Tutor: Marzieh Farahani

Principles of ProgrammingAssignment 1

Word Count: 1000

Student ID: 1472703

Contents

[Introduction 2](#_Toc31144224)

[Part A: Exercise 1 3](#_Toc31144225)

[Design, source code and testing 3](#_Toc31144226)

[Part A: Exercise 2 5](#_Toc31144227)

[Design, source code and testing 5](#_Toc31144228)

[Part A: Exercise 3 7](#_Toc31144229)

[Design, source code and testing 7](#_Toc31144230)

[Part A: Exercise 4 9](#_Toc31144231)

[Design, source code and testing 9](#_Toc31144232)

[Part A: Exercise 5 11](#_Toc31144233)

[Design, source code and testing 11](#_Toc31144234)

[Part A: Exercise 6 13](#_Toc31144235)

[Design, source code and testing 13](#_Toc31144236)

[Part A: Exercise 7 15](#_Toc31144237)

[Design, source code and testing 15](#_Toc31144238)

[Part A: Exercise 8 17](#_Toc31144239)

[Design, source code and testing 17](#_Toc31144240)

[Part B: Exercise 1 19](#_Toc31144241)

[Design, source code and testing 19](#_Toc31144242)

[Part B: Exercise 2 21](#_Toc31144243)

[Design, source code and testing 21](#_Toc31144244)

[Part B: Exercise 3 23](#_Toc31144245)

[Design, source code and testing 23](#_Toc31144246)

[Part C: Exercise 1 25](#_Toc31144247)

[Design, source code and testing 25](#_Toc31144248)

[Part C: Exercise 2 27](#_Toc31144249)

[Design, source code and testing 27](#_Toc31144250)

[Part C: Exercise 3 29](#_Toc31144251)

[Design, source code and testing 29](#_Toc31144252)

[Part C: Exercise 4 31](#_Toc31144253)

[Design, source code and testing 31](#_Toc31144254)

[Part C: Exercise 5 33](#_Toc31144255)

[Design, code source and testing 33](#_Toc31144256)

[Part D 35](#_Toc31144257)

[Design, source code and testing 35](#_Toc31144258)

[Reflection 38](#_Toc31144259)

[Conclusion 40](#_Toc31144260)

[References 41](#_Toc31144261)

[Bibliography 41](#_Toc31144262)

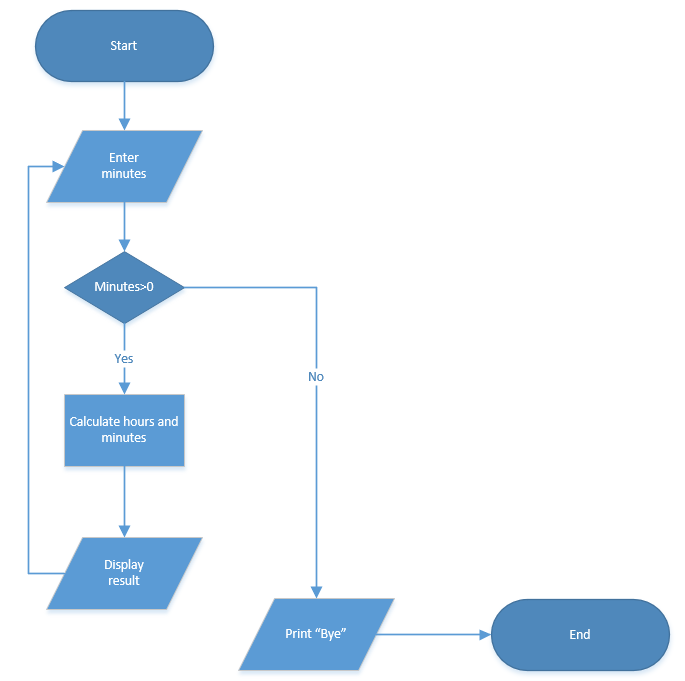
[Appendix A 42](#_Toc31144263)

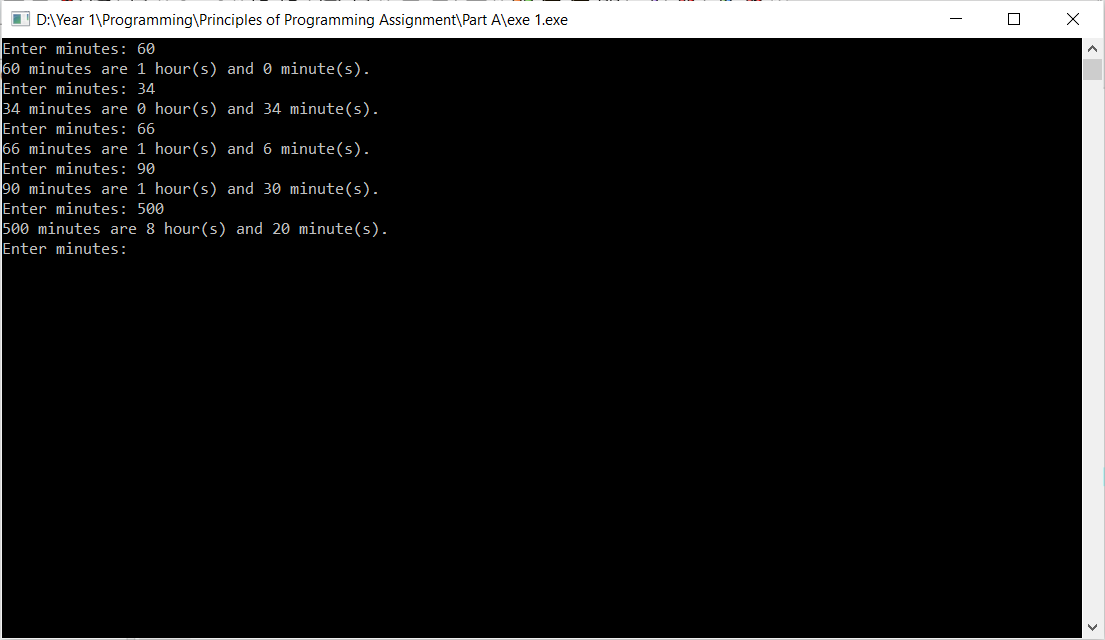
# Introduction

This report will showcase the various number of exercises that were attempted by myself. Showing a basic design plan of the code, the code itself and the result. A reflection of what went well and what can be improved will also be discussed. As well as changes and lengths that were made when creating these programs in the reflection. Finally, there will also be a conclusion of concluding the overall project and my experience with it.

# Part A: Exercise 1

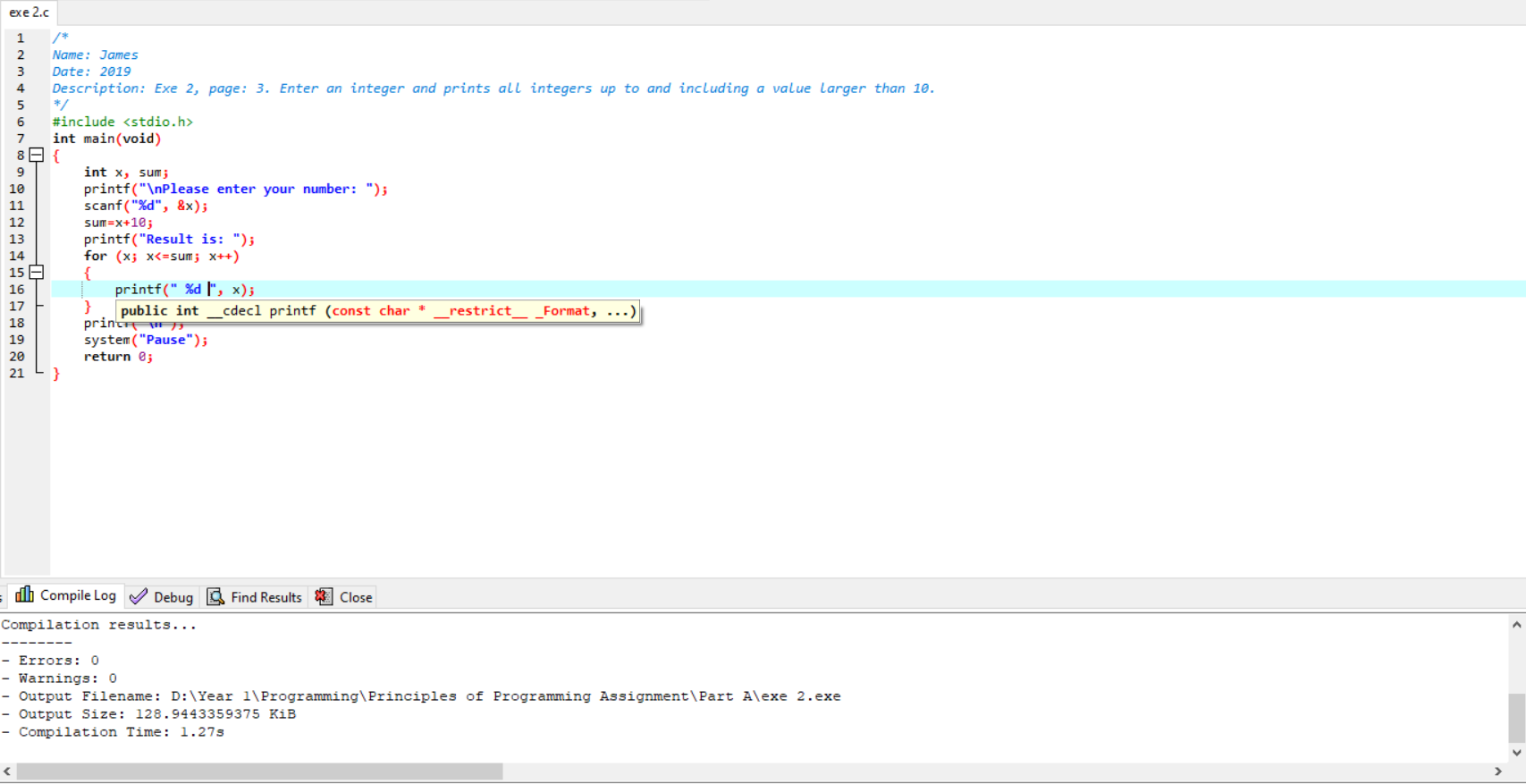
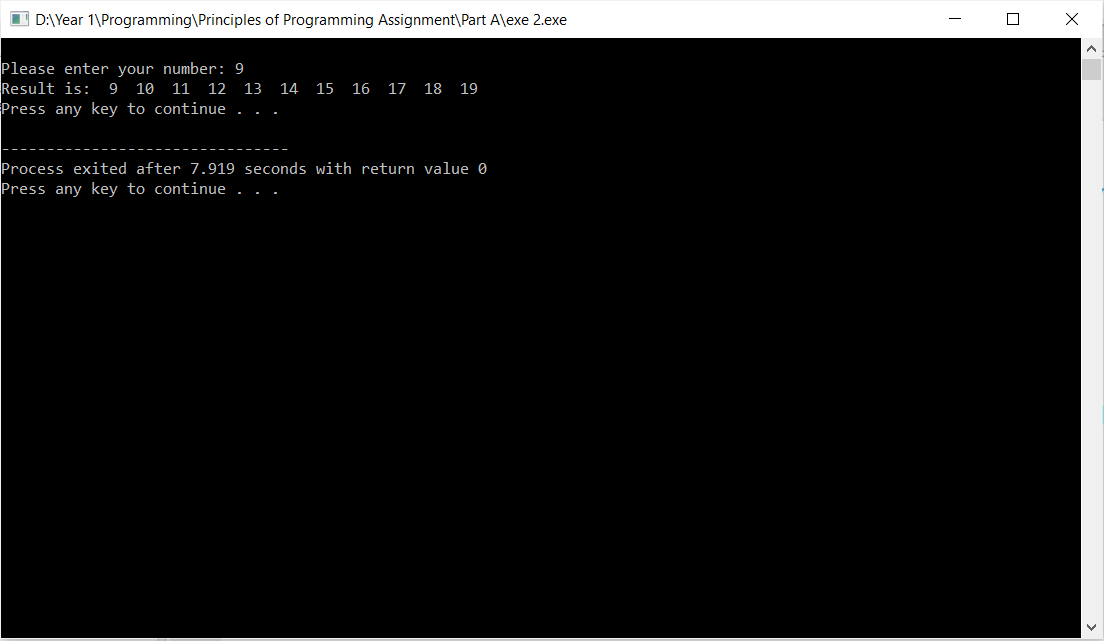
## Design, source code and testing





