

MSC Research Proposal Programming a
deterministic rule set to Find coding errors

Matthew Cooze 1911017

12/11/2021

Supervisor: Tim Bashford
Second marker:

Project submitted as part of the requirements for the award of Computer
Networks and Cyber Security (MSc)

November 2021

Chapter 1

ABSTRACT/SYNOPSIS

The goal of this project was to prove if deterministic rule sets can be used for finding errors in code, and security vulnerabilities. with secondary goals such as finding logical errors and if they are found can a solution be found automatically, another secondary goal of the project was to see if the deterministic rule set could also be used for automatic optimization after all errors were removed from the code, the research proved successful in some of its goals and found failure in others. the secondary goals had to be removed as they were more complex than expected and this lead to errors in the main goals, however the results of the research show that it will defiantly be possible to use a deterministic rule set to find computer errors and security vulnerabilities.

Contents

1	ABSTRACT/SYNOPSIS	2
I	introduction	6
2	aim	10
3	objectives	11
4	structure of the project	12
II	literature review and research	14
5	introduction	15
6	literature methodology	16
7	Regex	18
8	optimization flags	19
8.1	different types of optimization flags	20
8.2	compiler optimization effects on code reliability	21
8.3	compiler optimization automation	21
9	algorithms	22
9.1	rapidly exploring random tree	22
9.2	Recursion	23
9.3	Logical	23
9.4	Deterministic or Non-Deterministic	23
9.5	Exact or approximate	24
9.6	Divide and conquer	24
9.7	Back tracking	24
10	ECC Memory	25

11 deterministic rule set	27
III research methodology	28
12 introduction	29
13 methodology	30
14 research overview	32
14.1 research process	32
14.2 research philosophy	33
14.3 research methodologies	34
14.4 data collection	35
15 research design	37
15.1 research deliverables	37
16 triangulation	38
17 conclusion	39
IV design and implementation	40
18 introduction	41
19 problem formulation	42
20 design	44
21 implementation	45
22 conclusion	50
V testing and evaluation	51
23 testing	53
24 evaluation	55
VI conclusions and recommendations	57
25 summary	58
26 conclusion	59

VII	Appendices	60
VIII	Declaration of Originality	61
IX	FORM OF CONSENT	63
	27 recommendations for further research	65
X	reflections	66
XI	references	68
XII	ethics form	72
XIII	logbook	86
XIV	other material	88
	28 main code	89
	29 code for testing other programs	94
	29.1 code 1	94
	29.2 code 2	96
	29.3 code 3	98
	29.4 code 4	100
	29.5 code 5	102
	29.6 code 6	104
	29.7 code 7	106
	30 link to demo	118
XV	glossary/key	119

Part I

introduction

the goal of this project is to undertake a major project to gain experience in a chosen field and do in depth research. the chosen field for this project is programming more specifically if deterministic rule sets can be used for finding errors and security vulnerabilities.

one of the two main goals is to increase security this is due to machines being increasing automated and in charge of key systems, yet security and reliability are not being advanced at the same rate and a lot of programs seem to be lacking any security measures at all, there are many examples of this online from large companies keeping passwords in plain text to changing a URL on moonpig to get into random users accounts without a username and password, these are small threats compared to some e.g no security checks on external connections for a nuclear power plant in America.

programming errors are more common and security features always seem lacking, as memory space and ram have been made readily available, optimization seems to be only a concern to high end programs, some programs now use more memory and ram than needed due to lack of optimization knowing newer computers have the memory and ram to run them anyway companies have decided to save money instead of doing optimisation for some software. this is clearly visible in the games industry currently. bugs left in for later patching, current pace makers have had examples of being hacked in 2018 black hat security conference researchers Rios and Butts(2018) showed the ability to hack a pacemaker and withhold or add more shocks at will, later on Butts said. "Almost anybody with the implantable device in them is subject to the potential implications of exploitation." if it were in active use this could be used by assassins/hackers to kill a target from miles away using remote controls due to lack of security features. "If you just code sign, all these issues go away, but for some reason they refuse to do that," Rios said, code signing is a very basic very easy to implement system that takes up very little memory space and the company refused so as to save the small amount of time and effort needed to defend the system. another example is the attacks on US nuclear power plants luckily the systems are separated and hackers only got into the business network,"the operational computer systems are completely separate from the corporate network,"Jenny Hageman(2017).

in conclusion as automation takes over more roles around the world it will be more important to have a way to find and fix errors before programs are run for the first time, especially for critical systems: rockets, nuclear plants, factories etc places where many lives are at risk if something goes wrong or a rocket comes back down to earth and self destruct does not work due to a software issue. in case of optimization it is crucial for future industries to make sure automatised systems can react in time to unexpected situations for example a human inside a robot arms swing path needs the system to stop immediately. lag from lack of optimization or bugs could kill the worker in this situation. security will be key for making sure hackers cannot weaponise future inventions.

the second main goal of the project is to help reduce coding errors, these can be massive problems waiting to happen from small problems that do not effect many people for instance a website going down shortly to massive errors that could endanger humanity itself e.g nuclear power plant disasters, however computers are critical for modern day life and are used in everything the programs are used to automate simple jobs and calculations as such solutions must be found or made, two good examples of this are the missing hyphen in the "Mariner I" code due to this error transmission of incorrect signals were allowed pass the checks in place and were mistaken for guidance signals, this meant NASA had to self destruct the rocket when it went off course and would of crash landed somewhere on earth.

another example of missing code causing errors is lockhead martin once had an agreement to make planes for a military due to the length of time to produce all the planes, the agreement had a program used for calculating inflation over the years the project would take, this had a typo in the formula this caused the company to loose 70 million. NASA'S(1962-2021) official report said "During the periods the airborne beacon was inoperative the omission of the hyphen in the data-editing program caused the computer to incorrectly accept the sweep frequency of the ground receiver" "This caused the computer to swing automatically into a series of unnecessary course corrections with erroneous steering commands which finally threw the spacecraft off course." in conclusions the goal of this project is to develop a program that can detect errors in code to prevent such mistakes from being made, for example missing end lines or a badly made loop.

the two optional goals of the project can be seen bellow

the first optional goal of the project is to find and remove logical errors, logical errors can cause massive problems these are normally worse than code errors as there will be no sign if it is working incorrectly if not tested extensively, an example of logical code errors causing problems is the Ariane 5 Flight 501 coding errors created due to errors of trying to input a 64 digit number into a 16 bit space caused the rocket to tip and destroyed the engines 37 seconds into its maiden voyage, this is an example of a logical error, these are much harder to find as the code might all look like it is working perfectly if one feature is not tested extensively sometimes the tests show it is working correctly but when the test is done in the real world it does not work as it should causing worse errors. making things even worse is that logical errors come in many different types from calling and executing the code in the wrong order, to a wrong variable type being assigned, this means some code when tested will show as working or the outputted answer will be wrong as the formula was completed in the wrong order. in conclusion the secondary goal of the project is to point out logical errors in code to reduce security errors and risks.

there is a secondary optional goal related to coding errors this goal is to

provide multiple possible solutions to problems found, test all the solutions automatically looking for the best fit and lastly to solve the errors before the program being tested goes live or they can cause an error. for example a small check that will show if the output of a formula is less than expected or over the expected output.

