Borrowing

When a function transfers its control over a variable/value to another function temporarily, for a while, it is called borrowing. This is achieved by passing a reference to the variable (& var_name) rather than passing the variable/value itself to the function.

```
fn vec_fun(x: &Vec<i32>) {
    println!("{:?}", x);
}


fn main() {
    let a = vec![1, 2, 3, 4];
    vec_fun(&a);
    println!("{:?}", a);

}
```

It is very inconvenient to pass the ownership of a variable to another function and then return the ownership. Rust supports a concept, borrowing, where the ownership of a value is transferred temporarily to an entity and then returned to the original owner entity.