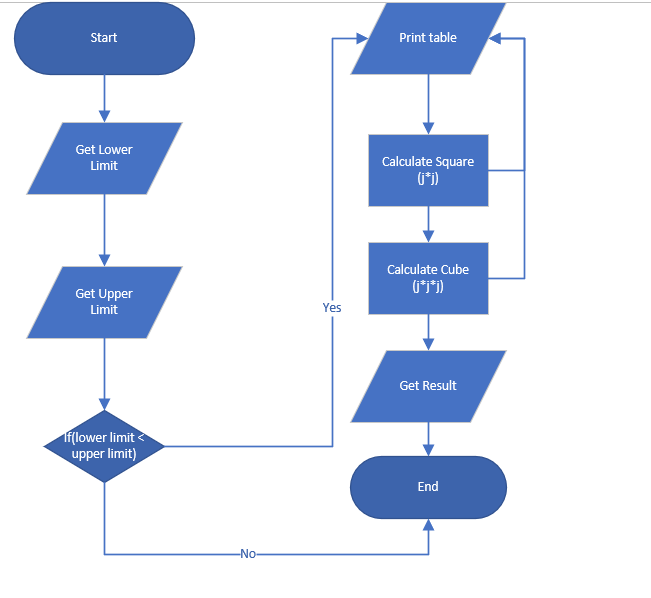
# Part A: Exercise 2

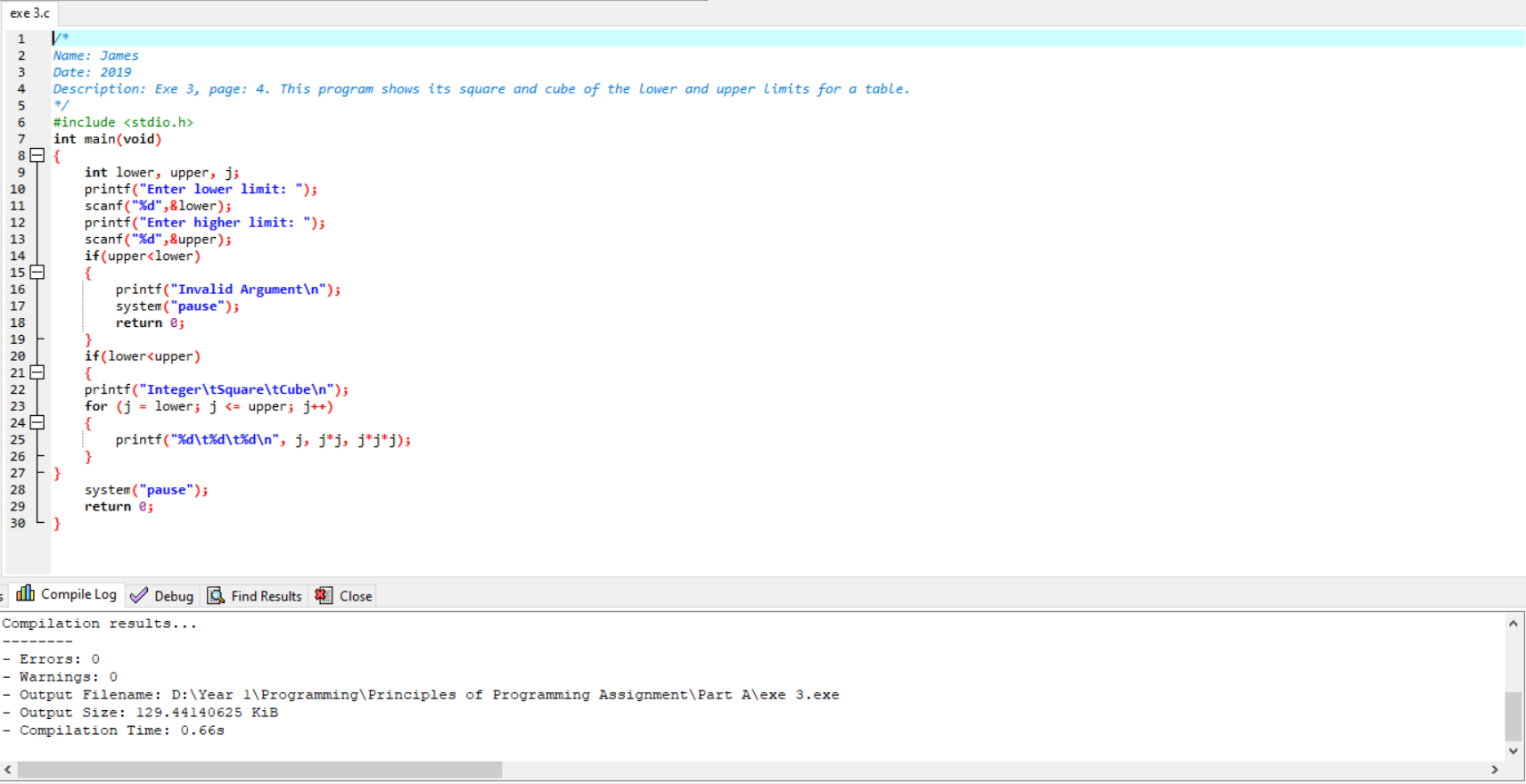
## Design, source code and testing

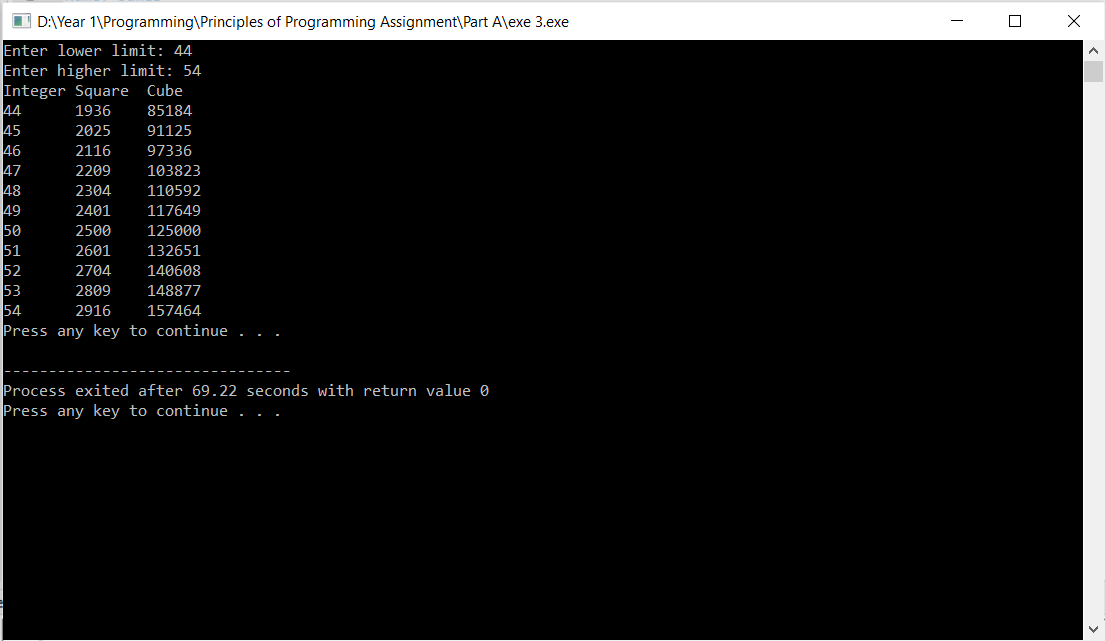


# Part A: Exercise 3

## Design, source code and testing

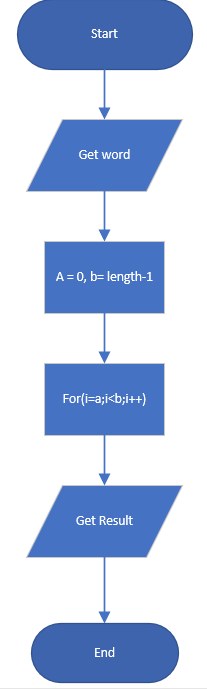


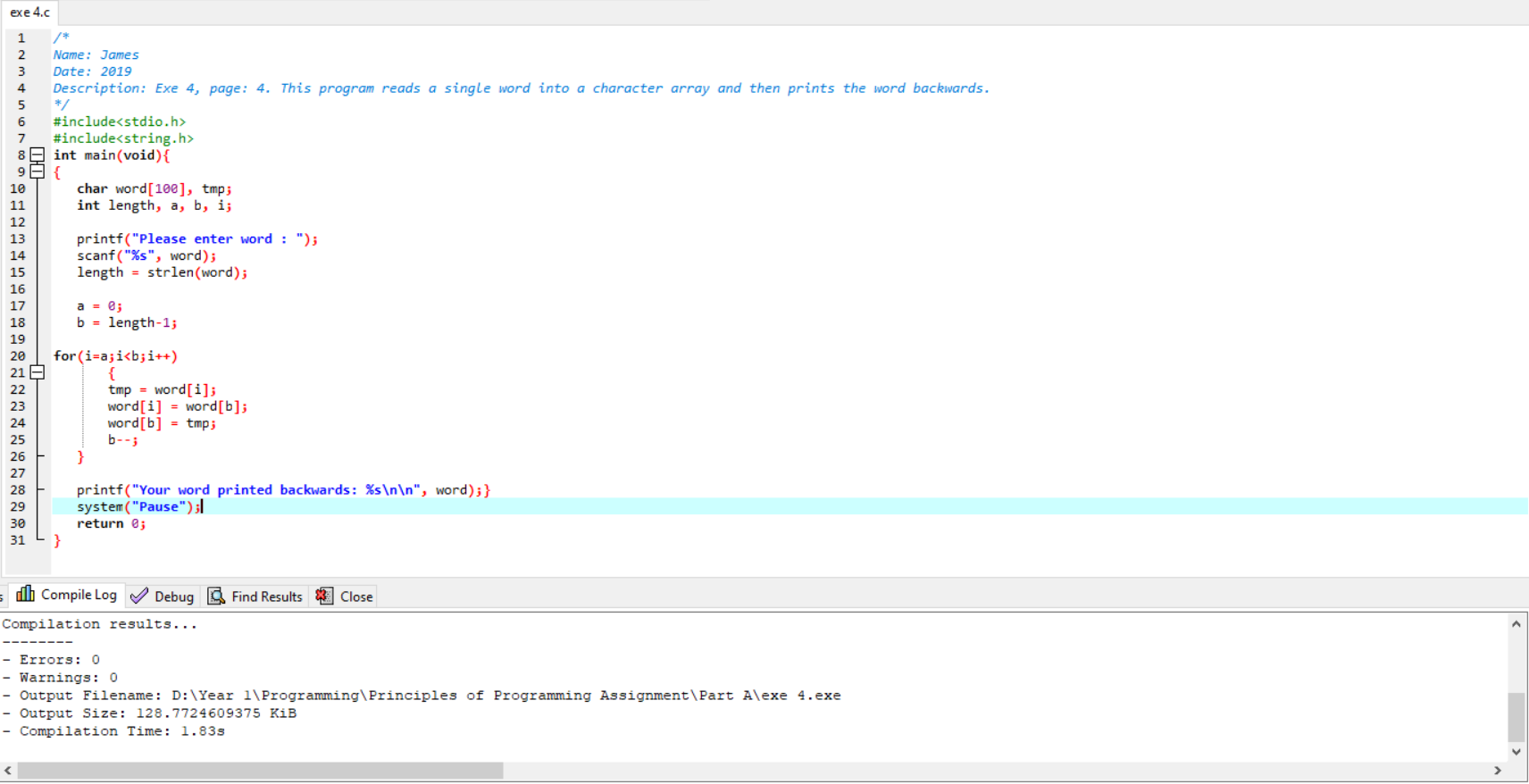
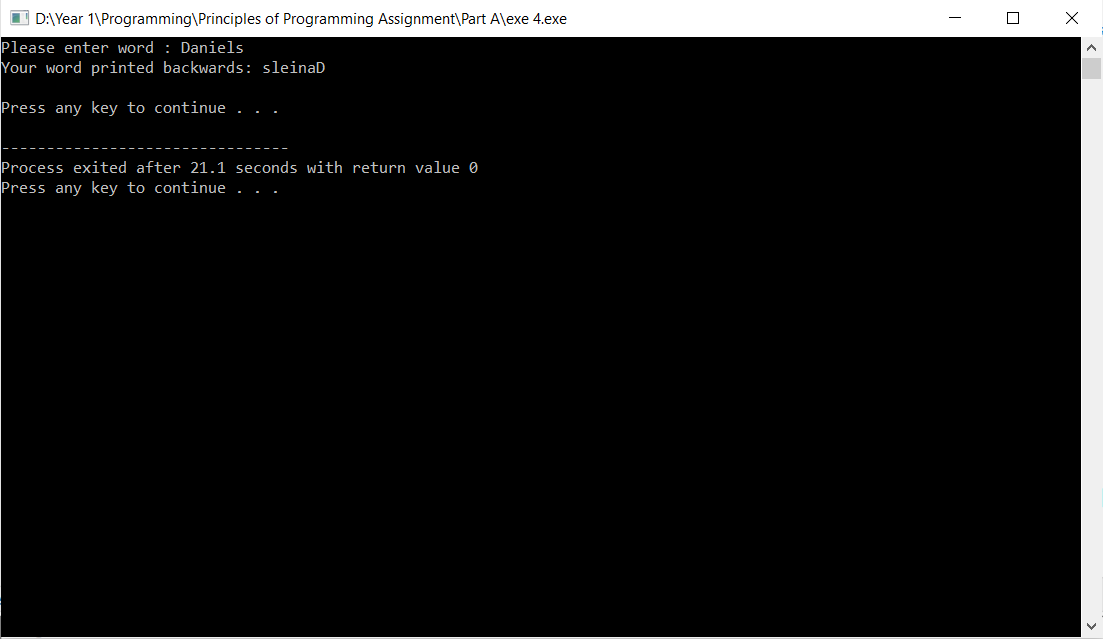




