

Assignment Four

Ada Programming

Quin'darius Lyles-Woods

qlyleswo@students.kennesaw.edu

Concepts of Programming Languages

Professor Jose Garrido

Section W01

4308



Bachelors of Computer Science

Kennesaw State University

1100 South Marietta Pkwy SE

Marietta, GA 30060

November 8, 2021

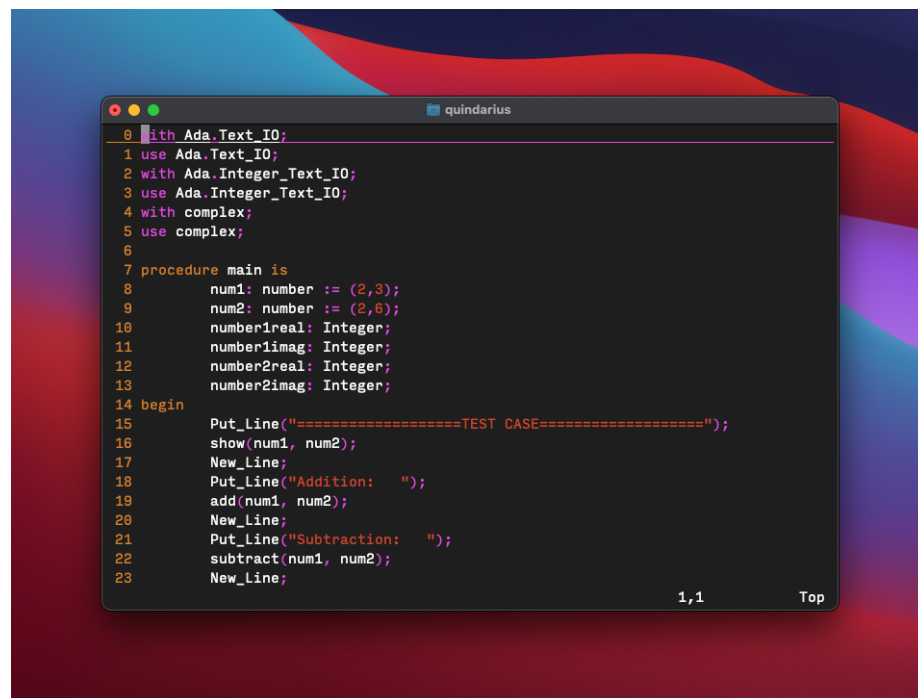
Task

The development of an Ada program for exploring the properties of complex numbers.

Assignment Goals

- Author a complex numbers package in Ada.
- Write a report on the development of such a package.
- Have the following operations; addition, subtraction, multiplication, and division.

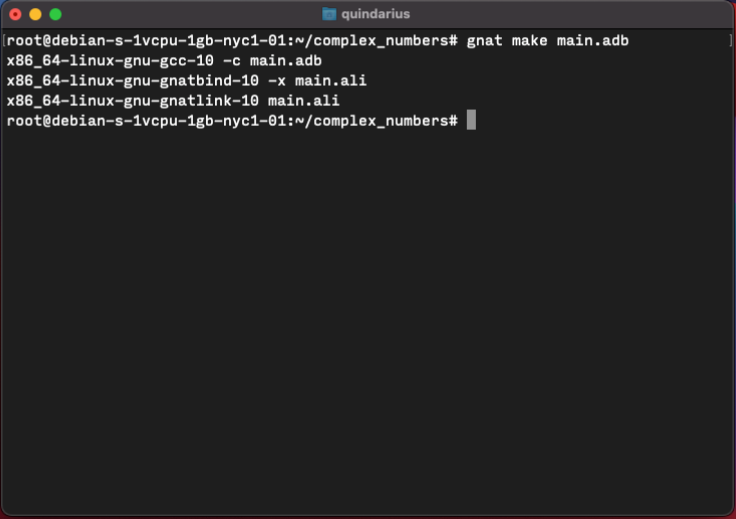
Source Code



```
0 with Ada.Text_IO;
1 use Ada.Text_IO;
2 with Ada.Integer_Text_IO;
3 use Ada.Integer_Text_IO;
4 with complex;
5 use complex;
6
7 procedure main is
8     num1: number := (2,3);
9     num2: number := (2,6);
10    number1real: Integer;
11    number1imag: Integer;
12    number2real: Integer;
13    number2imag: Integer;
14 begin
15     Put_Line("=====TEST CASE=====");
16     show(num1, num2);
17     New_Line;
18     Put_Line("Addition:  ");
19     add(num1, num2);
20     New_Line;
21     Put_Line("Subtraction:  ");
22     subtract(num1, num2);
23     New_Line;
```

