**Principles of Programming Language**

**LAB 11 -RUST**

**Name: Harishkar B Roll no:CH.EN.U4CYS22019**

**Github Link:** **https://github.com/itsharishkar/Rust3**

--------------------------------------------------------------------------------------------------------------------------------------

**Question:**

1)Create a custom iterator named EvenNumbers that generates even numbers starting from 2 up to a given limit.

* Implement the Iterator trait for the struct.
* Use the next() method to return even numbers sequentially.
* Demonstrate the iterator in main() by printing the first 10 even numbers.

**OBJECTIVE:**

To Create a custom iterator named EvenNumbers that generates even numbers starting from 2 up to a given limit and Implement the Iterator trait for the struct.Use the next() method to return even numbers sequentially.Demonstrate the iterator in main() by printing the first 10 even numbers.

**Code:**

struct EvenNumbers {

    current: u32,

    limit: u32,

}

impl EvenNumbers {

    fn new(limit: u32) -> Self {

        Self { current: 2, limit }

    }

}

impl Iterator for EvenNumbers {

    type Item = u32;

    fn next(&mut self) -> Option<Self::Item> {

        if self.current > self.limit {

            return None;

        }

        self.current += 2;

        Some(self.current - 2)

    }

}

fn main() {

    let evens = EvenNumbers::new(20); // Limit set to 20 for demonstration

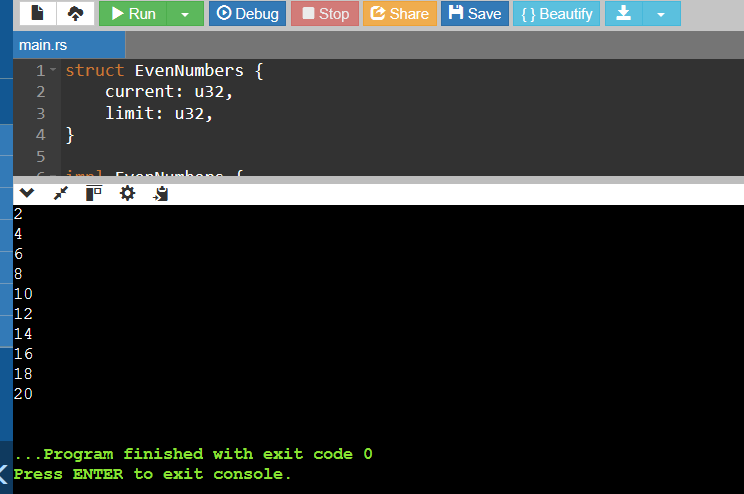
    for num in evens.take(10) {

        println!("{}", num);

    }

}

**Output:**



**Explanation:**

* This code creates a custom iterator EvenNumbers that generates even numbers starting from 2 up to a specified limit.
* Each call to next() returns the current even number and increments by 2, following Rust's Iterator trait pattern.
* The main() function demonstrates usage by creating an iterator with limit 20 and using take(10) to print the first 10 even numbers.

**CONCLUSION:**

This exercise strengthened the understanding of iterators in Rust by implementing a custom iterator to generate even numbers sequentially