

20CYS312 -PPL -Lab Exercise 7 – Programming with RUST

Name: Kiruthik Pranav P V

Roll No: CH.EN.U4CYS22026

Date:21/02/25

Github Link:<https://github.com/kpsan12/Haskell-Programming/tree/main/lab7>

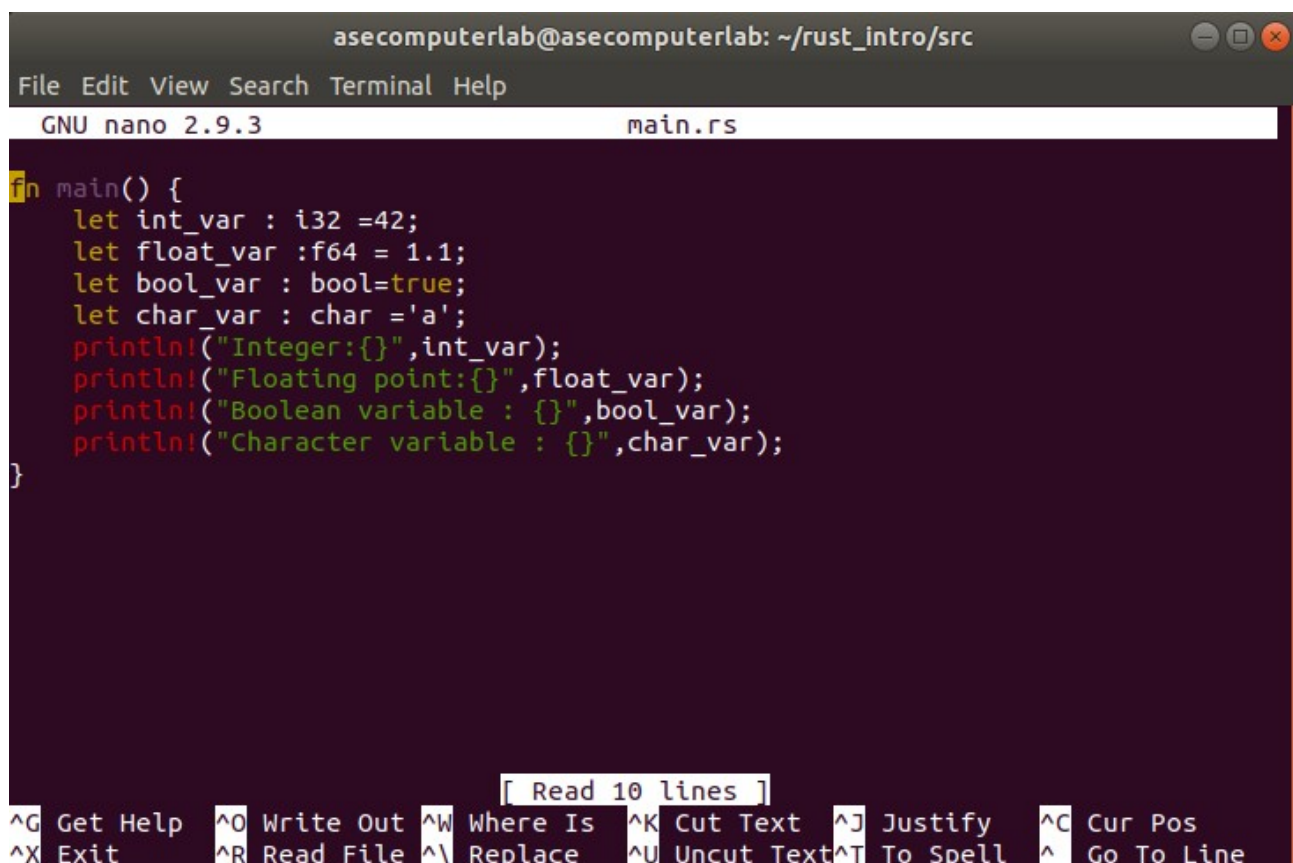
Task 1: Data Types and Variables

Qn1.Declare variables of the following types : integer,floating point,boolean and character.Print the value of each variable.