# Part A: Exercise 4

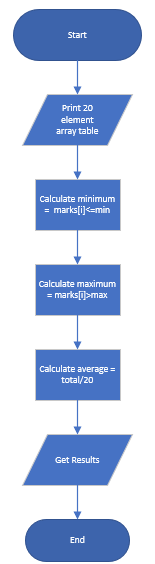
## Design, source code and testing

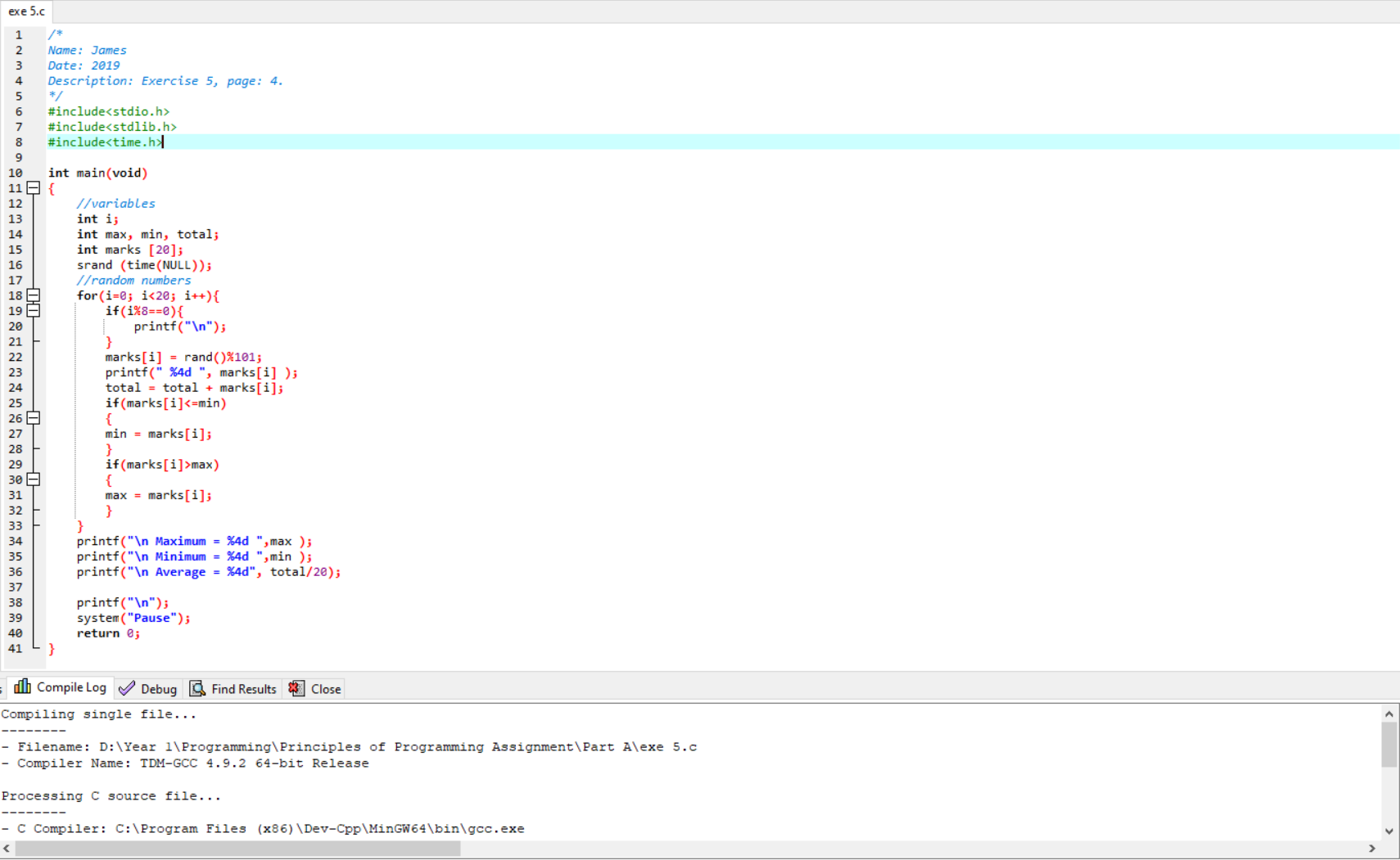
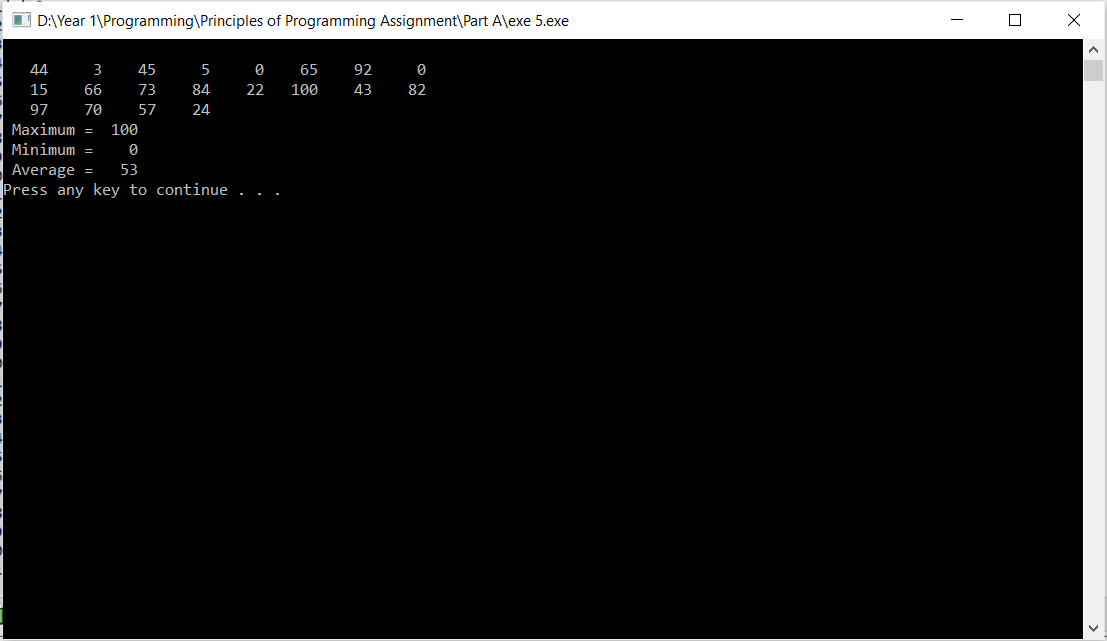




