Homework # 4

01286120 Elementary Systems Programming

Software Engineering Program

Faculty of Engineering, KMITL

By

66011149 Phatthadon Sornplang

1. Write a function to convert temperatures

1.1) From the Fahrenheit to Celsius formula °C = (5/9)(°F - 32) , write a function fahr_to_cel_v to convert temperatures in °F to °C using ordinary loop. Add a unit test for the function to cover use cases.

(Code in attached file)

1.2) Rewrite the function fahr_to_cel_v by using recursion instead of ordinary loop

(Code in attached file)

```
phatt@Macbook_Pro MINGW64 ~/OneDrive/Desktop/Code Files/Rust/Lab_4/HW4/temp/src (master)
$ cargo test
    Finished test [unoptimized + debuginfo] target(s) in 0.00s
     Running unittests src\main.rs (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\temp\target\debug\deps\temp-172f1ce4cac63ed6.exe)

running 2 tests
test test_celcius ... ok
test test_celcius_recur ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

2. Write a function to calculate grades from scores

2.1) Using the same criteria as in exercise 1.1) in homework #3, write the function make_grades to create a list of grades from a list of scores using ordinary loop. Add a unit test for the function to cover use cases.

(Code in attached file)

2.2) Rewrite the function make_grades by using recursion instead of ordinary loop.

(Code in attached file)

```
phatt@Macbook_Pro MINGW64 ~/OneDrive/Desktop/Code Files/Rust/Lab_4/HW4/grade/src (master)
$ cargo test
   Compiling grade v0.1.0 (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\grade)
    Finished test [unoptimized + debuginfo] target(s) in 0.38s
     Running unittests src\main.rs (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\grade\target\debug\deps\grade-92190405648ed478.ex
e)

running 2 tests
test test_grade_list ... ok
test test_grade_list_recur ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```

3. Write a function to create arrow patterns with varying sizes

3.1) The function make_arrow1 should create arrow pattern as a string like shown in exercise3.1) in homework #3 using ordinary loop. Add a unit test for the function to cover use cases.

(Code in attached file)

3.2) The function make_arrow2 should create arrow pattern as a string like shown in exercise 3.2) in homework #3 using ordinary loop. Add a unit test for the function to cover use cases.

(Code in attached file)

```
    Running tests\cli.rs (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\arrow\target\debug\deps\cli-a91ba25bd8184c03.exe)

running 2 tests
test test_arrow_loop1 ... ok
test test_arrow_loop2 ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.94s
```

3.3) Rewrite the function make_arrow1 and make_arrow2 by using recursion instead of ordinary loop.

(Code in attached file)

```
phatt@Macbook_Pro MINGW64 ~/OneDrive/Desktop/Code Files/Rust/Lab_4/HW4/arrow_recur/src
$ cargo test
   Compiling arrow_recur v0.1.0 (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\arrow_recur)
    Finished test [unoptimized + debuginfo] target(s) in 0.26s
     Running unittests src\main.rs (C:\Users\phatt\OneDrive\Desktop\Code Files\Rust\Lab_4\HW4\arrow_recur\target\debug\deps\arrow_recur-8ebf652
56ab45eb8.exe)

running 2 tests
test test_arrow_recur1 ... ok
test test_arrow_recur2 ... ok

test result: ok. 2 passed; 0 failed; 0 ignored; 0 measured; 0 filtered out; finished in 0.00s
```