Objective:

To declare variables of the following types : integer,floating point,boolean and character and to print the value of each variable.

Program Code:



The screenshot shows a terminal window titled 'asecomputerlab@asecomputerlab: ~/rust_intro/src'. The window contains a nano editor editing a file named 'main.rs'. The code in the editor is as follows:

```
in main() {  
    let int_var : i32 =42;  
    let float_var :f64 = 1.1;  
    let bool_var : bool=true;  
    let char_var : char ='a';  
    println!("Integer:{}",int_var);  
    println!("Floating point:{}",float_var);  
    println!("Boolean variable : {}",bool_var);  
    println!("Character variable : {}",char_var);  
}
```

At the bottom of the terminal, there is a status bar with various keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^C Cur Pos, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell, and ^_ Go To Line. A prompt '[Read 10 lines]' is also visible.

Explanation of the Code:

- Declares variables of integer (i32), floating-point (f64), boolean (bool), and character (char) types.
- Prints their value using Println!

Input/Output Example /Output Screenshot:

```
asecomputerlab@asecomputerlab:~/rust_intro/src$ cargo run
  Compiling rust_intro v0.1.0 (/home/asecomputerlab/rust_intro)
  Finished dev [unoptimized + debuginfo] target(s) in 0.17s
  Running `/home/asecomputerlab/rust_intro/target/debug/rust_intro`
Integer:42
Floating point:1.1
Boolean variable : true
Character variable : a
```

Conclusion:

- Successfully learnt how to declare variables and print them

Task 2: Simple Arithmetic Operations

Qn2.Declare two integer variables and perform the following operations:

- Addition
- Subtraction
- Multiplication
- Division
- Modulo

Qn3.Print the Result of each operation

Objective:

- To declare two integer variables and perform arithmetic operations and print the result of each

Program Code:

```
asecomputerlab@asecomputerlab: ~/qn2
File Edit View Search Terminal Help
GNU nano 2.9.3 main.rs

fn main(){
    let a : i32 = 7;
    let b : i32 = 10;
    println!("Addition:{}",a+b);
    println!("subtraction:{}",a-b);
    println!("Multiplication:{}",a*b);
    println!("Division:{}",a/b);
    println!("Modulus:{}",a%b);
}

[ Read 9 lines ]
^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Explanation of the Code:

- Declared two variables of int and assigned values
- Printed the value of arithmetic operations

Output Screenshots:

```
asecomputerlab@asecomputerlab:~/qn2$ nano main.rs
asecomputerlab@asecomputerlab:~/qn2$ rustc main.rs
asecomputerlab@asecomputerlab:~/qn2$ ./main
Addition:17
subtraction:-3
Multiplication:70
Division:0
Modulus:7
asecomputerlab@asecomputerlab:~/qn2$ nano main.rs
asecomputerlab@asecomputerlab:~/qn2$
```

Conclusion:

- Successfully learnt how to do arithmetic operations and the ways to print them

Task 3: If-Else Decision Making

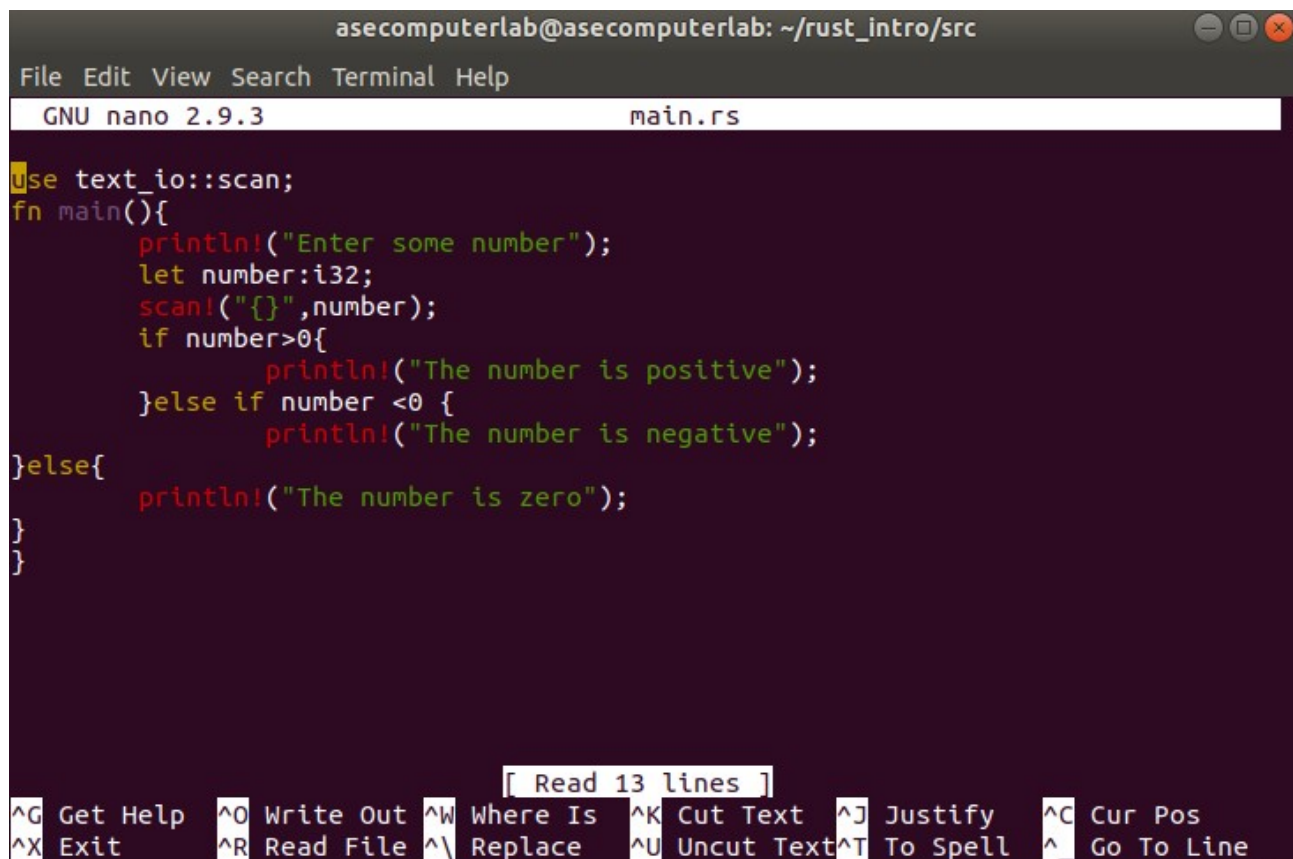
Qn1. Write a program that:

- Takes a number as input.
- Checks whether the number is positive, negative, or zero using an if-else statement.
- Print a message based on the result.

Objective:

- To check if a number is positive or negative or zero

Program Code:



```
asecomputerlab@asecomputerlab: ~/rust_intro/src
File Edit View Search Terminal Help
GNU nano 2.9.3 main.rs

use text_io::scan;
fn main(){
    println!("Enter some number");
    let number:i32;
    scan!("{}",number);
    if number>0{
        println!("The number is positive");
    }else if number <0 {
        println!("The number is negative");
    }else{
        println!("The number is zero");
    }
}
```

[Read 13 lines]

^G Get Help	^O Write Out	^W Where Is	^K Cut Text	^J Justify	^C Cur Pos
^X Exit	^R Read File	^_ Replace	^U Uncut Text	^T To Spell	^_ Go To Line

Explanation of the Code:

- First we are getting an input
- We are checking if the number is positive and negative or zero and printing it accordingly

Input/Output Examples:

INPUT: 1 OUTPUT: The number is positive
INPUT:-1 OUTPUT: The number is negative
INPUT:0 OUTPUT: The number is zero

Output Screenshots:

```

    Running `target/debug/rust_intro`
Enter some number
20
The number is positive
asecomputerlab@asecomputerlab:~/rust_intro$ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.00s
    Running `target/debug/rust_intro`
Enter some number
-1
The number is negative
asecomputerlab@asecomputerlab:~/rust_intro$ cargo run
    Finished dev [unoptimized + debuginfo] target(s) in 0.00s
    Running `target/debug/rust_intro`
Enter some number
0
The number is zero
asecomputerlab@asecomputerlab:~/rust_intro$ cargo run

```

Task 4: Checking for Even or Odd

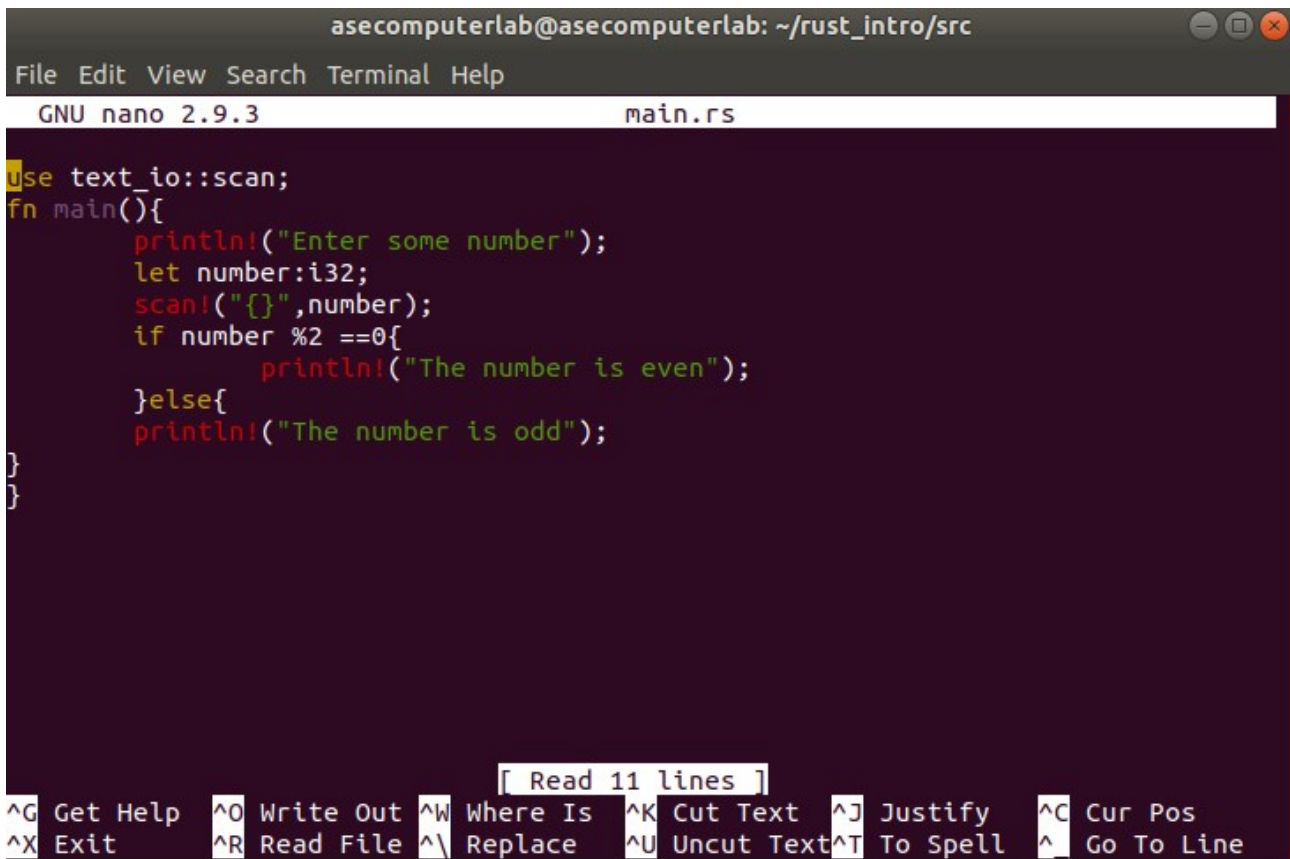
Qn1. Write a program that:

- Takes an integer as input.
- Uses an if-else statement to check if the number is even or odd.
- Print "Even" if the number is even and "Odd" if the number is odd.

Objective of the Code:

- To get a number as input and check if it is even or odd and print accordingly

Program Code:



The screenshot shows a terminal window with the title bar "asecomputerlab@asecomputerlab: ~/rust_intro/src". The window contains the GNU nano 2.9.3 editor editing the file "main.rs". The code in the editor is as follows:

```
use text_io::scan;
fn main(){
    println!("Enter some number");
    let number:i32;
    scan!("{}",number);
    if number %2 ==0{
        println!("The number is even");
    }else{
        println!("The number is odd");
    }
}
```

At the bottom of the terminal, there is a status bar with the text "[Read 11 lines]" and a table of keyboard shortcuts:

^G	Get Help	^O	Write Out	^W	Where Is	^K	Cut Text	^J	Justify	^C	Cur Pos
^X	Exit	^R	Read File	^_\	Replace	^U	Uncut Text	^T	To Spell	^_	Go To Line

Explanation of the Code:

- Got input from user
- checked if it is odd or even using modulus functions
- Printed Accordingly

Input/Output Examples:

Input: 2 Output:The number is Even

Input:1 Output:The number is Odd

Output Screenshots:

```

asecomputerlab@asecomputerlab:~/rust_intro$ cargo run
   Compiling rust_intro v0.1.0 (/home/asecomputerlab/rust_intro)
   Finished dev [unoptimized + debuginfo] target(s) in 0.20s
   Running `target/debug/rust_intro`
Enter some number
2
The number is even
asecomputerlab@asecomputerlab:~/rust_intro$ cargo run
   Finished dev [unoptimized + debuginfo] target(s) in 0.00s
   Running `target/debug/rust_intro`
Enter some number
3
The number is odd

```

Conclusion:

- Implemented odd or even checker

Task 5: Using a Loop to Print Numbers

Qn1. Write a program that uses a for loop to print the even numbers from the range 1 to 20.

Objective

- To Write a program that uses a for loop to print the even numbers from the range 1 to 20.

Program Code:



```

asecomputerlab@asecomputerlab: ~/rust_intro/src
File Edit View Search Terminal Help
GNU nano 2.9.3 main.rs

use std::io;

fn main() {
    println!("Even numbers from 1 to 20:");

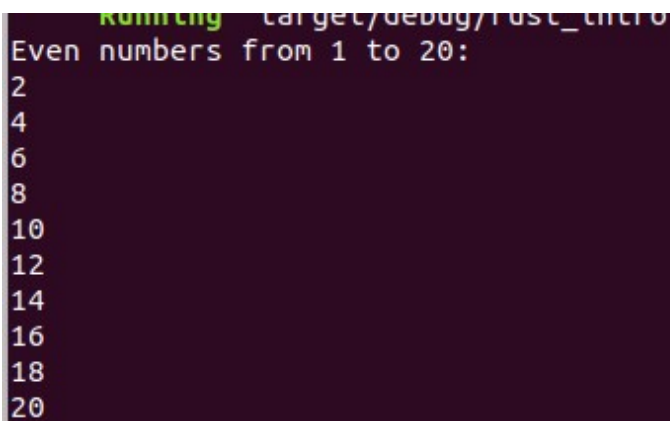
    for number in 1..=20 {
        if number % 2 == 0 {
            println!("{}", number);
        }
    }
}

```

Explanation of the Code:

- Used even checking
- Printed numbers which are even using loop

Input/Output Examples / Output Screenshots:



```
Running target/debug/rust_intro
Even numbers from 1 to 20:
2
4
6
8
10
12
14
16
18
20
```

Conclusion:

- Implemented a for loop to print even numbers

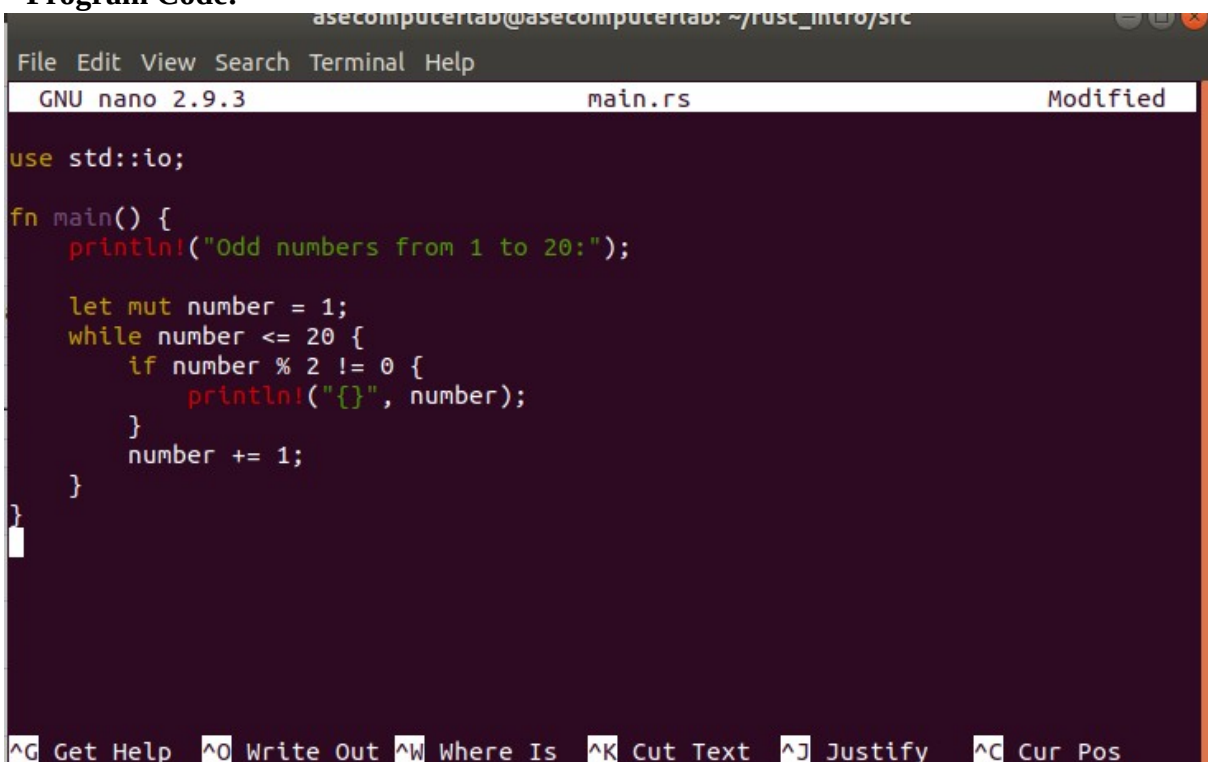
Task 6: While Loop Example

Qn1. Write a program that uses a while loop to print odd numbers from the range 1 to 20.

Objective of the Code:

- To Write a program that uses a while loop to print odd numbers from the range 1 to 20.

Program Code:



```
asecomputerlab@asecomputerlab: ~/rust_intro/src
File Edit View Search Terminal Help
GNU nano 2.9.3 main.rs Modified

use std::io;

fn main() {
    println!("Odd numbers from 1 to 20:");

    let mut number = 1;
    while number <= 20 {
        if number % 2 != 0 {
            println!("{}", number);
        }
        number += 1;
    }
}
```


Explanation of the Code:

- Created a number which is mutable using the `let mut number=1;`
- Used a while loop for numbers upto 20
- Printing the odd numbers alone

Input/Output Examples /Screenshots

```
warning: `rust_intro` (bin "rust_intro") generated 1 warning
Finished dev [unoptimized + debuginfo] target(s) in 0.17s
Running `target/debug/rust_intro`
Odd numbers from 1 to 20:
1
3
5
7
9
11
13
15
17
19
```

Conclusion:

- In a while loop printed all odd numbers upto 20

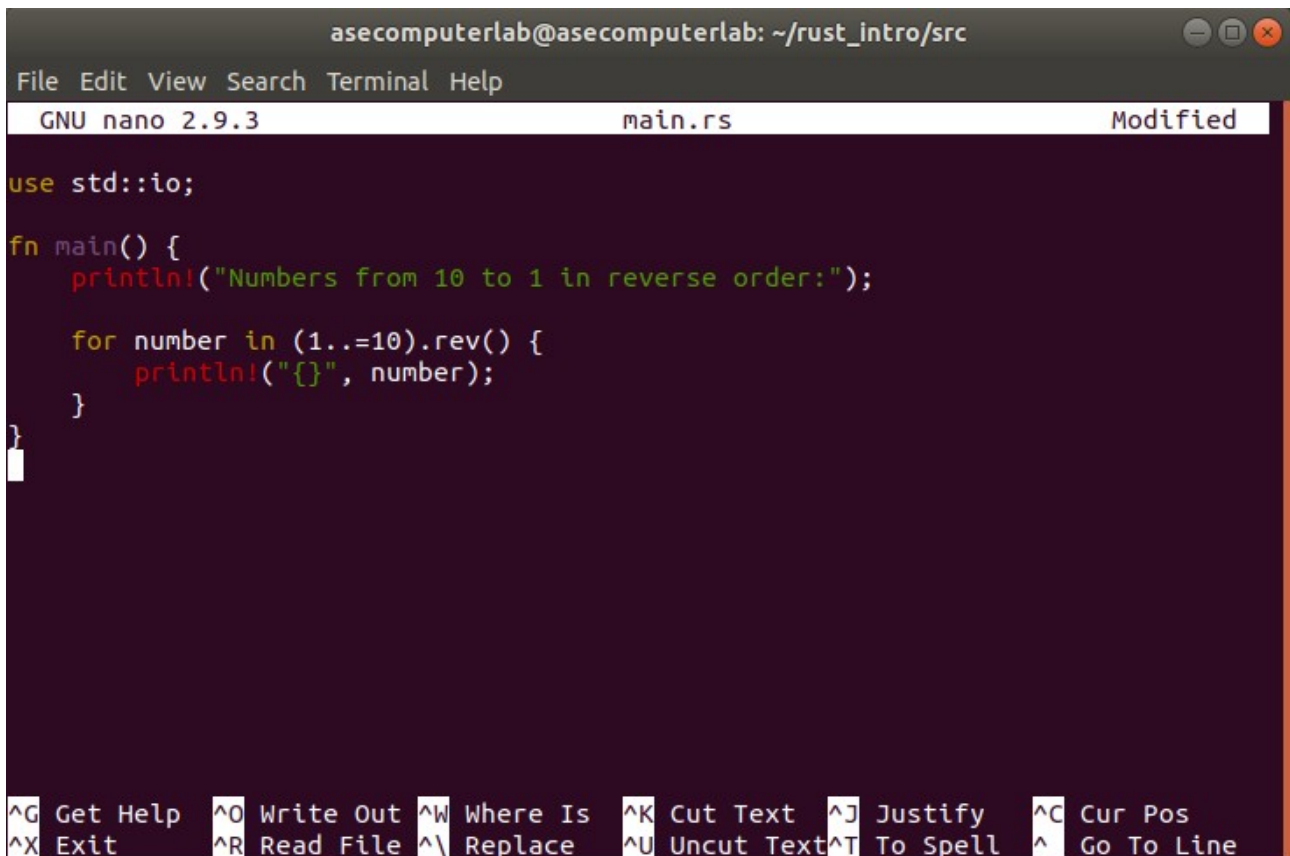
Task 7: Using a For Loop with a Range

Qn1. Write a program that uses a for loop to print the numbers from 10 to 1 in reverse order (10, 9, 8, ..., 1).

Objective of the Code:

- To write a program that uses a for loop to print the numbers from 10 to 1 in reverse order

Program Code:



```
asecomputerlab@asecomputerlab: ~/rust_intro/src
File Edit View Search Terminal Help
GNU nano 2.9.3 main.rs Modified

use std::io;

fn main() {
    println!("Numbers from 10 to 1 in reverse order:");

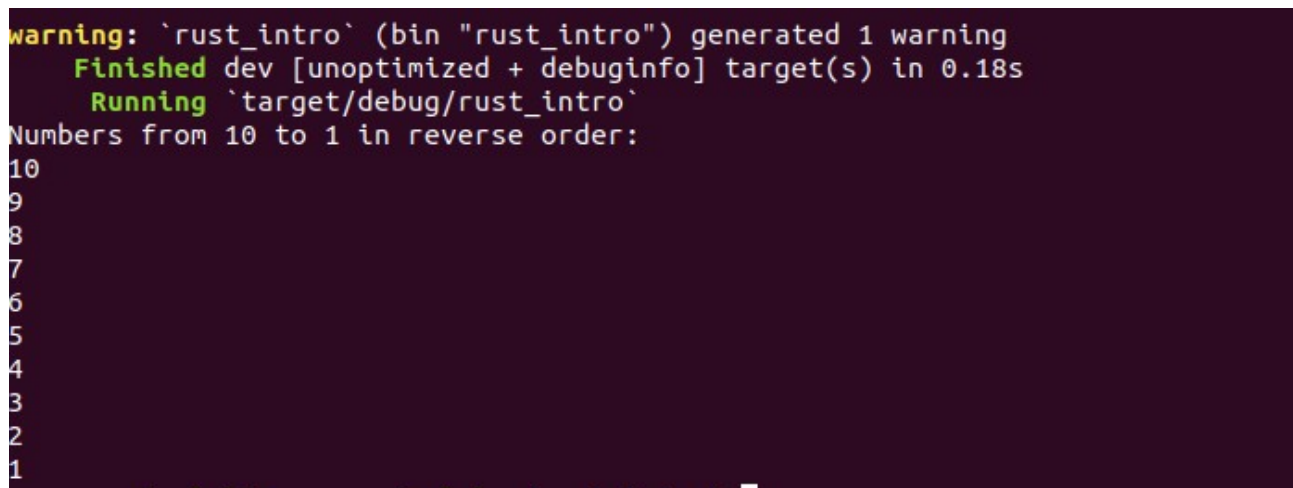
    for number in (1..=10).rev() {
        println!("{}", number);
    }
}

^G Get Help ^O Write Out ^W Where Is ^K Cut Text ^J Justify ^C Cur Pos
^X Exit ^R Read File ^\ Replace ^U Uncut Text ^T To Spell ^_ Go To Line
```

Explanation of the Code:

- Created a for loop for numbers from 1 to 10 and used the function `rev()` to reverse it
- So printed the values from 10 to 1

Input/Output Examples/Screenshots:



```
warning: `rust_intro` (bin "rust_intro") generated 1 warning
Finished dev [unoptimized + debuginfo] target(s) in 0.18s
Running `target/debug/rust_intro`
Numbers from 10 to 1 in reverse order:
10
9
8
7
6
5
4
3
2
1
```

Conclusion:

- Successfully printed the values of 10 to 1 in reverse order using for loop