# Part A: Exercise 5

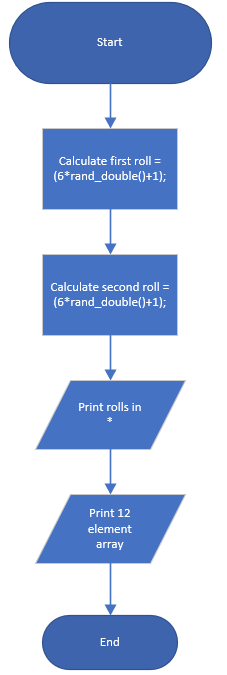
## Design, source code and testing

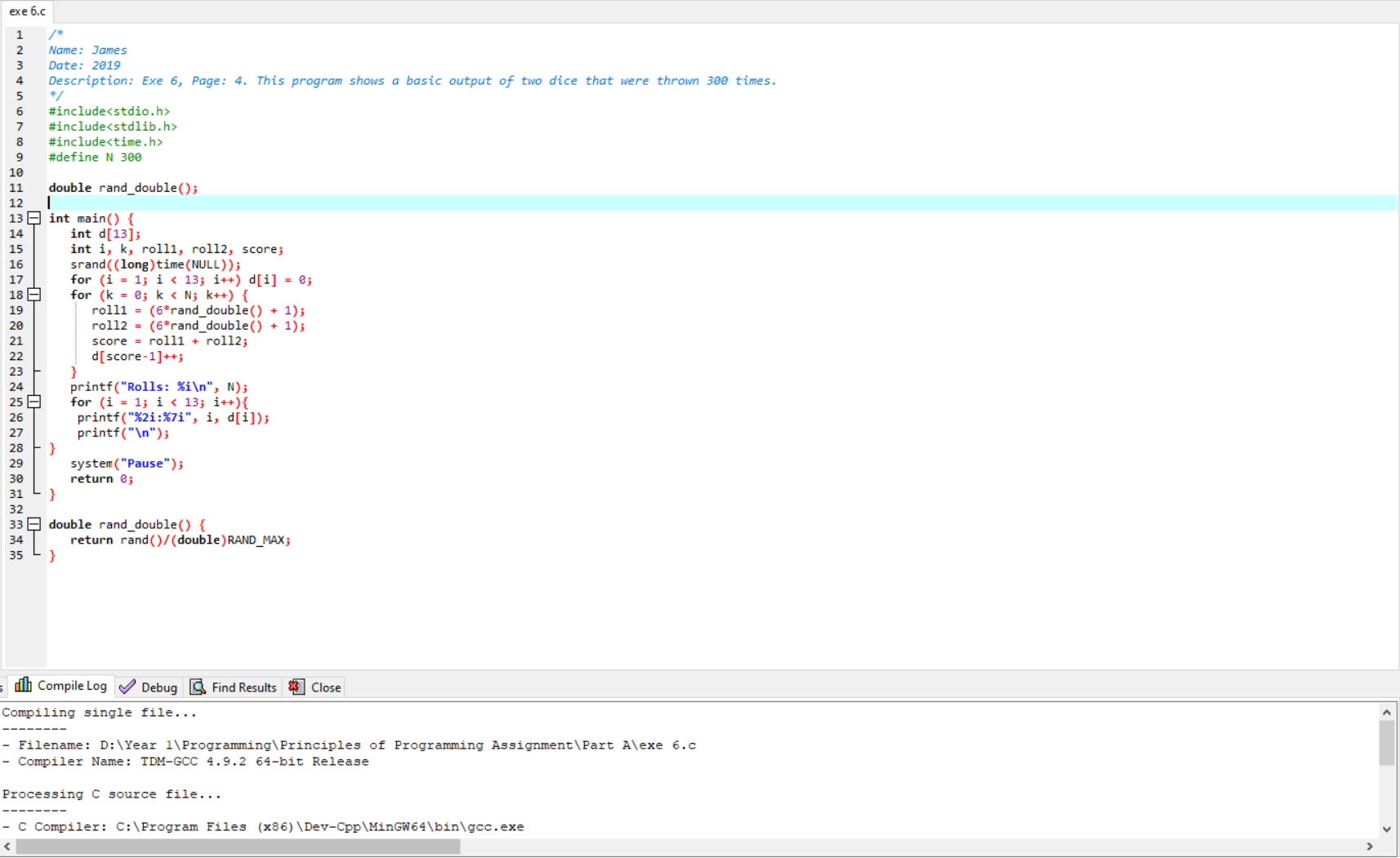
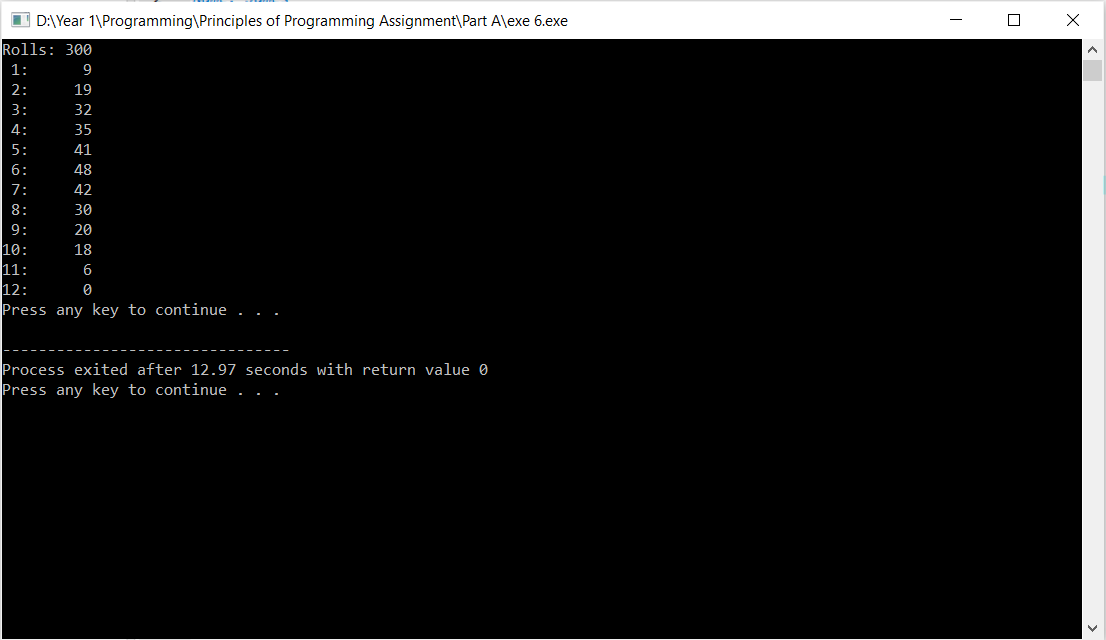




# Part A: Exercise 6

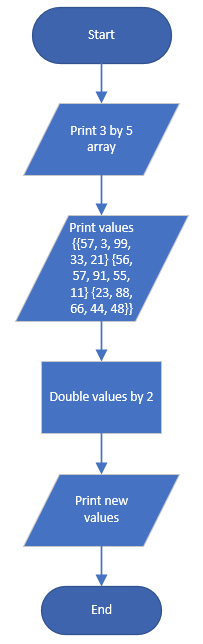
## Design, source code and testing

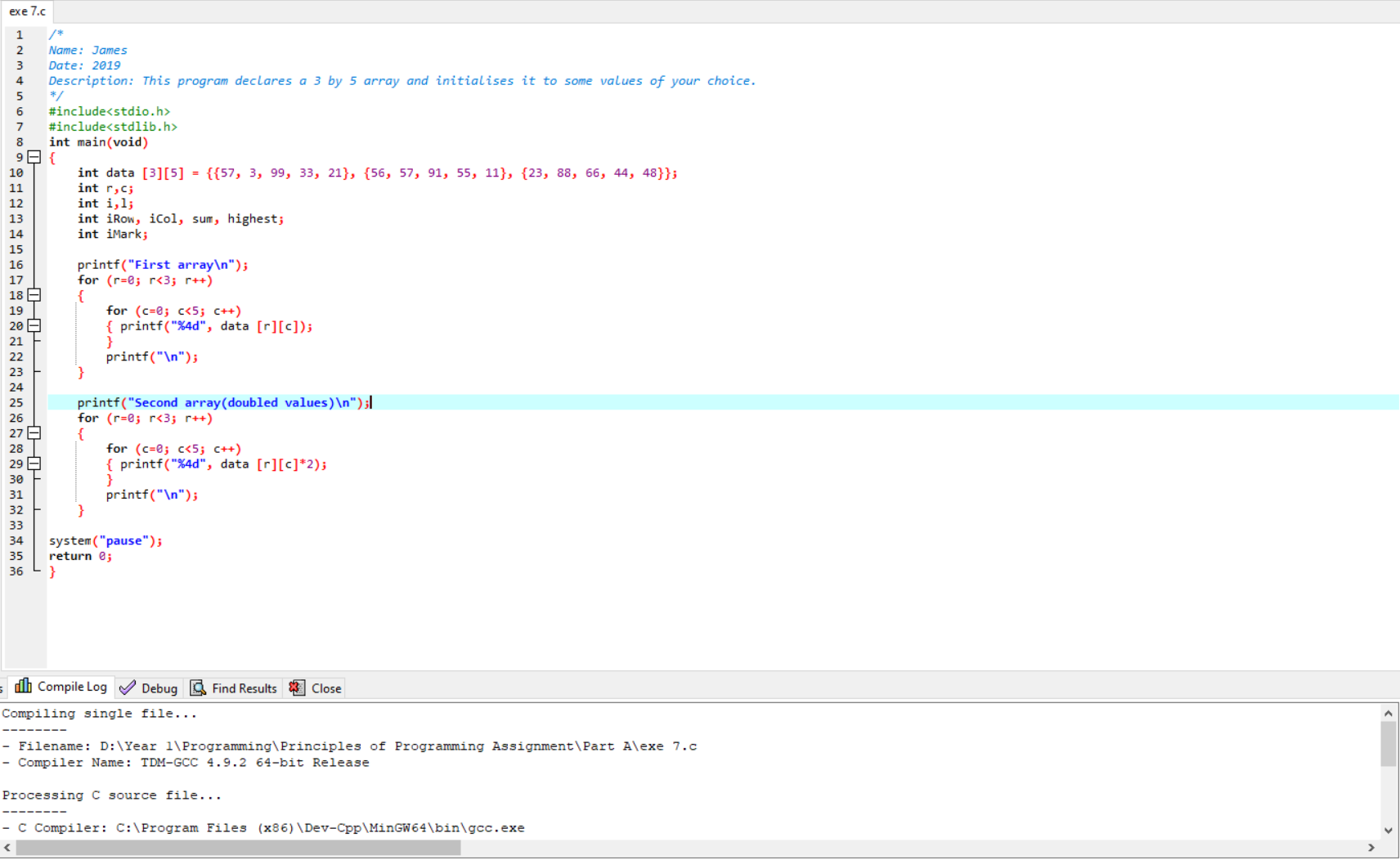
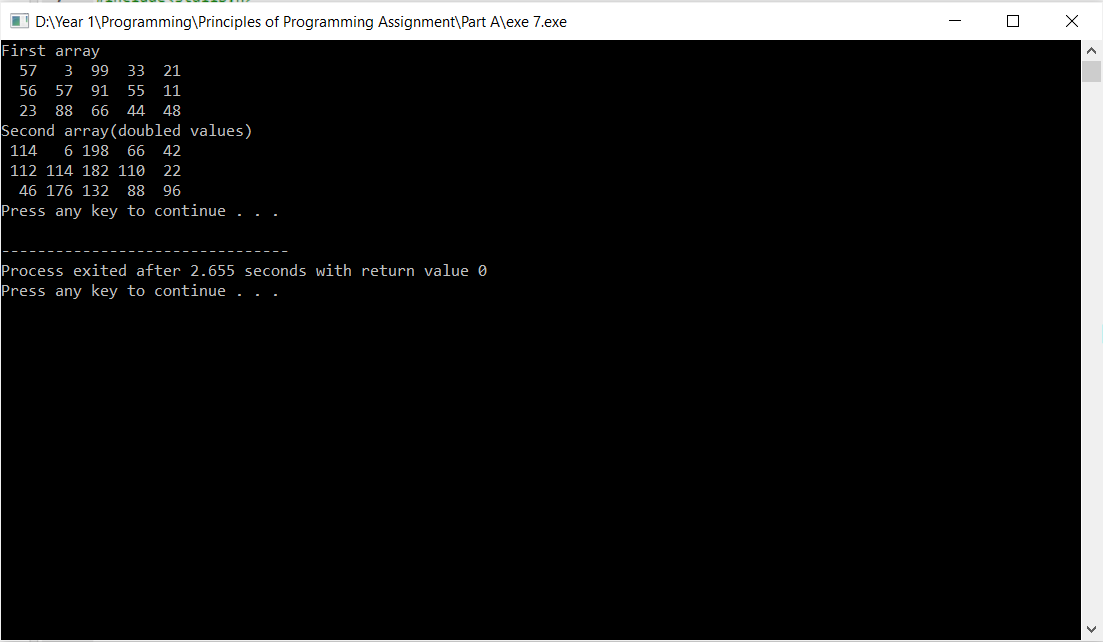