Chapter 2

aim

the aim of this project is to build and code a deterministic rule set and algorithms that can work to find faults in code fed into the program and point out if security measures are lacking

the secondary goals of the project are to include optimisation, finding logical errors and lastly provide possible solutions to errors found.

the project would take far too long if it was designed to work for all programming languages from the start as such it shall be designed and tested for feasibility using only one programming language. this will not only answer the hypothesis and allow for testing it will also make it possible to see how well it works and if a different method should be used instead for all programming languages in the future. this small scale test will also allow for changes to be made quickly as mistakes will not need to be found and fixed for all languages, there are also problems with this method due to it being small scale it is possible to miss how difficult it would be to make the program work for all languages and it will not show the true memory and ram needed for the program in the future.

Chapter 3

objectives

the objectives for this project are to find errors in code, point out when code is lacking security features, with secondary optional objectives of finding complex computing errors and logical errors before the code being tested is made live and to optimize programs, to use less memory etc. complex computing errors are normally made up of one of two categories logical errors and coding errors, as explained in the introduction logical errors have a greater chance to cause larger problems this is due to them being harder to find and potentially much more dangerous as the code will either get stuck in an endless loop or start doing its own thing if there are memory leaks or errors with the checks being preformed.

Chapter 4

structure of the project

the structure of the project is all about how to carry out the project, how it will be accomplished, what needs to be done to achieve the intended goals and how to go about them. for this project the structure will mostly be literature review for finding all needed information and filling in blanks in the study, this will then be followed up by coding a program to find the errors, lastly lots of testing to see the if the project can be successfully completed and if successful can it be replicated on a large scale with lots of different code samples.

the following research will be needed for the project

- research deterministic models in depth, this will be used to set the rules for the code to follow to find the optimal solutions to code errors and logic errors, this will also be used to find and point out lack of security and optimizations.
- study code used to find errors in depth, doing this will allow for the experimental code to have a solid foundation and make sure already preformed research is not repeated wasting both time and effort.
- make the first version of the code to be improved upon, the first version of the code will need to be basic with all flaws known to test the program later on the code will be tested with unknown code errors
- to make sure the output is correct before collecting data, this is to check the output is correct for the known code errors to see if the code is working correctly.
- analyse the collected data and start to improve the code, this is the step for making sure the correct answers are outputted for the known errors and then improving any mistakes until it works correctly and then checking with unknown code to make sure there are no errors.

- test project and collect data repeat, this is the stage of repeatedly testing to make sure the code always works and produces the correct answer and that there are no false negatives or positives.

Part II

**literature review and
research**

Chapter 5

introduction

the literature review is all about finding papers with similar research and topics to use as a foundation for the project, this can be used for parts of the project or types of algorithms needed for the calculations, this will be used for the project to make sure scientific basis are followed, it is best to have references to the best in the industry for the project you are studying and see if they have papers agreeing with your ideas or methodologies.

Chapter 6

literature methodology

the literature methodology is all about what will be included in the project, how it will be found, from what sources and how it can be validated as accurate and useful. what restrictions will be imposed on the gathered literature to make sure it remains useful and accurate, for example only papers from the year 1980s and up, this is done to make sure papers that are being referenced have not already had the problems they found solved, for example, ECC memory after a particle changed a bit nearly causing a plane to drop out of the sky, this is also done to make sure the papers used in the project are not outdated and wrong in past assumptions.

there are keywords that will be searched for during this project they are:

- Deterministic rule set
- algorithms
- optimization
- optimization flags
- Regex
- human coding errors
- compiler
- code automation
- code reliability
- code reliability

the keywords highlighted above will be used in search engines to find papers relevant to the project.

the search engines found below will be used for looking for the relevant papers

- Google search
- UWTSD library
- Academic Search Premier (EBSCOhost)
- ACM Digital Library
- IEEE Xplore digital library
- ProQuest Central
- NewsBank
- MarketLine Advantage

the types of sources allowed for this project will be

- Peer reviewed articles
- Peer reviewed journals
- Peer reviewed research
- Peer reviewed websites

there will also be an exclusion criteria and inclusion criteria for the papers. the exclusion criteria will be material older than the 1980's as that means it will be less relevant, papers without any of the keywords, research that is hard to prove using mathematical proof, vague and abstract descriptions making it difficult to code, program and test.

the inclusion criteria will be material newer than the 1980's, preferably having the keywords in title or abstract, peer reviewed, research that can be proven mathematically will be preferred. detailed papers no abstract and vague statements, can work or be used in programming languages, is referenced by other peer-reviewed work.

Chapter 7

Regex

regex is normally used in search patterns using a very specific combination, for example the regex used to make sure a credit card number is valid would look like `[0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9] [0-9]` this makes sure that only numbers are entered in reality the `[0-9]` would be smaller on some digits for personal shopping as some numbers would only be used for companies. experts use this for bank checks, finding information on webpages and finding words and replacing them in all parts of code at once. regex can be very useful when used correctly. in conclusion this will be used to check end lines are placed correctly, as there is no way to tell how many characters are in a line before the end line and this will have to be found out when the code is read in by the program as such a regex that allows any amount of numbers and letters then checks for a proper end line for that programming language will see if the end lines are in place correctly.

Chapter 8

optimization flags

optimization flags are used by compilers to make programs run more efficiently and smoothly, some programs put this to great use whilst others do not, for example an advanced super computer would want optimization for all its programs to get results quicker and using less resources for memory and more for calculations, were as modern video games are rarely optimized any-more this is due to modern hardware having the extra memory and ram to run the games unoptimized were as old games needed all the memory and ram plus limits on load size to run proficiently, (Fact Fiend, 14 Dec 2020) for example the original silent hill used fog to hide the render distance limit. in conclusion optimization flags make things run more efficiently and smoothly when used correctly.

when optimization flags are enabled the compiler tries to improve the code and run time performance it does this by taking longer time to compile the code into a running executable, without optimisation flags the compiler only makes the runtime smaller with optimisation flags it looks at the code overall (this means everything is compiled at once not statements at a time) and tries to make it more efficient, this can have unexpected consequences, for example, braking code that worked unoptimised, (Christophe Dubach, Timothy M. Jones, Edwin V. Bonilla, Grigori Fursin, and Michael F. P. O'Boyle. 2009.) as such all optimization needs careful testing on each type of processor as all processor models are slightly different, when optimization flags are enabled due to the way code can change it makes it much harder to find and debug errors in the finished compiled program. in conclusion optimization flags save memory and ram but make it more difficult to debug the finished program.

when using optimization flags it can be very helpful to be one generation of processor behind, this is due to the time it takes to update the optimization flags to be perfected for the processor, usually when the last generation has been finished the new generation is only just getting started, this compiled with the rate at witch microprocessors are released, means the last generation of

optimisation has just been finished and the new processor is out and needing to change the optimisation flags again, this makes it difficult to keep up with optimisation for the newer generation of microprocessors. (Christophe Dubach, Timothy M. Jones, Edwin V. Bonilla, Grigori Fursin, and Michael F. P. O’Boyle. 2009.) optimization changes between all processors and needs to be redesigned, this is a time consuming task that is difficult to preform due to the many slight changes that need to be preformed and large amounts of testing to make sure the optimisation is the best option possible.

