

Add Two Numbers

Namaste 🙏, Today in this program Im gonna show you

how to write a program of adding two numbers in Rust Programming language .

Program to add two numbers in Rust .

`use` keyword is used to import certain packages as we do in every programming languages

`print!` and `println!` are methods which are used to print something on screen .

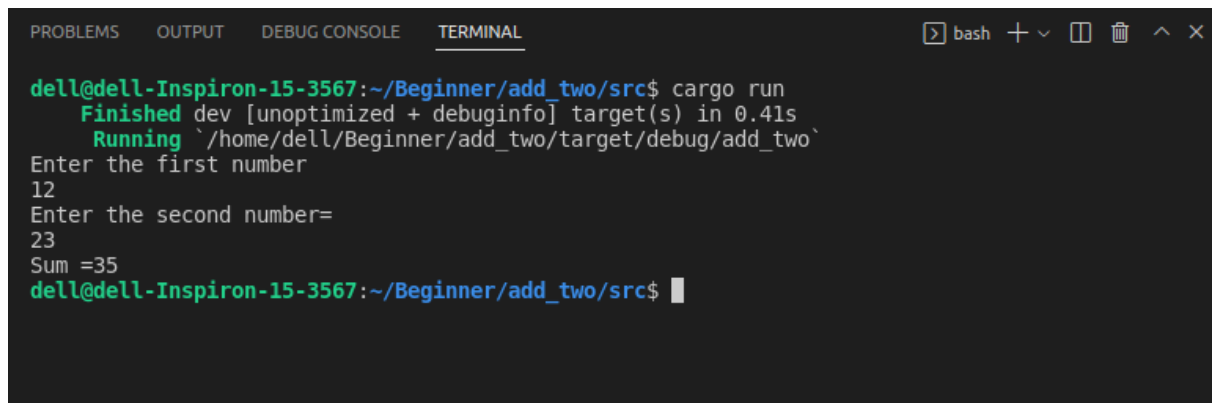
`print!()`- is used to print the values arguments on same line.

Whereas

`println!()`- shifts the arguments or the values to new line

First run the program using `cargo run` .

See what happens ?



```
PROBLEMS  OUTPUT  DEBUG CONSOLE  TERMINAL  bash + v [] [x] ^ x

dell@dell-Inspiron-15-3567:~/Beginner/add_two/src$ cargo run
   Finished dev [unoptimized + debuginfo] target(s) in 0.41s
   Running `/home/dell/Beginner/add_two/target/debug/add_two`
Enter the first number
12
Enter the second number=
23
Sum =35
dell@dell-Inspiron-15-3567:~/Beginner/add_two/src$
```

Now Lets see how this Program works 😊

use std::io; // Here we are importing
standard io library / package
to perform standard input / output
fn main { // fn keyword is used
to declare a method main
println!("Enter the value of a=");

// This actually prints the
arguments to new line.

let mut a = String::new();

// Creates a mutable
variable a and accepts String
input. {new} is actually
dynamic memory allocation
keyword. Hence it creates
a memory for String input.

```
io::Stdin().read_line(&mut  
a).expect("Not a number");
```

// Reads a mutable variable
a and expect is a method
which is a part of error
handling in Rust.

=
/* Now what is the last
thing left with us? */

Parsing the variable to another
datatype. like i32, floating
point double etc.

and displaying the result

Boom < ☹



Here I'll be taking 'extra variable to store' parsed output.

Let's take another name 'v'

✓ ~~Integer~~ {Integer Parsed}

```
let v: i32 = a.trim().parse()
```

```
ok().expect("Not a number");
```

// Printing the result

```
println!("Value Entered of a = ", v);  
}
```

```
    """Listen here!"""  
    -----
```

Format specifier is a must While printing the result .
You must use {} after that otherwise the compiler will give you the warning and

These type of errors are known as `Syntax Error` .

Your task is to

Write a program to input two numbers and multiply the result .Print the multiplied Value using print! method

Bye ! Catch you up next time