# Part A: Exercise 7

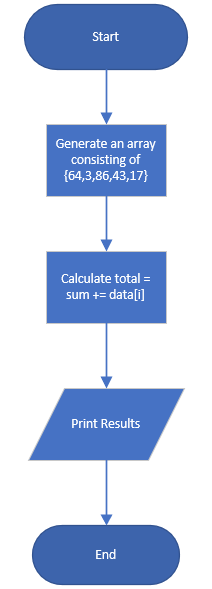
## Design, source code and testing

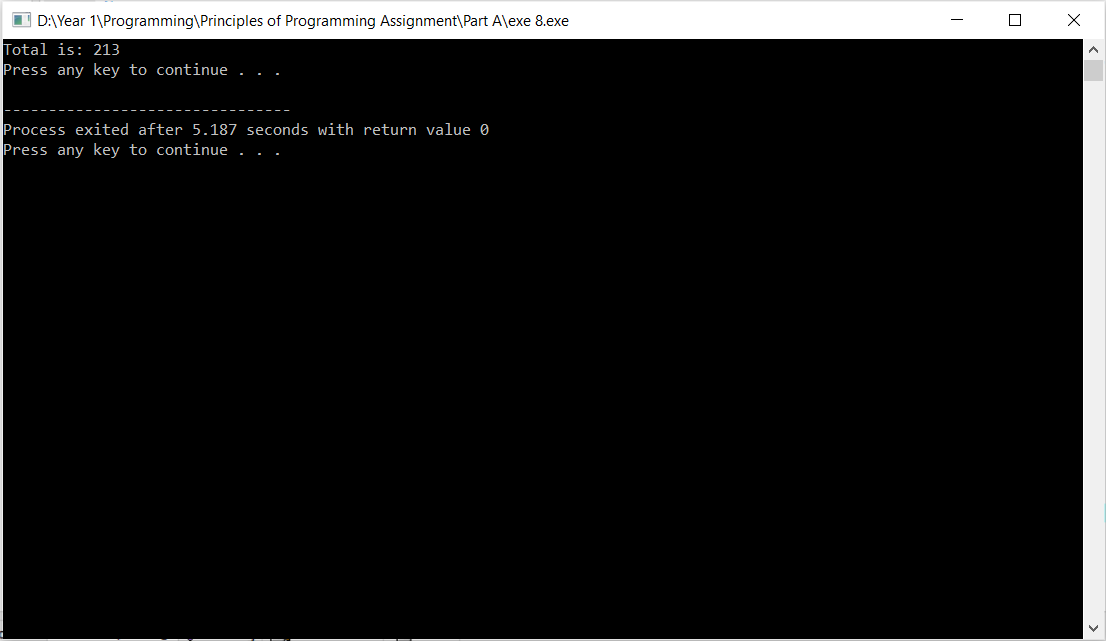




# Part A: Exercise 8

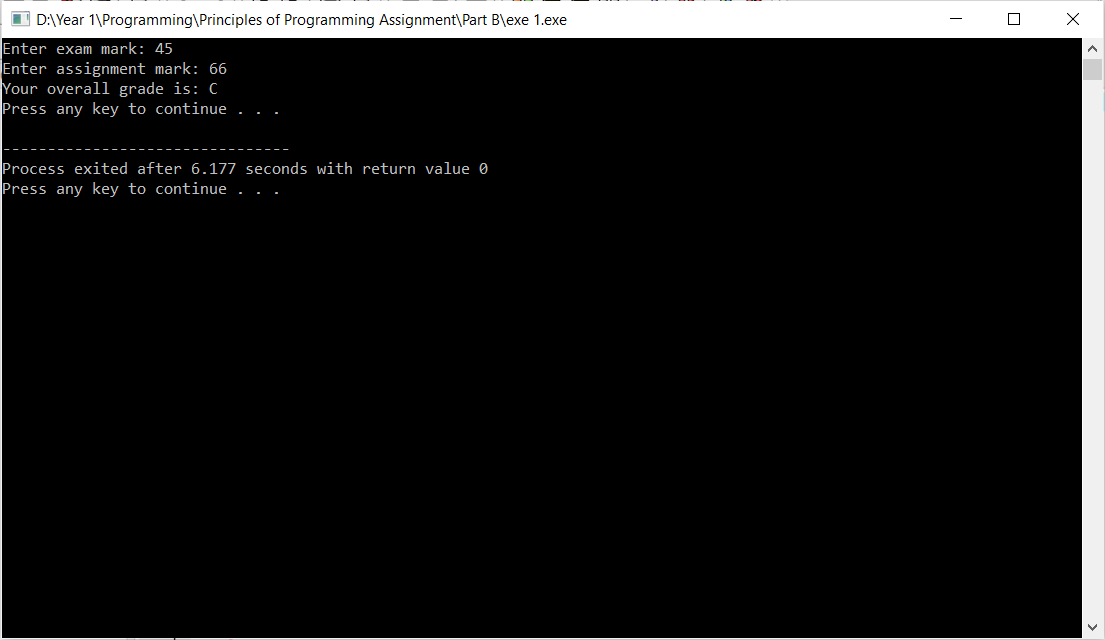
## Design, source code and testing





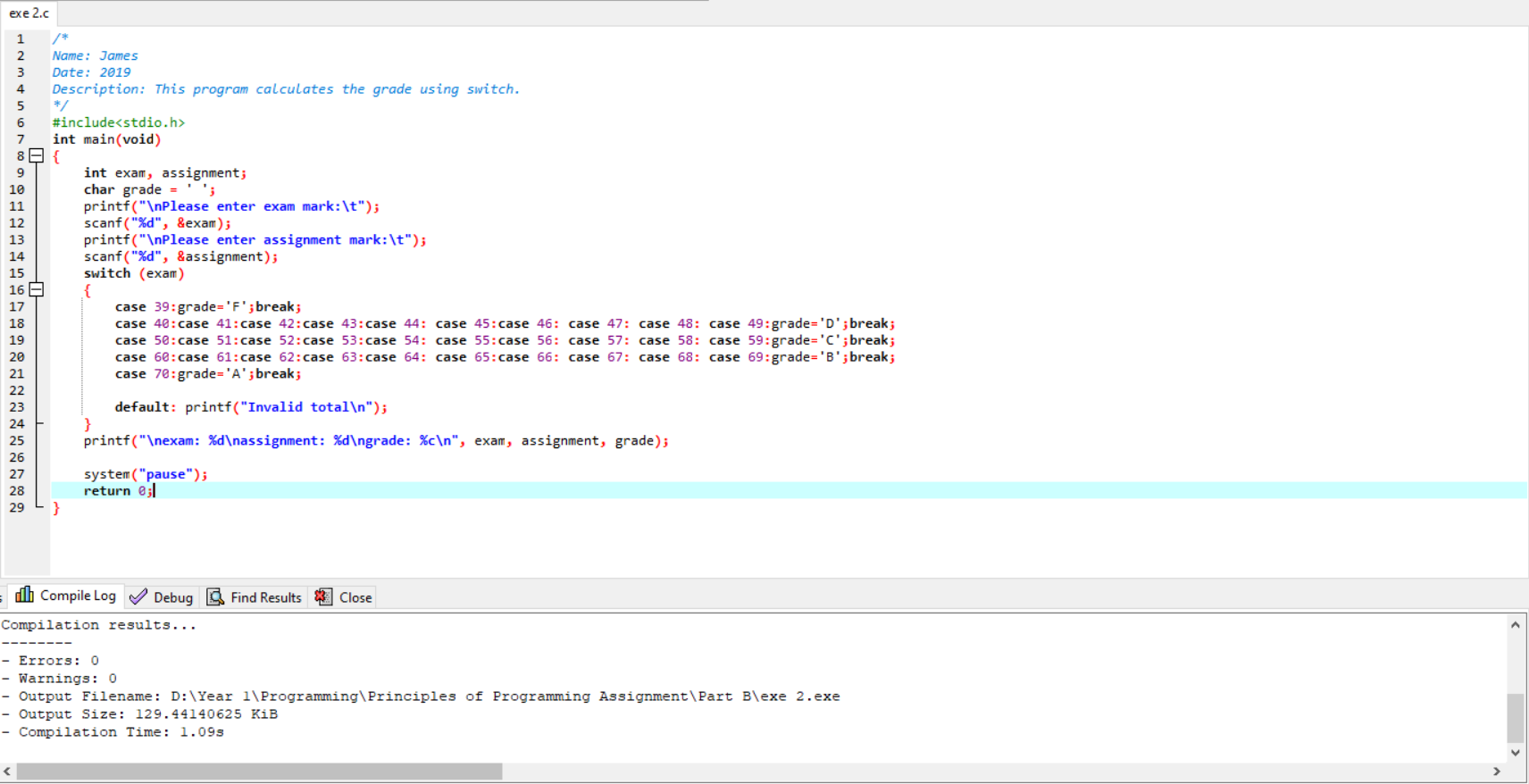
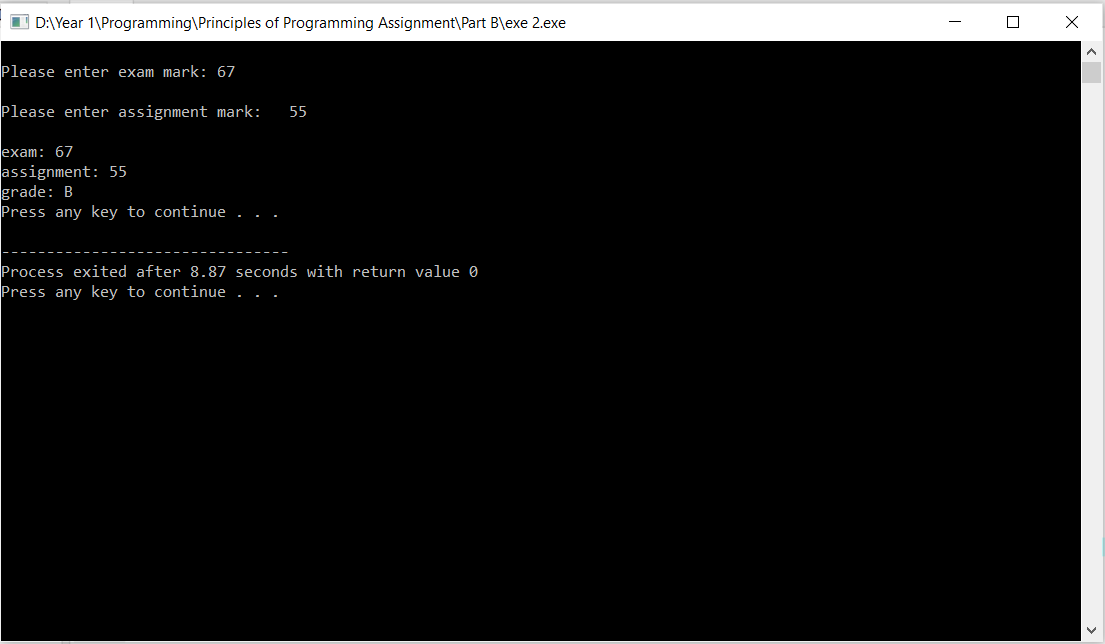
# Part B: Exercise 1