8.1 different types of optimization flags

there are lots of different optimization flag options that have many different effects, for example:

1. streamlined
the goal of this optimization is to make the code run as fast as possible
2. condensed
the goal of this optimization is to make the code size as small as possible whilst still maintaining all functionality.
3. power saving
when this type of optimization is running the goal is to have maximum performance for minimum power, the algorithm is designed to use the least amount of power possible to complete all goals. this is sometimes called Energy efficiency
4. performance increase
the goal of this optimization is to get the best performance out of all processors no matter the energy consumed, in this mode the best performance is aimed for without damaging the computer.

the above examples are possible ways of using optimization there are lots more, code flagging is a type of flagging were errors are found and pointed out by intellisense when compiling, this finds normal types of errors not logical errors but combining it with algorithms it should be possible to also find logical errors, optimization flags normally go through one line at a time optimizing them and making sure that during the process all matching variables are given the same name/memory location by changing how this is done it should be possible to read, check and either suggest changes or change code if needed to make it work. in conclusion it should be possible to use code related to flagging methods to find errors in the code.

8.2 compiler optimization effects on code reliability

compiler optimization can have both good and bad effects, this will depend on how well designed the optimization program is for the processor badly designed or ones with low efficiency have a chance of messing up due to optimization breaking loops or in a sequence it was never intended to run through, examples of this are two local variables being made global or a global variable not working for the intended functions correctly any more, but good optimization can make very intense code run much quicker and smoother than the unoptimized code the only problem with this is finding and removing bugs as optimization makes it slightly harder as mentioned previously, for example, (ASCE Hojjat Adeli, 1995) most supercomputers use optimization at some level. in conclusion optimization must be used carefully in the project for finding errors and carrying out optimization or it is likely to make the problem worse.

8.3 compiler optimization automation

it should be possible to automate the program looking for errors and optimization, for example, a lot of labs are starting to use AI or automated programs to repeat repetitive tasks for greater efficiency and accuracy as humans get tired and make mistakes where as machines do not, experts have performed tests on compiler optimization to see if it can be automated the finished experiment proved successful, (Stefano Cereda, Gianluca Palermo, Paolo Cremonesi, and Stefano Doni. (2020)) "The proposed approach has been validated using two widely used benchmark suites, cBench and PolyBench, including 54 different programs. Our solution, on average, extracted 90% of the available performance improvement 10 iterations before current state-of-the-art solutions, which corresponds to 40% fewer compilations and performance tests to perform." in conclusion it should be possible to automate the task due to the similarity between the projects program and optimization flags.

Chapter 9

algorithms

when algorithms are brought together it becomes possible to make code do things that would take insane amounts of code otherwise, for example a maze without algorithms would be insanely difficult if not impossible this would be due to all the dead ends and multiple paths, however using backtracking and an algorithm to pick a path the maze becomes much similar as the back tracking takes the locator back to the multiple paths point to choose a different path when hitting dead ends, an expert example of how algorithms come together to solve problems by experts (Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, Clifford Stein, (2009)) is "The Human Genome Project has made great progress toward the goals of identifying all the 100,000 genes in human DNA, determining the sequences of the 3 billion chemical base pairs that make up human DNA, storing this information in databases, and developing tools for data analysis. Each of these steps requires sophisticated algorithms." in conclusion algorithms will be key for the project to be successful for all programming languages and to find all errors, different types of algorithms can be found below.

9.1 rapidly exploring random tree

rapidly exploring random tree (RRT) is an algorithm that works by making two branching tree's and finding the shortest path between them, this is repeated many times until all tree's are linked and the best shortest path for the problem is found. this takes a long time to compute so extra parameters are normally used to lesson the number of computations needed. RRTs are normally used to map out movement paths these are mapped to graphs, for example a movement path for machines or animation, a statement from experts researching RRT explains that currently very little of their potential has been tapped (LaValle, S. M. (1998))"At the present time, we believe we have barely scratched the surface of potential applications of RRTs." experts also believe it can be used for simulating cars, plains and trains to test out how the finished product would

react to unexpected accidents like landslides, crashes and more. "Researchers In computer graphics, dynamical motions of simulated machines and digital actors can be automated." in conclusion using RRT it should be possible to find all errors and provide possible solutions.

9.2 Recursion

recursive algorithms work until a preset condition is met, for example until $x = 26$ do, this is normally done through having x be incremented(increased by 1) every time the loop goes round and the condition checked for before looping again, in conclusion recursive loops can be used for both finding problems and solutions depending on how it is coded into the project.

9.3 Logical

logical algorithms works by applying logic to the parameters fed into them, for example given variable A and statements about variable A then reach the answer/goal. this type of algorithm is used by programmers that need very specific functions that without the statements would get strange answers e.g people doing simulation might use something in this to test gravity etc with fixed perimeters accurate to the real world, an example from the Stanford university website being (Stanford university (2021))"1. All men are mortal. 2. Socrates is a man. 3. Therefore, Socrates is mortal." in conclusion the logical algorithm has restrictions on the parameters fed into it as such there are limitations, on the other hand with the correct parameters it can be used for finding errors in the code.

9.4 Deterministic or Non-Deterministic

1. Deterministic algorithms work step by step to solve problems with perfect precision at every stage, examples of this are code going through long formulas or a normal compiler, the same input will always get the same output, experts use these in normal code and calculations, in conclusion deterministic algorithms will be used in the code to make sure everything goes in the correct order.
2. Non-Deterministic algorithms use guesswork to get roughly the correct answers the variables inputted into the sum are different each-time, this is used when a good approximation will do, for example numbers to the power of 1000 etc, experts use this when calculating events with very small probabilities and when simulating reality to get a roughly accurate simulation compared to the real world, in conclusion this will be used

for finding roughly the best answers in the project and they will then be narrowed down from there.

9.5 Exact or approximate

Exact or Approximate are two different types of algorithms one works for working out exact mathematical formulas with only one possible answer, for example $2 + 2 = 4$, Approximate is exactly the opposite it is designed for questions with multiple answers where all are viable and the goal is to find the best answer, for example the knapsack problem the goal of a thief is to get the maximum money for his time there are 7 items of different value and weight, the thief can only carry weight of X, what is the best options to take considering weight and value. experts put this into use for finding solutions to mathematical problems with many different answers along with logical rules to get answers to complex mathematical problems, in conclusion the project will need a mix of both for finding answers to problems and finding the problems themselves.

9.6 Divide and conquer

divide and conquer algorithms splits a large problem up into lots of smaller instances of the same problem, this repeats until all instances are small enough to calculate the solution easily, an example of this is merge sorting data is sorted into segments and organised then it is re-merged once finished. experts use this for sorting, calculations and disk clean-up where the chunks are made tiny to move, in conclusion this can be useful for the project for speeding up the errors and corrections calculations.

9.7 Back tracking

Back tracking algorithms are used when there are multiple solutions to an answer that need testing, for example a maze in computer code would use backtracking to get back to the junctions from dead ends, this normally works incrementally as the test goes on each time the answer is wrong it goes back one or multiple save points as needed and tests a different route from there, experts use this for things like mazes, rubix cube, solve 9 etc, this is similar to saves in video games, in conclusion this will be used in the project for all wrong solutions to errors and a new solution will be tested.