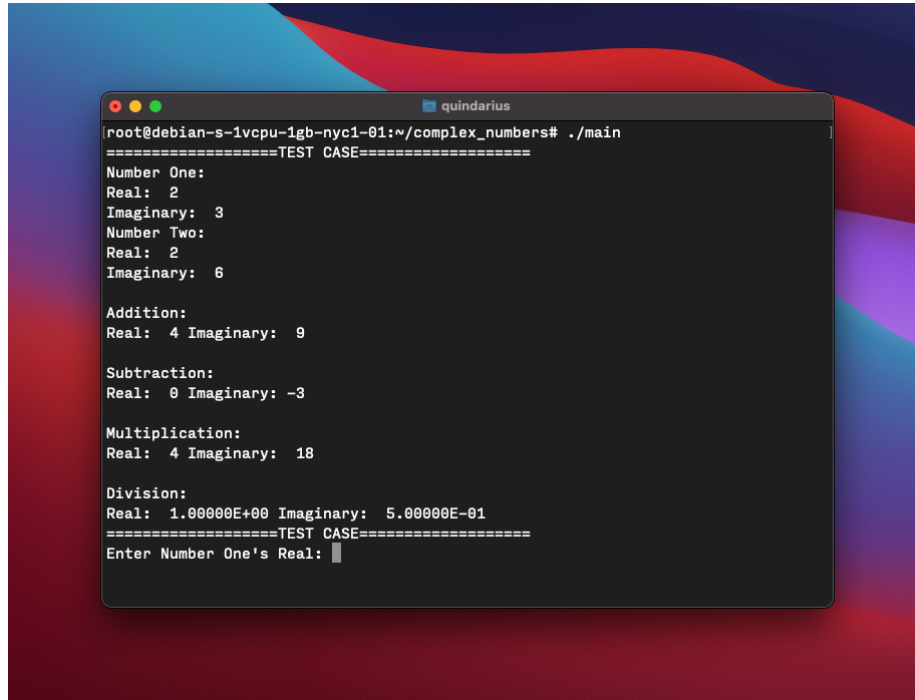
1,1 Top

Compile Output

A terminal window titled "quindarius" is shown against a background of abstract red, blue, and purple shapes. The terminal displays the output of a "gmake" command. The prompt is "root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers#". The output consists of three lines: "x86_64-linux-gnu-gcc-10 -c main.adb", "x86_64-linux-gnu-gnatbind-10 -x main.ali", and "x86_64-linux-gnu-gnatlink-10 main.ali". The prompt is repeated at the end of the output.

```
root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers# gmake main.adb
x86_64-linux-gnu-gcc-10 -c main.adb
x86_64-linux-gnu-gnatbind-10 -x main.ali
x86_64-linux-gnu-gnatlink-10 main.ali
root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers#
```

