффективно выполняли растровую развертку окружности при минимальной вычислительной сложности. Один из них приведен ниже.