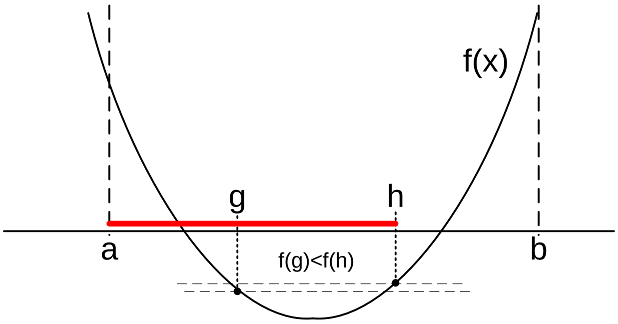
Ternary search

[E-OLYMP](#) [10169. Ternary search](#) Find the minimum of a function f(x) = $x^2 + a * x + b$.

► Use ***ternary search***.

Let f(x) be a continuous concave function that has a local minimum on segment [a; b]. Ternary search allows you to find the minimum point.



Let $g = a + (b – a) / 3$, $h = a + 2 * (b – a) / 3$. Points g and h divide a segment [a; b] into three equal parts (hence the name – ***ternary search***). Compute f(g) and f(h).

- If f(g) ≤ f(h), then the minimum point of the function f lies on the segment [a; h], set $b = h$.
- If f(g) > f(h), then the minimum point lies on the segment [g; b], set $a = g$.

The search process continues while $|a – b| > \varepsilon$.

In the sample given a function f(x) = $x^2 – 2 * x – 35 = (x – 7) (x + 5)$. Its roots are $x_1 = -5$, $x_2 = 7$. The minimum of the function is achieved at

$$x = (x_1 + x_2) / 2 = (-5 + 7) / 2 = 1$$

The problem can be solved using a mathematical formula. It is known that the minimum of the function f(x) $= ax^2 + bx + c$ ($a > 0$) is achieved at $x = -b / (2a)$. Therefore, the minimum of the function f(x) = $x^2 + a * x + b$ is achieved at $x = - a / 2$.

Declare the constant epsilon $\varepsilon = 10^{-7}$.

```
#define EPS 0.0000001
```

Declare a function ***f***, the minimum of which we'll compute.

```
double f(double x)
{
  return x * x + a * x + b;
}
```

Function ***triple*** finds the minimum of the function f(x) on a segment [a; b].

```
double triple(double f(double x), double a, double b)
{
  double g, h;
  while (b - a > EPS)
  {
    g = a + (b - a) / 3;
```

```
    h = a + 2 * (b - a) / 3;
    if (f(g) <= f(h)) b = h; else a = g;
  }
  return (a + b) / 2;
}
```

The main part of the program. Read the input data. Start ternary search. We are looking for the minimum of the function f(x).

```
scanf("%lf %lf", &a, &b);
double x = triple(f, -100, 100);
```

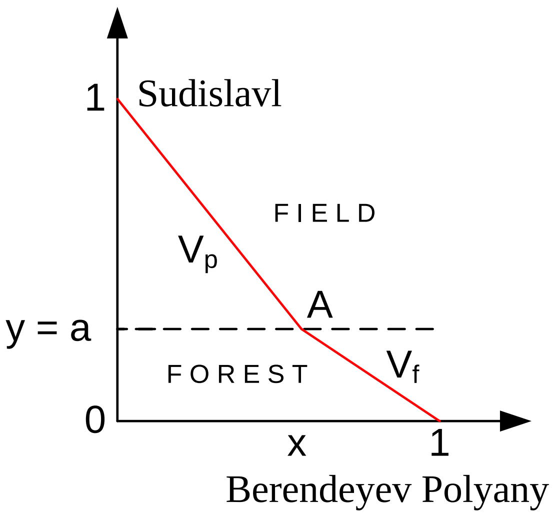
Print the value of the minimum x.

```
printf("%.2lf\n", x);
```

E-OLYMP **4419. Dense forest** To prevent the appearance of sanitary inspection in the camp, the LKSH administration dug the only road connecting the "Berendevy polyany" with Sudislavl, so now it is impossible to travel along it. However, the difficulties did not stop the inspection, for it remains only one possibility – to reach the camp on foot. As you know, Sudislavl is in the field, and "Berendevy polyany" are in the forest.

- Sudislavl is located at the point with coordinates (0, 1).
- "Berendevy polyany" are located at the point with coordinates (1, 0).
- The border between the field and forest is a horizontal line $y = a$, where a is some number ($0 \le a \le 1$).
- The speed of sanitary inspection through the field is V_p, the speed through the forest is V_f. Along the border it is allowed to move either by forest or by field.

The LKSH administration want to know how much time left to prepare for sanitary inspection visit. It asked you to find out at what point the sanitary inspection should enter into the forest to reach "Berendevy polyany" as fast as possible.



► Let the point A of intersection of the border field / forest have coordinates (x, a). The distance from Sudislavl to point A equals to $g_p(x) = \sqrt{x^2 + (1-a)^2}$. The distance from point A to Berendeyev Polyany is $g_f(x) = \sqrt{a^2 + (1-x)^2}$. The travel time from Sudislavl to Berendeyev Polyany is

$$g(x) = \frac{g_p(x)}{V_p} + \frac{g_f(x)}{V_f} = \frac{\sqrt{x^2 + (1-a)^2}}{V_p} + \frac{\sqrt{a^2 + (1-x)^2}}{V_f}$$

It remains to find the minimum of the function g(x) on the interval $x \in [0; 1]$. This can be done with ternary search.

Determine the distance from Sudislavl to Berendeyev Polyany as a function ***t*** depending on the abscissa *x* of the border between the forest and the field.

```
double t(double x)
{
  return sqrt(x*x + (1 - a)*(1 - a)) / vp +
            sqrt(a*a + (1 - x)*(1 - x)) / vf;
}
```

The main part of the program. Read the input data.

```
scanf("%lf %lf %lf",&vp,&vf,&a);
```

Run ternary search to find the minimum of the function $t(x)$ on the segment [0; 1].

```
Left = 0; Right = 1;
while(Right - Left >= EPS)
{
  f = Left + (Right - Left) / 3;
  g = Right - (Right - Left) / 3;
  if (t(f) < t(g)) Right = g; else Left = f;
}
```

Print the answer – the abscissa of the point where the sanitary inspection should enter the forest.

```
printf("%.9lf\n",Left);
```