## Design, source code and testing



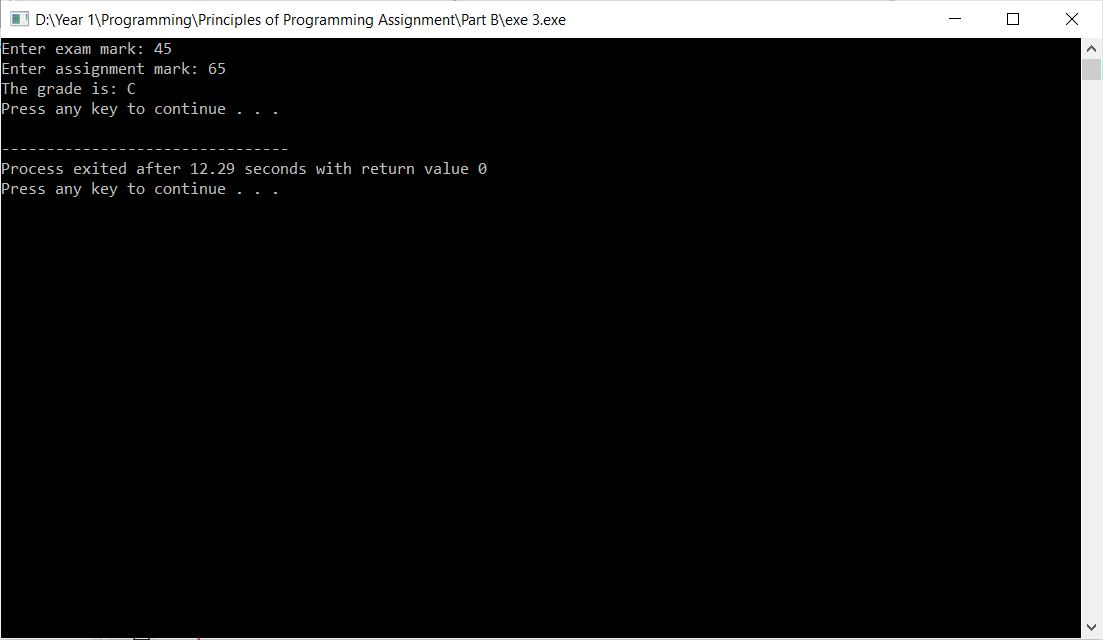
# Part B: Exercise 2

## Design, source code and testing



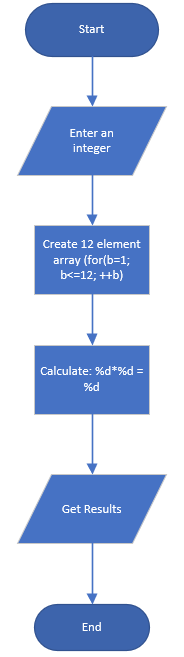
# Part B: Exercise 3

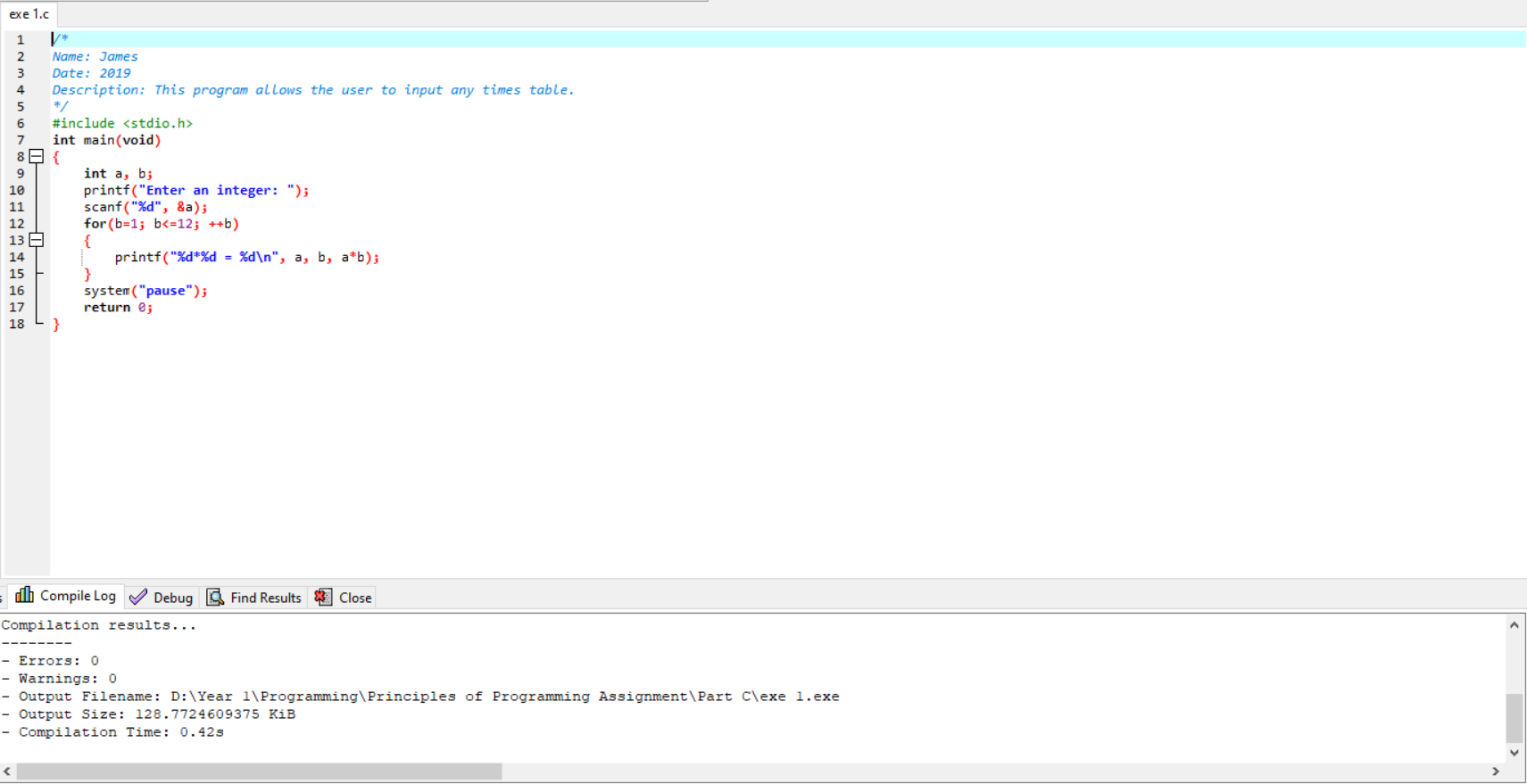
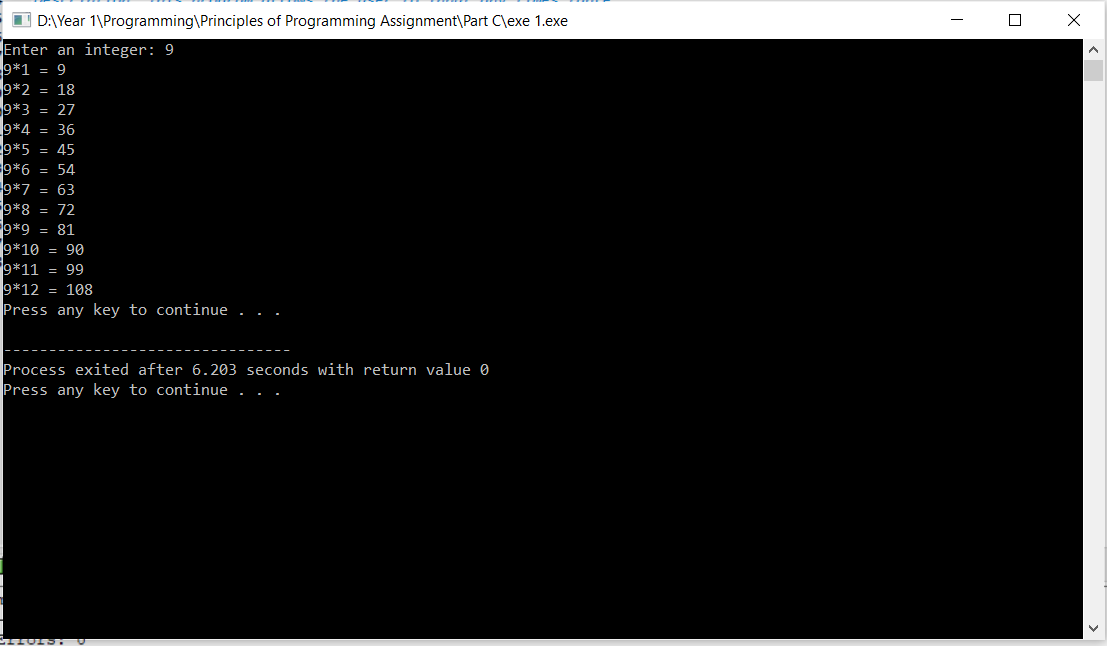
## Design, source code and testing



