Pogodi

Naš ludi grader.cpp je zamislio neki ( cijeli nenegativan broj) i vaš zadatak je da ga pogodite i to da ga pogodite u što manje pogađanja, jer se i to boduje.

Na početku programa, kad se grader.cpp spremi, on će pozvati vašu funkciju:

*unsigned long long int* ***pogodi*** *();*

Kroz nju ne dobijate nikakve informacije o broju, ali znate da je njegova veličina sigurno kao gore specificirano. Nakon toga vi u svojoj funkciji implementirate neku logiku, kojom ćete u što manje pogađanja pronaći broj. Funkcija koju pozivate za pogađanje je:

*int* ***ovajBroj*** *(unsigned long long int X);*

Ona će da broji koliko puta je pozvana i vi joj šaljete vaš upit za neki broj . Kako ona zna koji broj vi tražite, jer je ona u grader.cpp i vi je ne implementirate, nego samo imate pravo pozvati, ona će vam kao povratnu vrijednost tipa vratiti jedan od brojeva , koji znače sljedeće:

Nakon pogađanja, povratna vrijednost vaše funkcije ***pogodi*** mora biti broj koji je zamislio grader.

**Primjeri**

**Primjer br. 1** Neka je grader zamislio broj 3.

|  |  |
| --- | --- |
| *Poziv funkcije* | *Vraćena vrijednost* |
| pogodi()  ovajBroj(0)  ovajBroj(1)  ovajBroj(4)  ovajBroj(3) | 3  -1  -1  1  0 |

U ovom slučaju smo uspjeli pogoditi i trebalo nam je tačno četiri upita za odgovor, nakon čega je funkcija pogodi vratila vrijednost 3. Na serveru će se pogledati tačan broj upita i uporediti sa (super secret brojem) koji vi ne možete znati (možete samo pogađati) i ukoliko se ustanovi da ste imali poziva manje od , dobijate sve bodove testa, a ukoliko ne, onda dobijate samo dio bodova, prema nekoj (super secret formula).

**Ograničenja na resurse**

U slučajevima vrijednim bodova će vrijediti da zamišljeni broj manji od i da smo vam šapnuli SSB = 500 000. U ostalim će biti kao u tekstu.

Vremenska i memorijska ograničenja su dostupna na sistemu za ocjenjivanje.