```
/* binary search only works if f(x) increases w.r.t of x
 * but in this problem, f(x) is unimodal
 * that is, at first, increasing x, f(x) increases, after a certain value, increasing x causes f(x) to
decrease
 *      |      /                              |
 *      |     /                               |
 *      |    /                                |    ^
 *      |   /                                 |   / \
 *      |  /                                  |  /   \
 *   ^  | /                                ^   | /     \
 *   |  |/                                 |   |/       \
 * f(x)|-------------------               f(x) |-------------------
 *      x ->                                      x ->
 *
 *  Binary Search Works Here      Ternary Search Works Here
 */

long double ternarySearch(long double lo, long double hi) {
      long double l1, l2, ans;
      for(int i = 0; i < 100; ++i) {
            l1 = lo+(hi-lo)/3.0;
            l2 = hi-(hi-lo)/3.0;
            if(f(l1) > f(l2)) {                              // Function f(x) generates output for input x
                  hi = l2;
                  ans = l1;
            }
            else
                  lo = l1;
      }
      return ans;
}


// Bisection Loop:
double Bisect(double lo, double hi) {
      double ans, mid;
          for(register int i = 0; i < 50; i++) {
             mid = (lo+hi)/2.0;
             if(f(mid) <= EPS) {         // As it returns double, for precision EPS is added instead of
                ans = mid;               // f(mid) == 0
                hi = mid;
             }
             else
                lo = mid;
          }
      return ans;
}
```