# Part C: Exercise 1

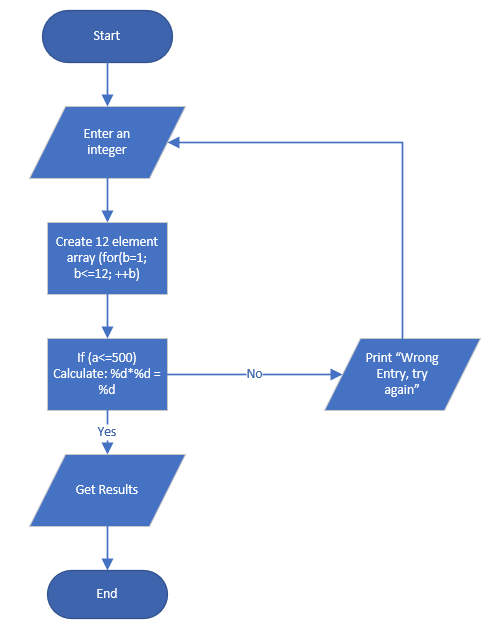
## Design, source code and testing

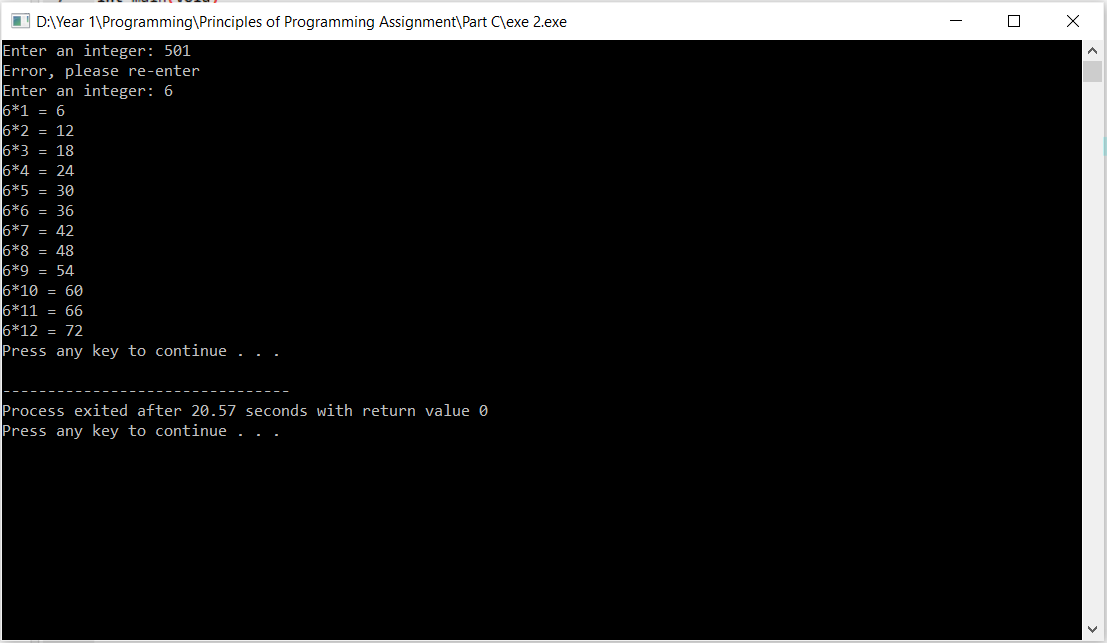




# Part C: Exercise 2

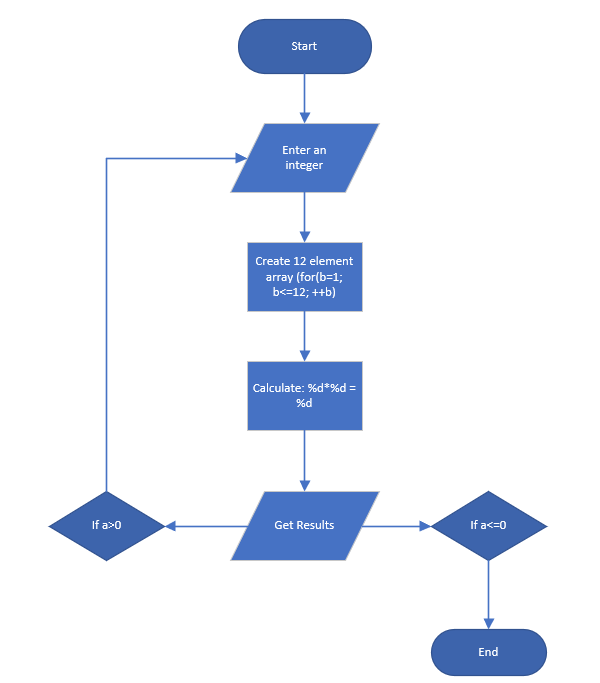
## Design, source code and testing

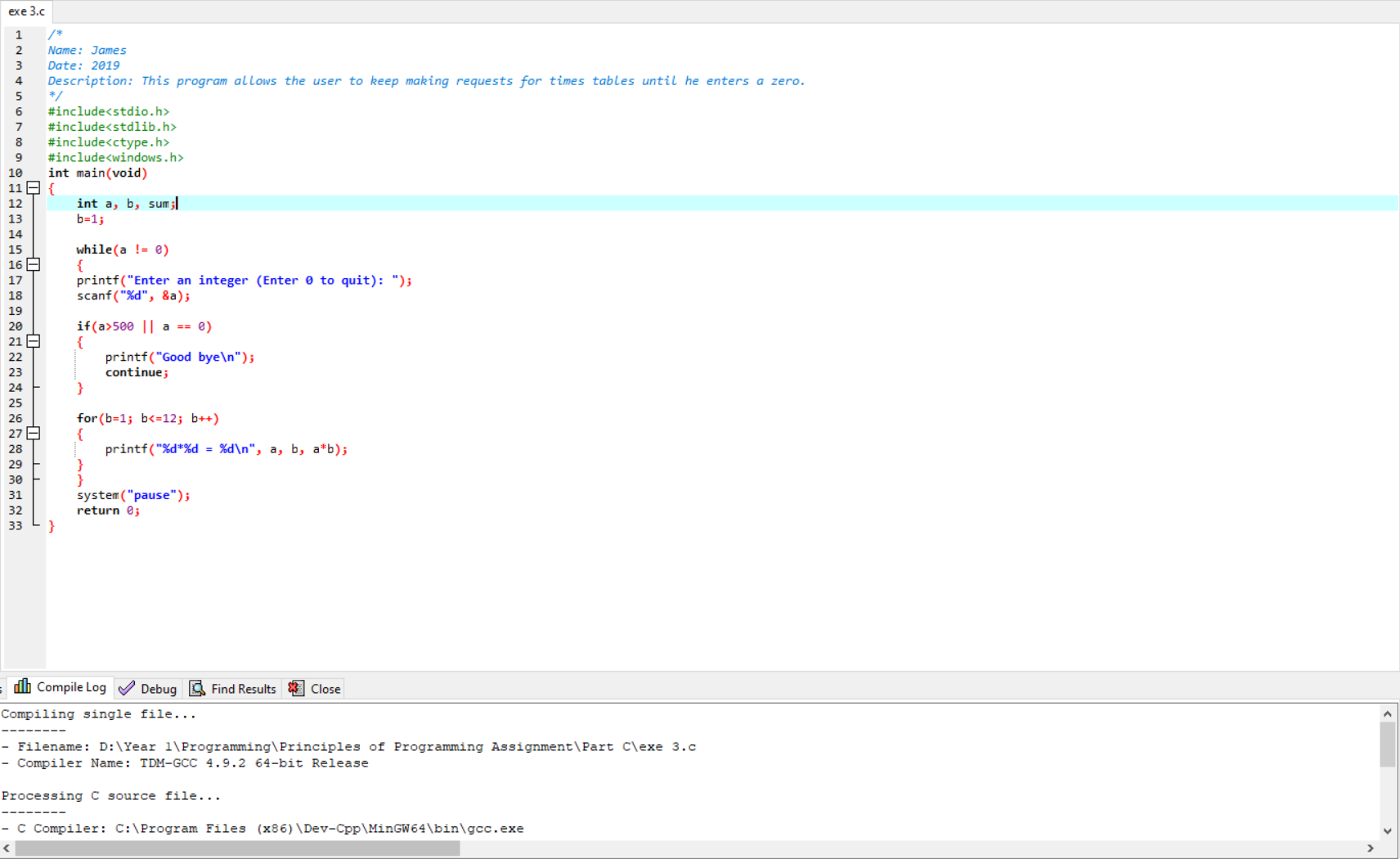




# Part C: Exercise 3

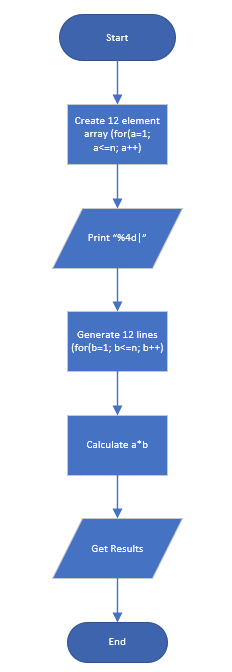
## Design, source code and testing

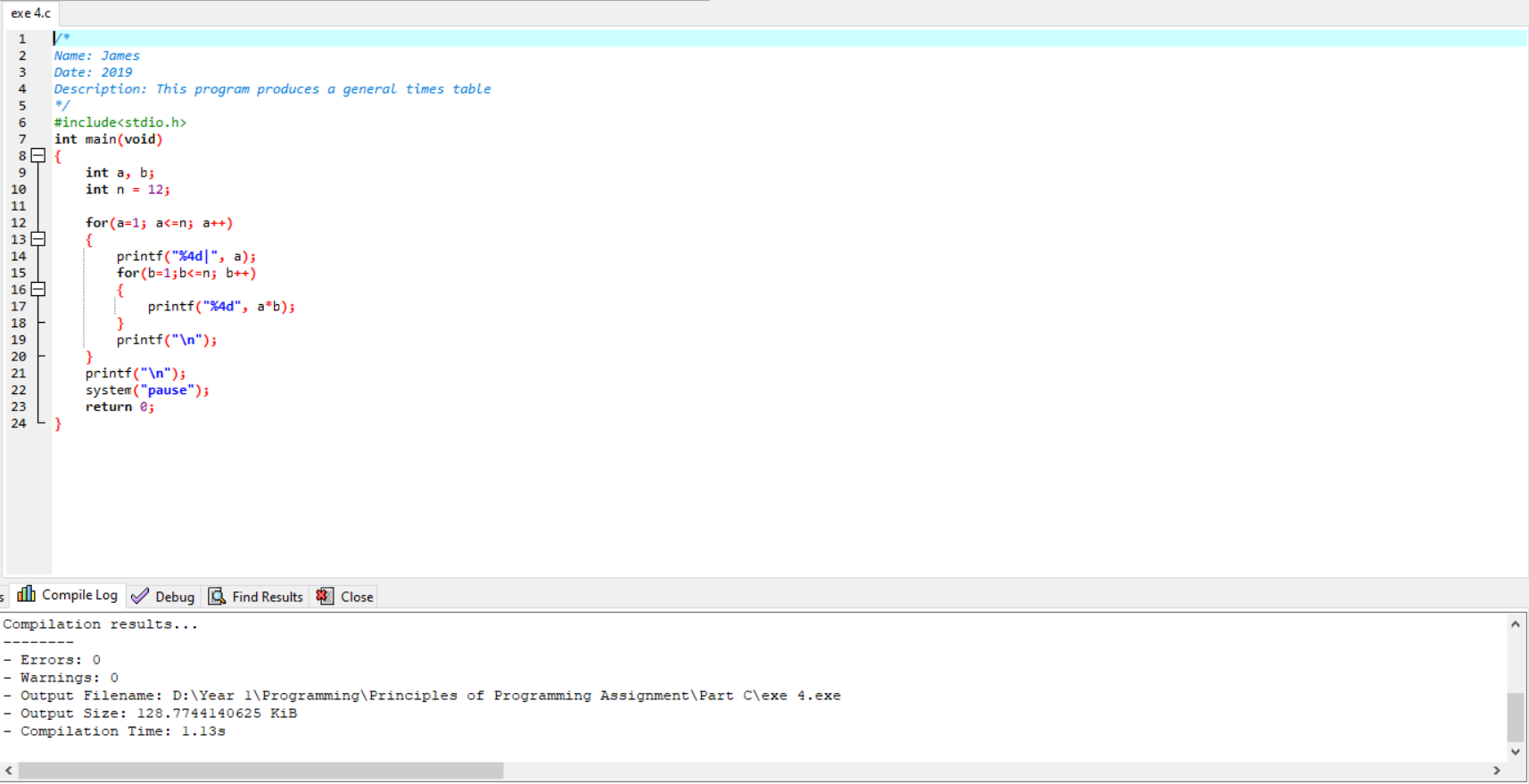
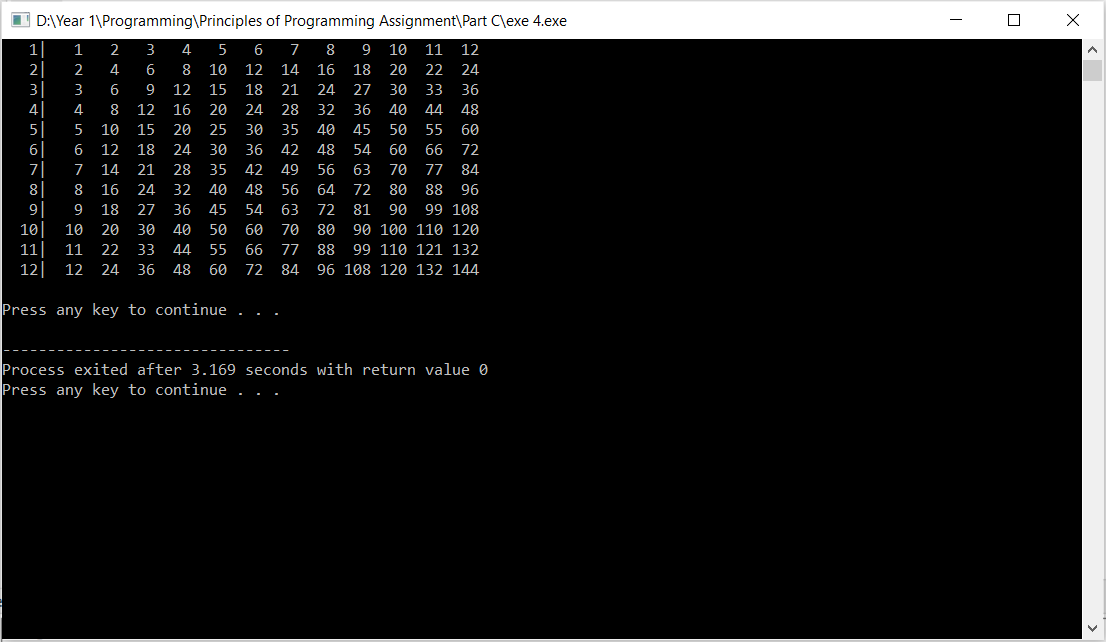