Chapter 10

ECC Memory

error correction code is used to prevent bit flips(single event upset) this is when a particle entering the atmosphere from space collides with other molecules roughly 15.5 miles above the ground and breaks up into even smaller molecules, these continue colliding which continues making more particles heading down to the earth, when one of these particles hit the correct area of the computer, it can swap a bit causing damage to calculations and software it is not harmful permanently to the computer though as the wrong bit can be deleted and re-corrected via software known as error correction code (ECC). without this protection bit flips happen once a month for every 256mb of ram according to (J. F. Ziegler,(1996)) IBM, the higher up you go from sea level the more likely you will have problems due to particles and neutrons causing bit flips. an example of this is a bit flip causing a plane to drop 200 meters in 20 seconds people in the plane would be experiencing 3g it then dropped a further 120 meters this even happened on October 7th 2008, this happened because it was an old plane without single upset event protection. (Veritasium (2021)) experts at NASA use ECC memory to protect shuttles and rovers when they are sent into outer space. in conclusion this is important for the project as a check will need to be in place to make sure the code does not find good code as an error due to a bit flip.

error prevention hardware is when a system uses multiple sources of data where the majority answer is taken as the correct answer if one computer provides a different answer, this is used in important systems like nuclear plants were a single error could be catastrophic,(Veritasium (2021)) for example NASA uses four computers running the same software in most rockets and if one provides a different answer the other 3 overrule the different answer normally caused by the bit flipped computer. in conclusion this is one of the ways the projects program could be protected from errors by having multiple instances running at once and seeing if they all come to the same answer.

these are two different methods of ECC memory one for hardware and one

for software, these can be programmed to work in tandem to make sure there are minimal errors for important software and hardware making sure programs are outputting the correct answers, this will be used in the project to make sure the code is reliable and trustworthy.

Chapter 11

deterministic rule set

a deterministic rule set works by settings rules that must be followed by the code, many rules come together to make the final rule set, due to the fact that problems are solved step by step it is more likely to reach the correct answer but will take longer to compute compared to other solutions. an example of this would be a game puzzle being solved by an AI, there has been research by scholars (Berndt, M.,Fischer, C.,Krummacker,D(2013)) to see if under strict conditions if it is possible for deterministic rule sets to be used for making truck schedules to move between warehouses to make sure routes are optimised to have less traffic collision possibilities. another expert example is deterministic research is currently being paired with quantum computing, to try and model a quantum network. using mathematical models and deterministic models, if this is successful then it says a lot for the potential of deterministic nature for modelling and predicting results.

using a deterministic rule set it will be possible to go through all choices found by the rapidly exploring branching tree to find the one with the best fit for the solution, using the deterministic rule set the answer with the highest probability of working can be chosen, if the choice made does not work for any reason back tracking will allow for going back and finding a different solution until either a correct solution is found or all possible solutions have been tested. deterministic rule sets do have problems though do to the large amount of calculations they will go through checking all possible answers, they will take longer to calculate compared to other algorithms, the upside of this extended calculation time is the answers are more likely to be correct as long as the rule set is made correctly. in conclusion deterministic rule sets can be used in the project to make sure all solutions are possible before they are tested individually.

Part III

research methodology

Chapter 12

introduction

research methodology is all about collecting data, collating data (grouping), analysing the collected data and how the research shall be conducted. an example of the main types of data is qualitative data (words) and quantitative data (mathematical), an example of this would be how much pop is in a bottle, how much liquid fills a cup, this data can be repeatedly tested on different days and normally will come out the same, qualitative data is different for example the colour of cars in a car park the results will be different when you retest if either a single car has left or its a different day, as such qualitative data is much harder to prove, compared to quantitative data that can be retested many times with very similar results, as long as the test is done correctly and accurately. this is a great advantage when working with things that can be analysed mathematically computer programs are normally easier to analyse the results if they are in mathematical format another great reason for using quantitative data is that explain the test results will be much easier using numerical format. (Neuman, W. L., Wiegand, B. (2000)) in conclusion The planned research methodology is to look for quantitative data and analyse it to test the research. theory.

Chapter 13

methodology

the project currently has two different methods for checking the code please see below:

- the first solution to checking the code is to read the code in from a document and have a group of algorithms look through it line by line, when it comes to loops the algorithms will first look to make sure all loops have the correct start and end, then it will check the code inside the loop to make sure it works, then try for optimization, the downsides of this method is it will take longer time wise and more computational power. in conclusion this method will work best for lines and might produce problems when optimizing loops if variable names are reused or if it relates to global variables from other functions.
- the second solution for checking the code is to turn the code into an image and scan it using algorithms to make sure all lines end correctly, that all loops start and end correctly, the downside of this method will be checking the logical part of the code, this method would work better for seeing if loops begin and end correctly and faster for checking lines end correctly as it will only be looking for specific pattern pixels in all lines with code. in conclusion this method would be quicker for some parts of the code but it would also be much harder to check the logic using this method, this method would also struggle with multi programming languages code.

for the program to work accurately and complete the project goals it will need to be a hybrid method of both, this should reduce errors in the code and make it much more accurate for finding simplistic errors.

even with this the program could encounter errors when programmers put more than 1 section of code on a line and if any of the code is encrypted either intentionally or due to using a special compiler. this will be difficult to work around and as such will not be solved during this project due to the complica-

tions it would lead to in making sure code is working correctly and logically.

in conclusion the hybrid method will be used for the project.

Chapter 14

research overview

research overview is important for the project as it lays out the initial questions to make sure research stays on track, the initial questions are very important for the project as they make sure the researcher keeps on top of the subject matter and does not get side tracked. this means the research stays relevant throughout the project until it is completed.

for example some questions to be asked for research are: Is this possible?, if so could it be done in the real world without super strict(LAB) conditions?, what impact would it have on the future ?, what improvements would it make for programmers ?

if the project is completed successfully it should not only improve memory usage in computers and reduce memory wastage, it will also reduce the number of errors in software/hardware and improve an area of security that is massively lacking, this can be seen from how often bugs and glitches are leaking user-information online. programmers would have much less to worry about bugs would not be a major issues as there would be potential solutions, optimization would be done automatically saving time and effort in having to improve them and test one by one, finally security would not be missing from the system due to a programmer forgetting or being lazy.

in conclusion the research overview is the topics entailed, if it is possible, can it be done outside a lab and finally the impact it will have on the world if successful.

14.1 research process

the goal of this project is to test if deterministic rule-sets and algorithms can work together to find complex coding errors. to complete this task a research method and process must be laid out in advance and followed until completion

of the project. for example this projects research process will lay out, gathering data, research, testing, collation of data and displaying of results.

there are two main ways to go about this research inductive or deductive

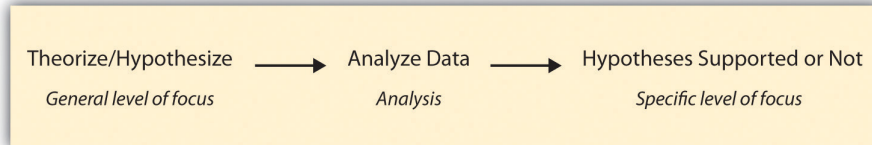
Inductive: inductive research is all about collecting large amounts of data, then looking for patterns in the data (analysing it) then developing a theory to explain the patterns in the data.

Figure 1 inductive research (Saylor Academy (2012))



Deductive: deductive research is the inverse of inductive research, starting with a theory that interests the scientists/researchers they will read up on the theory analysing as they go along, and then test all the hypothesis (data) to see if their theory is correct or not.

Figure 2 deductive research (Saylor Academy (2012))



The reason for choosing deductive research is due to the lacking data online and not having a similar program to test with the data will only be collected for analysis once the program has been coded and tested numerous times. as such there is a theory to be tested to see if it can work or not. in conclusion this was the only reasonable choice for a research process due to all factors combined previously stated meaning there was no large data easily available to gather.

