Finding the Root

1 seconds, 256 megabytes

You are given a pure function from $\mathbb{R} \to \mathbb{R}$ which in C++ is:

```
double function(double);
```

Your goal is to find the root of the function, a.k.a the x where $f(x) = 0$

To simplify things, the function you are dealing with is **Polynomial Function**, it may have many roots but all roots are **Whole Number** and you are only required to find one of them.

Implement the following function:

```
int find_root(std::function<double(double)> f);
```

This function accepts function f as an argument and should returns x such that $f(x) = 0$

However, the function f take notes of how many times you called them and your score depends on how often you call the function.

Constraints

- All roots lie between -10^9 and 10^9 and degree of Polynomials **do not exceed** 10
- $\mathcal{O} = 10$
- You **cannot** call the function more than $2 * 10^9 + 1$ times (All the possibilities of roots)
- Your function will be called **multiple times**, this will not exceed 100 times. Your score for that test case is the **average score** of all function calls.

Scoring

Let Q be the number of times you have called the function f.

Condition	Ratio of your score to the full score of the test case
Answer is **Correct** and $Q \leq \mathcal{O}$	1
Answer is **Correct** and $Q > \mathcal{O}$	$\frac{2*\mathcal{O}}{\mathcal{O}+Q}$
Answer is **Incorrect**	0

Rounding will occur at test case level, if required, it will be **rounded down**.

Note: If the process get killed no matter by time or memory limit exceeded, or any runtime error, the score for that test case **will be** 0 no matter how many times have you answered the questions correct. You may need to plan *ejection strategy* for your function.

Subtasks

1. (1 Point) The answer of all tests are 69

2. (5 Points) The polynomial is linear

3. (9 Points) The polynomial is parabola

4. (17 Points) All roots lie between $-1\,000$ and $1\,000$

5. (10 Points) $\mathcal{O} = \infty$

6. (13 Points) $\mathcal{O} \in \{69\,420, 177\,013\}$

7. (14 Points) $\mathcal{O} = 1000$

8. (31 Points) No Additional Constraints

Examples

Definition of polynomial function f: *(Note: This does not reflect the real grader code.)*

```
auto f = [](double x){return x*x - 3*x + 2;};
```

Then, pass the function as argument to your function.

```
find_root(f);
```

Then, in your function, you call

```
f(0); // returns 2
f(1); // returns 0, is root
return 1; // correct answer
```

Your function should returns 1 or 2 which will make $f(x) = 0$

Sample Grader

First Line: T, F represents the number of tests and Full Score of the test case

For each test t_i has 2 lines

- **First Line of** t_i: R**,** O represents number of roots and O
- **Second Line of** t_i: $x_1, x_2, \cdots, x_R$ represents roots