Test Output

A terminal window titled 'quindarius' is shown against a background with red and blue wavy patterns. The terminal displays the output of a program that performs operations on complex numbers. The output includes prompts for 'Number One' and 'Number Two', followed by calculations for addition, subtraction, multiplication, and division. The division result is shown in scientific notation. The terminal ends with a prompt to 'Enter Number One's Real:'.

```
root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers# ./main
=====TEST CASE=====
Number One:
Real: 2
Imaginary: 3
Number Two:
Real: 2
Imaginary: 6

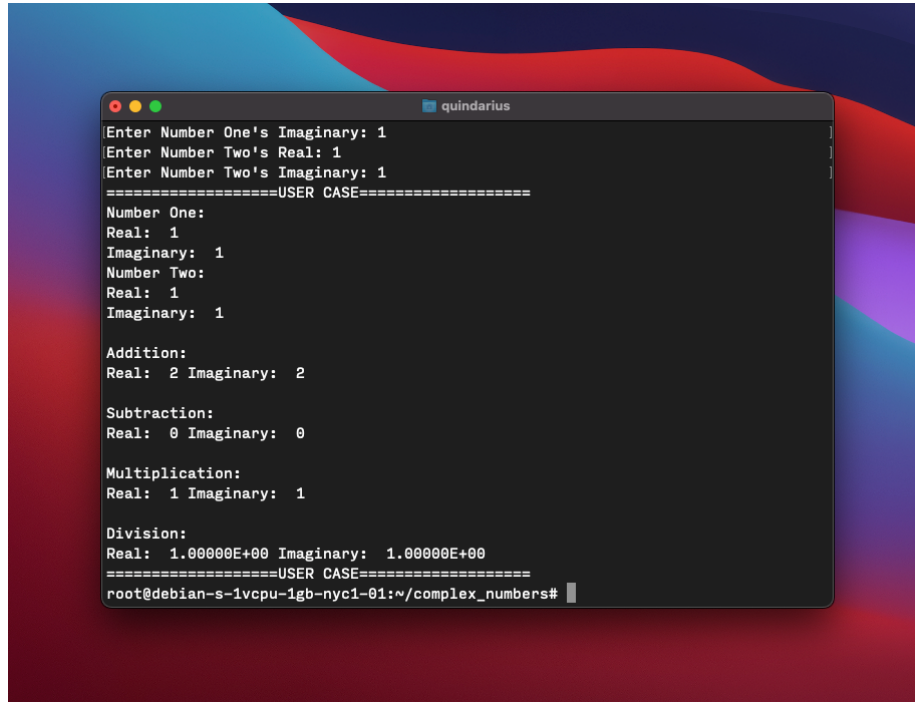
Addition:
Real: 4 Imaginary: 9

Subtraction:
Real: 0 Imaginary: -3

Multiplication:
Real: 4 Imaginary: 18

Division:
Real: 1.00000E+00 Imaginary: 5.00000E-01
=====TEST CASE=====
Enter Number One's Real: 
```

User Input Output

A screenshot of a terminal window titled 'quindarius' with a dark background and light-colored text. The terminal shows a series of prompts and user inputs for a program that handles complex numbers. The prompts include 'Enter Number One's Imaginary:', 'Enter Number Two's Real:', and 'Enter Number Two's Imaginary:'. The user has entered '1' for all three. The program then displays a separator line '=====USER CASE====='. It shows the input for 'Number One' (Real: 1, Imaginary: 1) and 'Number Two' (Real: 1, Imaginary: 1). The results of operations are shown: 'Addition: Real: 2 Imaginary: 2', 'Subtraction: Real: 0 Imaginary: 0', 'Multiplication: Real: 1 Imaginary: 1', and 'Division: Real: 1.00000E+00 Imaginary: 1.00000E+00'. Another separator line '=====USER CASE=====' follows. The prompt 'root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers#' is visible at the bottom.

```
quindarius
Enter Number One's Imaginary: 1
Enter Number Two's Real: 1
Enter Number Two's Imaginary: 1
=====USER CASE=====
Number One:
Real: 1
Imaginary: 1
Number Two:
Real: 1
Imaginary: 1

Addition:
Real: 2 Imaginary: 2

Subtraction:
Real: 0 Imaginary: 0

Multiplication:
Real: 1 Imaginary: 1

Division:
Real: 1.00000E+00 Imaginary: 1.00000E+00
=====USER CASE=====
root@debian-s-1vcpu-1gb-nyc1-01:~/complex_numbers#
```

0.1 Summary

After finishing this assignment I have a better understanding on how Ada works. Defiantly not my favorite experience but I loved how it resembled c with adding external code. Another notation about the language that I might what to follow in my other programs is the stark distinction of where declaration and operational code was within. They don't even allow the two to mix and I think that is a plug in the long run for the language design. Luckily for me, that more of a style aspect in other language and I will be able to take that style and put it almost anywhere.

Lastly, this assignment allowed me to review complex numbers. They are so simply but for some reason have been seen as almost mystical to me. So I appreciated that aspect of the assignment as well.