14.2 research philosophy

there are two main research philosophies quantitative vs qualitative, their definitions can be found below:

Quantitative research: data in number form this is mainly done through statistics this relates to how statistics are displayed, analysed and interpreted. If data is presented wrongly it can become a false positive or false negative. sometimes this happens accidentally,(TED-ed (2021) 1+2) some researchers do

it intentionally this is called P-hacking.

Qualitative research: deals in data about people and the way they behave in certain locations and how they communicate, this information is normally gained by watching people or by doing face to face interviews [1] this is normally data that is much harder to explain with numbers.

in conclusion this project will use quantitative research as it works best with code and analysis, when numbers can be used to make graphs and it can simplify the display of results, this makes it easier to test, improve and collect data when working with coding projects. this will make collection of results and analysing of results much quicker compared to qualitative.

14.3 research methodologies

research methodologies are put forward to make sure the results of the research are accurate and true, for example to make sure P-hacking does not happen accidentally there needs to be measures in place to make sure the data is analysed fairly and objectively, not with a goal in mind as if you stop analysing once the goal is reached you might miss the turning point that disproves the project entirely due to not finishing the analysis on all the gathered results. as such all good research must have a methodology put forward before it is carried out this ensures all data is gathered and analysed fairly and that the results from this gathered data will be accurate and true. there are a multitude of different ways of carrying out the research to find the final results that will confirm if the research is correct or not, a few that are relevant to this project can be found below:

(alzheimer-europe(Friday 21 August 2009)

experiment: in this type of study the research takes part in changing variables and collecting the newly generated data. an example of this is mixing certain chemicals will get result x or getting groups of people to do specific actions and seeing if there are any differences.

development: in this type of study software is created and tested to see if it is up to the requirements of the project or goal. this type of methodology needs the researcher to develop software to test.

in this project the software will need to be built on a PC, the software of: virtual studios, notepad++; will both be required one for test code and the other for creating the project itself.

this method of developing the project and then testing was selected due to a lack of available resources and knowledge online, the two types of methodologies chosen were for the purpose of making the project and making sure it worked

correctly.

the advantages of the software if successful will be: improved security features, less bugs and errors in computer software and lastly programs running more smoothly using less memory.

the disadvantages of the software will be: the code will be difficult to make work correctly for all programming languages, rules might need to be put in place to make sure it works correctly as some programmers use one lines of code with more than one end line it might be difficult to check if code is working correctly and logically if that is implemented in code to be tested.

in conclusion this project will be development based.

14.4 data collection

data collection is all about finding and keeping track of data useful for the project, an example being in an experiment writing down variables, for example the temperature, the volumes of chemicals added and the times they were added, the more accurate the variables, the more easy it will be to do the exact same experiment again, this is done to make sure things are scientific and can be later used for mass production of the item in question, without the variables and formulas it would become much harder to replicate the experiment and future product.

there are many different types of data collection this project will involve the following:

multi-method: this is where multiple sources of data are used, compared to mono method were only one type of data collection method is used. multi-method comes with the added bonus of being more reliable as it is harder to get the wrong conclusion with many different sources of data, showing it is either a correct solution or many different sources all pointing out it is wrong.

Archival records: this is looking through past records and work, to gather needed data and evidence for the experiment or project, this is done to make a good foundation for the project if similar work has been done before or to point out if something will not work, if it has been tested previously and always failed. this allows for making sure only accurate data is used in the project and it does not start off on a dead end path. an example of this would be google scholar and the papers that can be found there.

Documentation: is recording of all evidence of inspiration from seeing something, ideas used from others, information from the past, the present, this should be used for all sources, to make sure everything was documented, all videos, pa-

pers and searches used were saved in a separate folder called "masters project for 2021", links were also copied along with the authors name and page number and pasted into the document under the related chapter with a tag called "further research and reading" this was kept in the document until the final version.

to prevent errors in the project and the data collection during research for the project, each section was a separate word document this was done due to errors in the past with word documents being too large and loosing work, to prevent this everything was split into separate sections on word documents, for example, introduction, literature review each has a separate word document, this included references, paragraphs and its own outlined plan these were then copied into the word document for references and the finished project in order and the complete project was assembled, as an extra precaution and to make it look nicer this finished project was then moved over to TeXstudio to make a PDF final version, this was to prevent repeat of errors in the past of word documents loosing thousands of words even using print pdf and destroying finished coursework before submission. to make sure this did not happen to any research or the code itself two copies of all documents were kept including the code.

to make sure all tests are accurate and the program is working as intended, before each test the code to be tested along with the version number of the current software will be recorded, all results will be added to the table for that particular test with that particular software version, if anything throws off the numbers it will be noted why, for example a bug in the system. the table will record the number of problems found and the number of times scanned for all versions and all code tested. a grand total of both errors found and number of errors in the program will be used to work out the overall efficiency of each version of the software.

the data of each test will be analysed along with the table for each version of the application, each test will be run multiple times with different code inputs, the inputted code will then be checked and results outputted, the outputted results will then have to be checked to make sure there are no false positives or negatives that will skew results. if for some reason errors are not found or false errors are found they will be noted down in the table and an investigation into why launched later.

in conclusion the data that will be collected are: the name of the code tested, the number of faults found, the number of bugs in the code being tested, number of times scanned to find faults, version of the application scanning the code and lastly the outputted results from the system from each version. this will all be collated and made in to a table and graph to compare the final results.

Chapter 15

research design

this stage is all about making plans for the research, analysis and conclusions, the reason for this is making sure the project is carried out fairly, will be analysed with no bias and the conclusions reached are reasonable. for example some studies intentionally skew results by giving up or ignoring a set of results to get the result they set out to find, others give up analysis half way through the test once the results they want have been reached to prevent this a good research design must be in place.

15.1 research deliverables

the research deliverables are all the things that will come out of the project, for example in this project if it succeeds completely the research deliverables would be: the program for checking code works correctly for one programming language, code that can find security weaknesses, code that can preform optimization and check it works correctly, a new way of using deterministic rule sets to find errors and solutions, of course there will also be the test results and graph displaying how successful the finished project was.

Chapter 16

triangulation

triangulation is the art of using multiple sources of data analysis to verify the conclusion reached, this is done to make sure all conclusions point to the same answer and that there are no biased thoughts in the conclusion reached, an example of this in real life would be police using triangulation to work out where a mobile phone is, the reason for triangulation is to make sure the answer was not reached using p-hacking, this is where the analysis of the results stops when the goal is reached even if the entire data collected has not been completely analysed and would show the answer is not true or very unlikely, some people do this intentionally by removing part of the results to get the answer they want so triangulation is done to lower the risk of accidental p-hacking and to improve the reliability and accuracy of the project tests. in conclusion this is a critical step for the project in proving its validity and accuracy and that no bias has made its way into the project.

Chapter 17

conclusion

the conclusion is taking all gathered results together extracting the essence that has been analysed and finding out the truth of the matter, this will be used to show if the project was: partially successful, a failure or a complete success. the things used to determine this will be: the test results, the analysis, the graphs, the original goal of the project and whether it has been completely achieved or not. this will be done by analysing each result separately and then bringing them together into one large conclusion, it makes sure all tests are carried out fairly with no bias and all conclusions reached are true and accurate to the results given, this is done to make sure numbers are not drawn at random or false positives or false negatives make it into the results of the project destroying the accuracy of the tests.