# Part C: Exercise 4

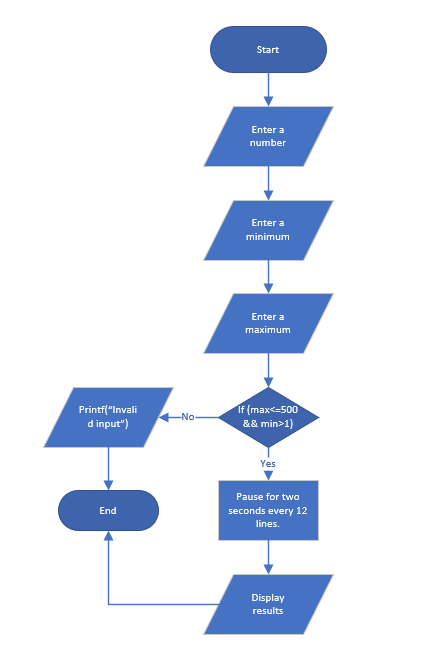
## Design, source code and testing

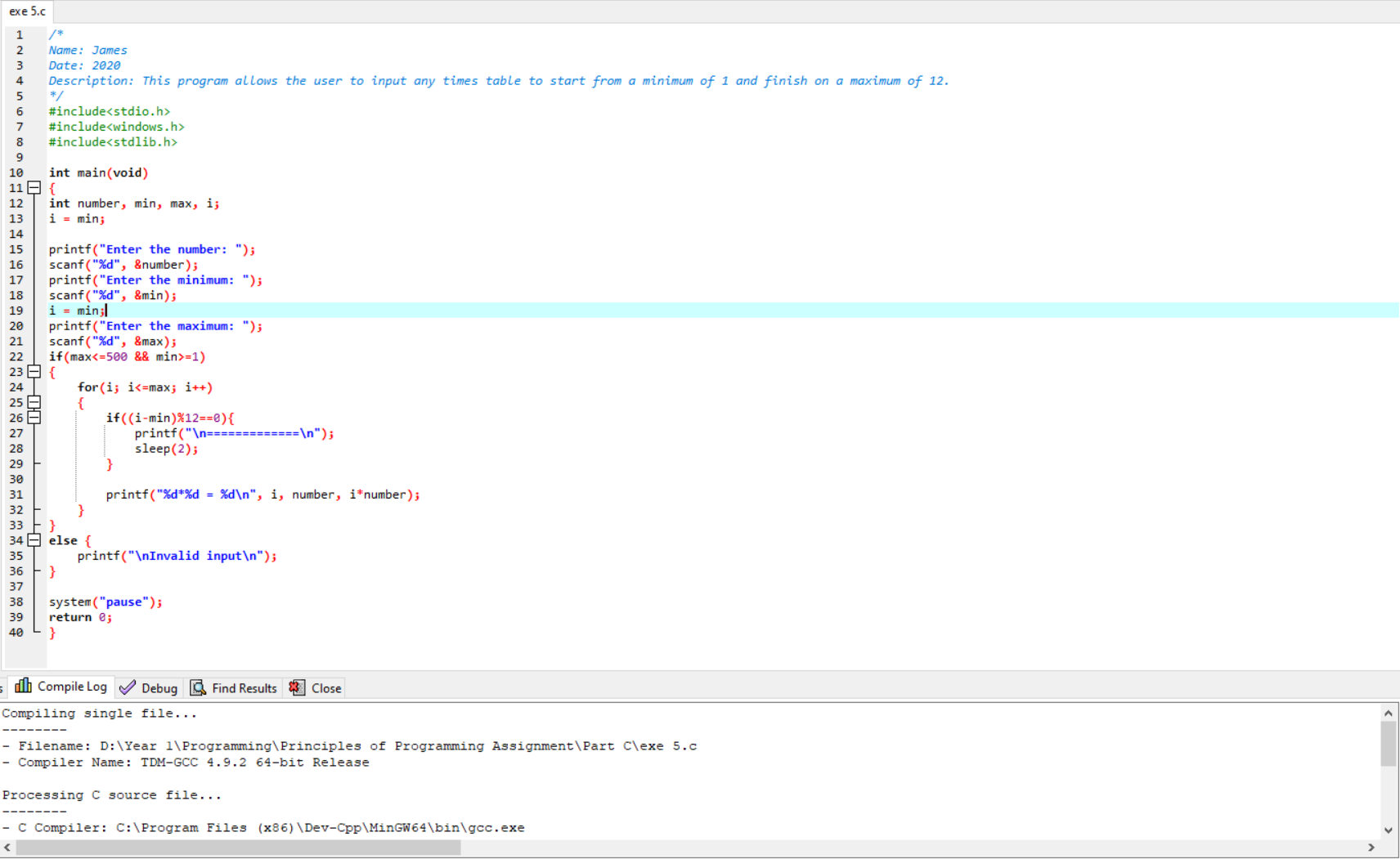
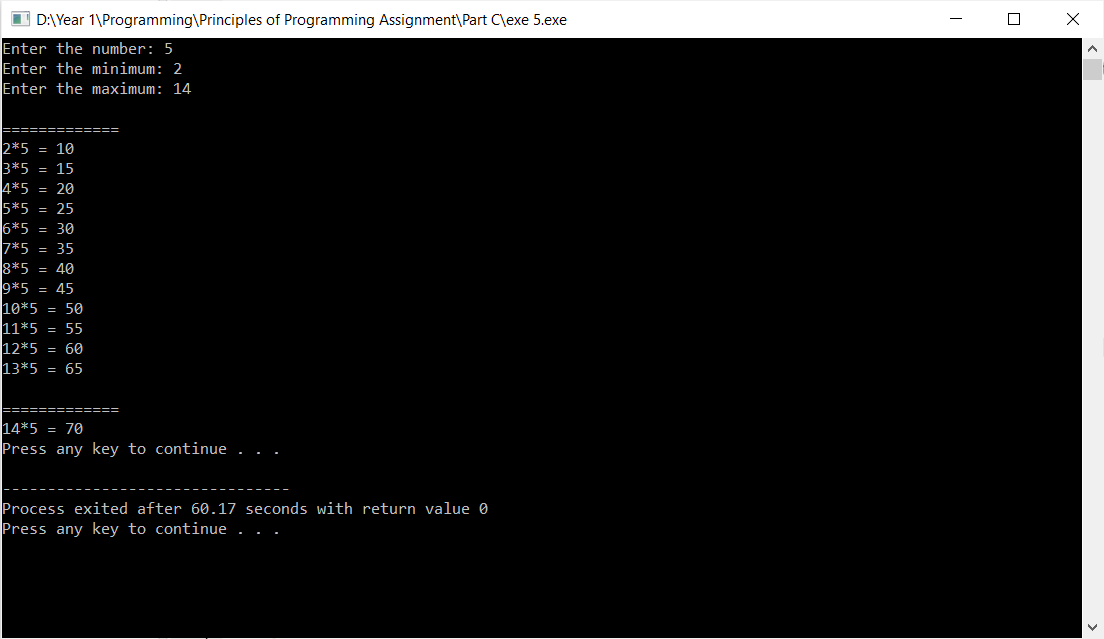




# Part C: Exercise 5

## Design, code source and testing

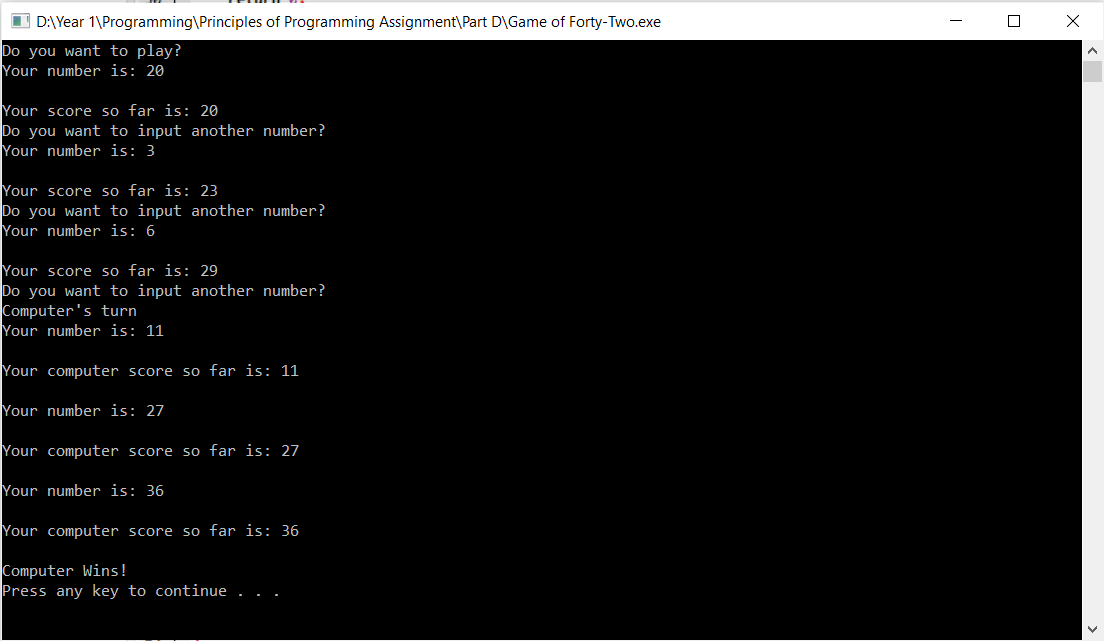




# Part D

## Design, source code and testing





# Reflection

A number of resources have been used for assistance with the programming such as a book one (Farahani, 2020) based on principles of programming and book two (Farahani, 2020). As well as some websites such as w3 resources (w3resource, 2011). While all exercises have been attempted, unfortunately it was unsuccessful to convert the number of dice rolls into spots in Exercise 6, Part A. However this will be used for reference in the future as this area will be studied in order to better understand this area of programming. This entire project will be used as a basic idea of what areas to study further on in order to improve as a student.

As well as that however, from my perspective it’s clear to see what other fields in programming that I personally know well. Some examples of this are the if-else statements, for loops, arrays and while loops. From my mindset, these are easy to define what they do and what purpose they have in c-programming. Nevertheless, these will still be studied further when given the time along with the many other factors of c programming in order to improve as a university student.

During development of this project, a few changes have been made here and there as opposed to my original ideas of how to program some of these exercises. One example is Part A, Exercise 7. At first glance it was planned to declare a 3 by 5 array of inputs chosen by myself, and then create another 3 by 5 array with all the inputs doubled. However this was then changed to just simply doubling the first array below. Using nested for loops I was able to print both the first original array and then post it again but doubled.

Some of these exercises were also approached with different ideas in mind compared to everyone else. With Part C, Exercise 5 for example it was my own personal approach to have the times table start and end on a certain number up to 500 inputted by the user. Other ideas of this exercise included creating a program for times table but limiting it to just 1-500 while letting the user decide what times table to use.

A variety of control structures have been shown with these exercises such as for loops, nested for loops, else-if and switch statements, while loops and arrays. These have been used as they are needed to complete the programs while also showcasing a variety of different solutions in c programming.

If I was given the chance to improve some of these exercises then I would’ve taken advantage of functions more, so that a general understanding of how to program easier and more efficiently could be developed further. Also if given more time, another statement other than the if-else and switch statements could be attempted to again further develop knowledge. Most certainly improvements on Part A, Exercise 6 needs to be researched. Another thought on how to improve is creating the design tables on Visio first before building the program, as maybe it could give a clearer idea of what to code in future.

# Conclusion

In conclusion, this project has for the most part been a success and it will hopefully prove useful in future along with feedback given for it. C programming is also extremely important to learn efficiently as it’s a popular language in the IT industry, thus having good knowledge of C is essential for a good position for an IT job. In fact to prove this, C has been reported to be TIOBE’s programming language of the year as of now. With a yearly increase of “*2.4%”* this year according to the SD times (Sargent, 2020). This just proves that c programming is still very much relevant today and is arguably even the most important language in the entire industry. It is used for several and extremely important areas in IT such as *“developing desktop applications”*, *“developing system applications”* and *“developing an operating system”* (Guru99, 2020).

The overall experience with this project has mostly been decent. All these exercises have been capable of being attempted, and the assistance was there if it was needed. Of course there are things that can be improved upon and this project has helped showcase a general idea of what needs to be looked upon the most.

This project has been worthwhile and beneficial as a university student since it has given the opportunity to learn more about c programming in general and has provided more insight into the language. Hopefully it will also provide some guidance for similar popular languages in computing such as C sharp or PHP. C programming is overall fairly beginner friendly to learn, so it’s recommended to learn it first before any other computer languages as it’ll give a nice basic idea of what to expect from programming in general as you’ll need to know a huge variety of languages for programming in general.

# References

# Bibliography

Farahani, M., 2020. *Principles of Programming Book 1 of 2.* Colchester: University Centre Colchester, University of Essex.

Farahani, M., 2020. *Principles of Programming Book 2 of 2.* Colchester: University Centre Colchester, University of Essex.

Guru99, 2020. *What is C programming language? Basics, Introduction and History.* [Online]   
Available at: https://www.guru99.com/c-programming-language.html  
[Accessed 27 January 2020].

Sargent, J., 2020. C is TIOBE's programming language of the year for 2019. *SD Times.*

w3resource, 2011. *Web development tutorial | w3resource.* [Online]   
Available at: https://www.w3resource.com/  
[Accessed 27 January 2020].

# Appendix A