Day 10

The Deref Debacle

The Story

The elves huddled in the workshop, their chatter buzzing with excitement.

"Have you seen the new ChatGPT model that just launched?" one elf asked, adjusting their tiny glasses.

"I did! But did you hear? It costs $200 a month!" another replied, shaking their head in disbelief.

The Story

"Two hundred dollars? That's a fortune!" chimed in a third, tossing a tinsel garland into a storage box.

The discussion heated up as they debated whether the advanced features were worth the hefty price tag, but their conversation was abruptly interrupted by a booming voice.

The Story

"I can't deref my snowballs! What's going on?" Santa's frustrated shout echoed through the workshop.

The elves froze mid-conversation, their heads snapping toward the jolly old man standing by his snowball workstation. He wore an expression of pure consternation as he jabbed a finger at the terminal screen displaying a stream of cryptic error messages.

The Story

One of the more tech-savvy elves hesitantly stepped forward. "Santa, you can just use `snowball.0` to access the first value in the tuple."

Santa's frown deepened, and he waved a hand dismissively. "Nonsense! I need to deref all the structs. Every single one!"

The Story

The elves weren't sure what Santa was coding, but they knew better than to argue with him when he was in debugging mode. Scrambling into action, they began reviewing his codebase, determined to find a solution before the big man lost his holiday cheer.

Your Mission

The code you wrote yesterday is a good start, but Santa needs easy access to the struct fields, instead of accessing them like this `snowball.0`, he wants to access them like this `*snowball`.

Your Mission

Here is what you need to do:

- Implement Deref for `SnowKg`.
- Implement Deref for `SnowLb`.
- Implement Deref for `Snowball`.
- Return the inner value of the tuple struct when dereferencing.

Hints

If you're stuck or need a starting point, here are some hints to help you along the way!

- Import the `Deref` trait from the `std::ops` module: `use std::ops::Deref;`.
- Implement the `Deref` trait for `SnowKg`, `SnowLb`, and `Snowball`. e.g., `impl Deref for SnowKg { ... }`.

Hints

- Define the associated type `Target` as the type of the inner value of the tuple struct. e.g.

```
type Target = f64;
```

- Implement the deref method for each struct. e.g.

```
fn deref(&self) -> &Self::Target {
    &self.0
}
```