Part IV

design and implementation

Chapter 18

introduction

this chapter is all about design choices and how they will effect the project, for example using one coding language over the other due to it being easier to test later to prove the theory of the project, another example would be using a coding language that is more familiar when developing the code to make it slightly easier, this will then talk about how they were implemented to create the needed software for the project. this will cover why one thing was chosen over another, for example c++ over python, or a pc over a laptop, and how these choices impacted the project overall. this is done to explain why choices were made for one thing over another, when the second item might of been more useful for limited circumstances.

for this project in particular a PC will be used over a laptop for better reliability and no worries about dying batteries, hard drive malfunctions and motherboards playing up. all work will be backed up onto a separate USB stick to make sure if the computer has errors it can still be saved and later recovered on a new device to continue being worked on . this project will be made using Tex-studio due to past errors of Microsoft word loosing thousands of words and destroying entire documents close to the hand in date, visual studio and notepad++ will both also be used for this project visual studio for making the main part of the project where to complete this test a new software must be engineered, this new software will be used for checking for errors, finding code with lacking security features and lastly providing optimization and checks, whereas notepad++ will be used for the code that will be tested, this is to make they are on different systems before the test and there is no chance of them being used in tests before the code is ready to be tested.

Chapter 19

problem formulation

problem formulation is all about how to lay out the problem in the best way to find answers, this means what is needed to get to the answer, how it will be eventually answered, and the steps in between.

the main problem of this project is can a deterministic rule set find errors. then there are the secondary goals of security, optimization and providing and testing answers automatically.

to test this many different problems were encountered, for example:

- the program needed to be created.
- lack of understanding of deterministic rule sets
- lack of understanding of the best ways to find errors in code.
- lack of knowledge of the best choices for optimising code
- lack of code with similar purposes to research

for this program to be successful lots of re-learning of c++ will be needed due to changes made in how some of it works.

to get answers to these problems and complete the project many different tasks needed to be completed

- do a literature review and find material.
- find learning material for knowledge blind spots.
- practice code more and read on ways of finding errors.
- find expert advice or material and get inspiration.

- find code that scans through anything and work out if it can be adopted to help the program.

for this project to be considered a complete success, the program must find errors, point out lacking security features and recommending solutions or possible optimization for a partial success it needs to either find errors and security flaws, whilst ignoring the secondary goals of recommending solutions or possible optimization.

Chapter 20

design

for the design of the project inspiration was found from the test made by "Stuff Made Here" of a robot wielding a chainsaw to make a bear carving, the inspiration it brought forth was the idea of using a rapidly exploring random tree to go through all possible solutions and find the best fit, as shown above in the literature review it takes longer to calculate the answer but gives the answer with the highest possible fit. this along with a deterministic rule set will allow for faster calculations of the method to be tested if for some reason the best fit answer does not work with the rest of the code, then the Back tracking algorithm will start working and a new possible solution can be tested.

the deterministic rule set will be designed to weed out impossible answers as well as weak answers, these weak answers will be solutions that do not work very well like 20 lines of code instead of a loop, another feature of the deterministic rule set will be for finding the mistakes in the code, as shown in the literature review when used correctly deterministic rule sets can be set up to automatically manage searching for best possible answers to questions input. so by having a proper rule set in place it should be possible to work out all problems using both algorithms for reading in the data, the only problems with reading in the data might be encoded data or where all data is on one line.

Chapter 21

implementation

the image below shows the planned implementation for the project after the literature review, design and implementation have been finished and a brief overview made for the project.

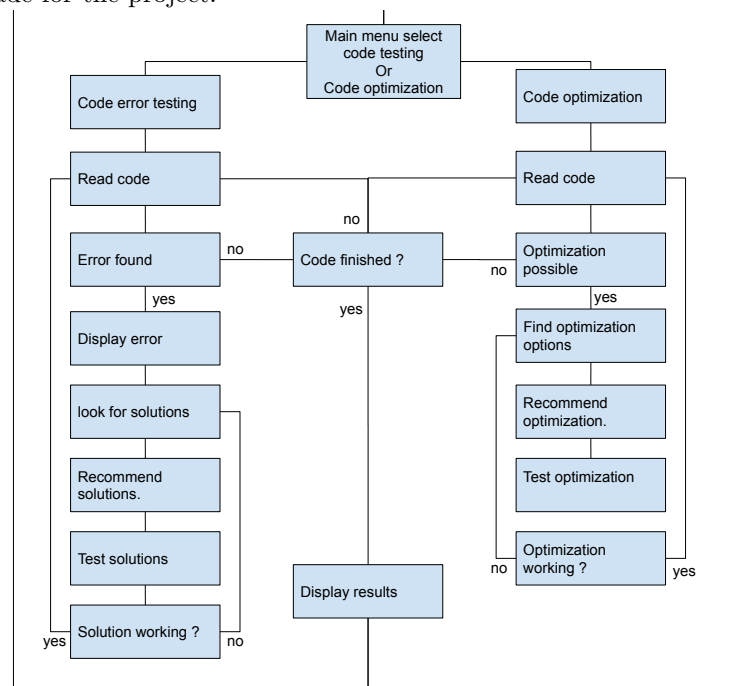


figure 3

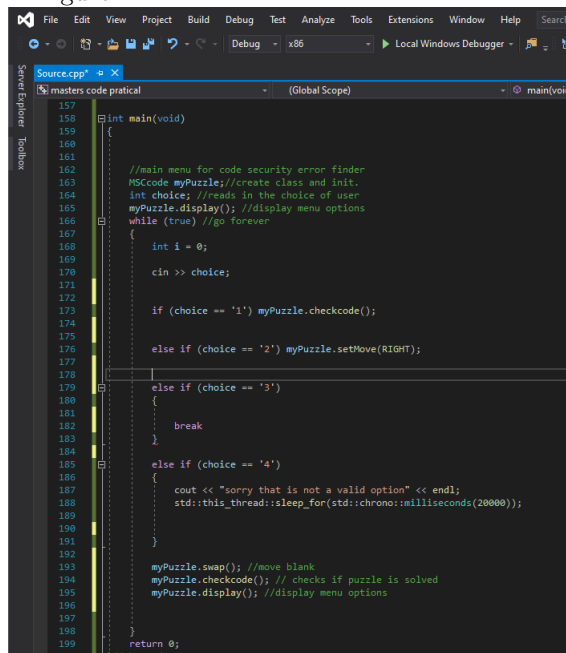
each stage in the diagram corresponds to an action in the code, the first choice in the program is to select one of the two option either: to remove errors option 1 or to conduct optimization option 2.

the program will default recommend option 1 first, to make sure there are no bugs in the program to be optimized and everything is working as intended.

the first option in the code the program will start with read code, until an error is found or the code ends, the code will then proceed to look for possible solutions, once all solutions are found it will start with the solution with the highest chance of working and test it, if the solution does not work it will repeat the above process, if it works it will continue by checking is this the end of the code if not continue reading the code until an error is found or the code ends.

the second option this time around knowing there are no errors and the code does as it is meant to optimization will be carried out on all possible parts of the code, the newly optimized code will be tested if it works as intended, it will go on to check if its at the end of the code or if there is more code to read, if for some reason the optimization did not work it will continue trying until a optimization works or all options have been tested and go back to the unoptimized code using back tracking.

