

Поиск значения монотонной функции, записанной в массиве, заключается в сравнении срединного элемента массива с искомым значением, и повторением алгоритма для той или другой половины, в зависимости от результата сравнения.

Пусть переменные L_b и U_b содержат, соответственно, левую и правую границы отрезка массива, где находится нужный нам элемент. Исследования начинаются со среднего элемента отрезка. Если искомое значение меньше среднего элемента, осуществляется переход к поиску в верхней половине отрезка, где все элементы меньше только что проверенного, то есть значением U_b становится $(M - 1)_i$ на следующей итерации исследуется только половина массива. Т.о., в результате каждой проверки область поиска сужается вдвое.

Например, если длина массива равна 1023, после первого сравнения область сужается до 511 элементов, а после второй — до 255. Т.о. для поиска в массиве из 1023 элементов достаточно 10 сравнений.

```
l = леваяГраница - 1
r = праваяГраница + 1
while (r - l > 1) {
    середина = (l+r) / 2
    if (массив[середина] < значение)
        l = середина
    else
        r = середина
}
if (массив[r-1] != значение)
    return -1 // элемент не найден
else
    return r
```

Программный код

- x_n — начало отрезка по x ;
- x_k — конец отрезка по x ;
- x_i — середина отрезка по x ;
- eps — требуемая точность вычислений.

Таким образом, весь алгоритм можно записать следующим образом (в псевдокоде):

1. Начало.
2. Ввод x_n, x_k, eps .
3. Если $F(x_n) = 0$, то Вывод (корень уравнения — x_n).
4. Если $F(x_k) = 0$, то Вывод (корень уравнения — x_k).
5. Пока $(F(x_i) \neq 0)$ и $|x_k - x_n| > eps$ повторять:
6. $x_i := (x_k + x_n) / 2$;
7. если $(F(x_n) * F(x_i) < 0)$, то $x_k := x_i$;
8. если $(F(x_i) * F(x_k) < 0)$, то $x_n := x_i$.
9. Вывод (Найден корень уравнения — x_i точности ε).
10. Конец.

Пример реализации алгоритма на языке Matlab