the implementation of the project had many difficulties at times such as: the menu for the project refused to work correctly this can be seen below, figure 4



```

157
158 int main(void)
159 {
160
161     //main menu for code security error finder
162     MSGcode myPuzzle; //create class and init.
163     int choice; //reads in the choice of user
164     myPuzzle.display(); //display menu options
165     while (true) //go forever
166     {
167         int i = 0;
168
169         cin >> choice;
170
171         if (choice == '1') myPuzzle.checkcode();
172
173         else if (choice == '2') myPuzzle.setMove(RIGHT);
174
175         else if (choice == '3')
176         {
177             break
178         }
179
180         else if (choice == '4')
181         {
182             cout << "sorry that is not a valid option" << endl;
183             std::this_thread::sleep_for(std::chrono::milliseconds(20000));
184         }
185
186         myPuzzle.swap(); //move blank
187         myPuzzle.checkcode(); // checks if puzzle is solved
188         myPuzzle.display(); //display menu options
189
190     }
191     return 0;
192 }
193
194
195
196
197
198
199
200

```

as such the project had to be altered such that it now runs in sequence. starting with error finding.

there were difficulties getting a loop to check for: all end lines,all open and close brackets for code, the code would successfully count the number of lines in the test code but not the end lines or brackets

figure 5

```

35 int opens = 0;
36 // creates a variable called opens used for checking how many opening loops there are
37 int closes = 0;
38 // creates a variable called closes used for checking how many closing loops there are
39 // these two variables can later be compared to check if loops are opening and closing correctly.
40 int lines = 0;
41 // creates a variable called opens used for checking how many opening loops there are
42 int endlines = 0;
43
44 string myText;
45 // creates a variable called myText to store and output the code being read in for testing
46
47 // Read the testcode file
48 ifstream myReadFile("testcode.txt");
49
50 // Use a while loop together with the getline() function to read the file line by line
51 while (getline(myReadFile, myText)) {
52
53
54
55     if (myText == "\n")
56     {
57         endlines++; // counts the number of end lines
58     }
59     else if (myText == "{")
60     {
61         opens++; // counts how many code brackets were opened
62     }
63     else if (myText == "}")
64     {
65         closes++; // counts how many code brackets were closed
66     }
67
68     // Output the text from the file and end each line as appropriate to make sure the output is correct
69     cout << myText << endl;
70
71
72     lines++; // counts the number of total lines in the code
73
74
75
76
77
78     cout << "number of lines " << lines << endl;
79     cout << "number of end lines " << endlines << endl;
80     cout << "number of code opened " << opens << endl;
81     cout << "number of code closed " << closes << endl;
82
83     // Close the file
84     myReadFile.close();
85     std::this_thread::sleep_for(std::chrono::milliseconds(200000));
86
87     return 0;
88 }

```

```

1 // University/postgraduate/masters project/masters code practical/Debug/masters code practical.exe
2
3 please press 1 for removal of code errors (Default recommended)
4 please press 2 for code optimization
5 please press 3 to exit the program
6
7 please choose a valid option
8
9 #include <iostream>
10 using namespace std;
11
12 int main() {
13     int i = -1;
14     do {
15         i++;
16         cout << i << "\n";
17     } while (i < 50);
18
19     cout << i << "\n"
20
21
22     int i = -1;
23     while (i < 5) {
24         i++;
25         cout << i << "\n";
26     }
27
28     return 0;
29
30     while (true) //go forever
31     {
32
33
34 }
35
36 number of lines 33
37 number of end lines 0
38 number of code opened 0
39 number of code closed 1

```

to fix this error in the code the string had to be pulled apart into characters and recognised separately, this needed to be done using ASCII as they were special programming characters

figure 6

```

34 bool HSCoder::checkCode(void)
35 {
36     int opens = 0;
37     // creates a variable called opens used for checking how many opening loops there are
38     int closes = 0;
39     // creates a variable called closes used for checking how many closing loops there are
40     // these two variables can later be compared to check if loops are opening and closing correctly.
41     int lines = 0;
42     // creates a variable called opens used for checking how many opening loops there are
43     int endlines = 0;
44
45     string myText;
46     // creates a variable called myText to store and output the code being read in for testing
47
48     // Read the testcode file
49     ifstream myReadFile("testcode.txt");
50
51     // Use a while loop together with the getline() function to read the file line by line
52     while (getline(myReadFile, myText)) {
53
54
55
56         std::string s(myText);
57
58         for (char& c : s)
59         {
60             // std::cout << "One character: " << c << "\n";
61             // the above code was for displaying the characters during testing
62             if (c == "\n")
63             {
64                 endlines++; // counts the number of end lines
65             }
66             // 123 is the character code for {
67             else if (c == 123)
68             {
69                 opens++; // counts how many code brackets were opened
70             }
71             // 125 is the character code for }
72             else if (c == 125)
73             {
74                 closes++; // counts how many code brackets were closed
75             }
76
77             c = "";
78
79         }
80
81         // Output the text from the file and end each line as appropriate to make sure the output is correct
82         cout << myText << endl;
83
84
85         lines++; // counts the number of total lines in the code
86
87
88
89
90
91     cout << "number of code brackets opened " << opens << endl;
92     cout << "number of code brackets closed " << closes << endl;
93     cout << "number of end lines " << endlines << endl;
94     cout << "total number of lines in tested code " << lines << endl;
95
96     // Close the file

```

```

1 // University/postgraduate/masters project/masters code practical/Debug/masters code practical.exe
2
3 the program will now start looking for errors please wait
4
5 #include <iostream>
6 using namespace std;
7
8 int main() {
9     int i = -1;
10    do {
11        i++;
12        cout << i << "\n";
13    } while (i < 50);
14
15    cout << i << "\n"
16
17
18    int i = -1;
19    while (i < 5) {
20        i++;
21        cout << i << "\n";
22    }
23
24    return 0;
25
26    while (true) //go forever
27    {
28
29
30 }
31
32 number of code brackets opened 4
33 number of code brackets closed 4
34 number of end lines 4
35 total number of lines in tested code 33

```

it had now become possible to check that all loops opened and closed correctly, as well as look for future problems.

figure 7

E:\university\postgraduate\masters project\masters code	E:\university\postgraduate\masters project\masters code	E:\university\postgraduate\masters project\masters code practical
<pre> } while (i < 50); cout << i << "\n" int i = -1; while (i < 5) { i++; cout << i << "\n"; } return 0; while (true) //go forever { } } </pre> number of code brackets opened 4 number of code brackets closed 4 number of end lines 4 total number of lines in tested code 33 all loops are working correctly	<pre> } while (i < 50); cout << i << "\n" int i = -1; while (i < 5) { i++; cout << i << "\n"; } return 0; while (true) //go forever { } } </pre> number of code brackets opened 4 number of code brackets closed 3 number of end lines 4 total number of lines in tested code 33 error found in loop closing	<pre> } while (i < 50); cout << i << "\n" int i = -1; while (i < 5) i++; cout << i << "\n"; } return 0; while (true) //go forever { } } </pre> number of code brackets opened 3 number of code brackets closed 4 number of end lines 3 total number of lines in tested code 33 error found in loop opening

the above is a demonstration that the code works as intended and can even point out which has not been opened or closed correctly.

an error was found during test 7 that made it only count half the endpoints

figure 8

```

1 #include <iostream>
2 using namespace std;
3 int main() {
4     int i = -1;
5     do {
6         i++;
7         cout << i << "\n";
8     } while (i < 50);
9     cout << i << "\n";
10    int p = -1;
11    while (p < 5) {
12        p++;
13        cout << p << "\n";
14    }
15    return 0;
16    while (true) //go forever
17    {
18    }
19 }

```

the program will now start looking for errors

```

#include <iostream>
using namespace std;
int main() {
    int i = -1;
    do {
        i++;
        cout << i << "\n";
    } while (i < 50);
    cout << i << "\n";

    int p = -1;
    while (p < 5) {
        p++;
        cout << p << "\n";
    }
    return 0;
    while (true) //go forever
    {
    }
}

```

number of code brackets opened 4
number of code brackets closed 4
number of end lines 9
total number of lines in tested code 19
all loops are working correctly

this was fixed as soon as the cause was found.
figure 9

```
1 #include <iostream>
2 using namespace std;
3 int main() {
4     int i = -1;
5     do {
6         i++;
7         cout << i << "n";
8     } while (i < 50);
9     cout << i << "n";
10    int p = -1;
11    while (p < 5) {
12        p++;
13        cout << p << "n";
14    }
15    return 0;
16    while (true) //go forever
17    {
18    }
19 }
```

```
using namespace std;
int main() {
    int i = -1;
    do {
        i++;
        cout << i << "n";
    } while (i < 50);
    cout << i << "n";

    int p = -1;
    while (p < 5) {
        p++;
        cout << p << "n";
    }
    return 0;
    while (true) //go forever
    {
    }
}
number of code brackets opened 4
number of code brackets closed 4
number of end lines 19
total number of lines in tested code 19
total number of blanklines in tested code 0
all loops are working correctly

please do not use using namespace std;
it is bad practice from a security standpoint due to all the packages it contains
```

to make code safer a recommendation of using `std::cout` over the package `namespace std` was put forward this is due to all the packages contained in `namespace std` that could be exploited if left in a normal program.

Chapter 22

conclusion

in conclusion the design and implementation stage has allowed for combining of all knowledge from the literature review and using it as reference to bring forth new ideas for the project, for example the algorithms, deterministic rule sets, ways to find errors, the many different algorithms that can all come together to make the project a success.

examples of this knowledge are: Real-Time Task Scheduling and Multi-Vehicle Navigation for Intra-Logistics by (Maximilian Berndt, Christoph Fischer and Dennis Krummacker), were they used algorithms combined with deterministic rule sets to make an automatic task scheduling and navigation AI for ways to automate trucks between factories buildings for possible future automation of the entire plant industry, this is were a group of buildings are built together for one purpose e.g melting and shaping steel starting from ore etc.

compilers optimize the code on a small scale when programs are compiled removing all useless information for the computer, this is due to the way the human brain works and computers work, computers work in 0s and 1s which is known as Binary base maths, humans use a 10 factor maths system this is known as decimal base maths. we are also much better with language than we are with numbers as such a system of compilers was made for transforming 10 base numbers and letters into 0s and 1s the computer can understand and compute easily.

with all the knowledge brought together the project can proceed to the next steps of creating the program, testing the program and gathering results. this encountered many difficulties as seen above.

Part V

testing and evaluation

Development type project focus on finishing code and submit put snippets of code into design and implementation explaining problems encountered for some marks. say what worked well and what didn't what troubles were encountered the testing of the project and evaluation of the finished program can be seen below, the testing followed the previously described research steps or logging all outputs and all tests preformed, all code changes made and all answers output both the numerical number of errors found and later on the possible solutions offered and the one recommended.

Chapter 23

testing

the results shown in the table below show that the research project was a partial success.

test	expected output	actual output	test code number
test 1	all end lines counted	total number of lines counted end lines ignored	test code 1
test 2	all end lines counted	1 end lines counted	test code 1
test 3	all end lines counted	all end lines counted	test code 1
test 4	all brackets counted	1 close bracket counted	test code 1
test 5	all open brackets counted	all open brackets counted	test code 2
test 6	all close brackets counted	all close brackets counted	test code 3
test 7	differentiate what bracket is missing	told missing close as missing open	test code 3
test 8	differentiate what bracket is missing	successfully differentiated what bracket was missing	test code 3
test 9	end lines all counted	missing end lines	test code 4
test 10	end lines all counted	end lines all counted glitch found and fixed	test code 4
test 11	number of headers in program calculated	correct number of headers displayed	test code 4
test 12	number of blank lines worked out	correct number of blank lines displayed	test code 4
test 13	the code to point out some vulnerabilities in coding practice	works if there are too many headers	test code 6

the first few failures were due to coding errors and logical problems in the algorithms for finding errors. there were mistakes made during this project such as forgetting to use ASCII code straight from the start.

the project was a success to some degree, this is due to the projects ability to find if loops are correctly working and identify what side of the loop has a problem if there are any errors, unfortunately the code is less complex than it could of been and as such can not pinpoint the exact loop causing errors.

the project also managed to succeed in identifying the total number of end liens this means it is possible to compare the number of total lines to end lines and work out if there is a missing end line.

the project can also successfully point out miner security vulnerabilities such as when a program put into it uses "using namespace std", unfortunately the project was not able to successfully pinpoint global variables from local variables so that part of the project was a failure.

the project can also successfully point out when lots of headers are being used and recommend to delete some if possible.

```
1 #include <iostream>
2 #include <thread>
3 #include <chrono>
4 #include <iostream>
5 #include <thread>
6 #include <chrono>
7 #include <iostream>
8 #include <thread>
9 #include <chrono>
10 #include <iostream>
11 #include <thread>
12 #include <chrono>
13 using namespace std;

int i = 37;
do {
    i--;
    cout << "there are " << i << " days till xmas" << "\n";
} while (i > 0);
cout << i << "\n";

std::this_thread::sleep_for(std::chrono::milliseconds(200000));

return 0;
while (true) //go forever
{
}

number of code brackets opened 3
number of code brackets closed 3
total number of headers in tested code 12
total number of blanklines in tested code 2
number of end lines 8
total number of lines in tested code 28
all loops are working correctly

please do not use using namespace std;
it is bad practice from a security standpoint due to all the packages it contains
try using std::cout; instead this means other types will
not be used by hackers in weakspots later as they will not be available

please delete some headers if possible;
as the number increased so does the security risk
```

figure 10

Chapter 24

evaluation

the data shows that it is defiantly possible to find errors using deterministic rule sets, unfortunately that will be a long complex process to create. the data also shows that it can be used for finding security vulnerabilities that are not too complex.

as for the projects ability to automatically find complex logical errors and pinpoint there exact source, eventually providing solutions to problems found unfortunately did not succeed, that part of the program was much more complex than expected leading to many errors and lots of wasted time that the project could not afford in the end it was removed from the program.

the programs ability to find possible optimization and go from there never managed to take off, this was due to the problems with finding complex errors that would be needed for optimization and also the difficulty of differentiating the global variables from the local variables,

most of the data from the literature review agrees with the project goal, only a few disagree without going into detail as to why deterministic rule sets should not be used. the data also agreed with the previously found disadvantage of the program being the time it takes to calculate all errors, during the project it was very hard to see the effects but when the program was scaled up, it became quite apparat, test code 7 was made to be ridiculous just as a test and the impact was already starting to become visible. this problem could be fixed by moving the perimeters around to be in the order of most used to least used but this is a time consuming processes that would need a much bigger test.

the program was a success in its ability to find errors in the code caused due to missing end lines or loops set up badly, it could also successfully point out when the code was lacking security features, however automatic optimization and the providing of solutions was too difficult due to all the possible errors and different ways to solve them. another problem with the program was the lack of ability to test code that use two programming languages at the same

time, this could possibly be fixed by making the program check for each type of language for all documents being tested and then compare the results, if lines of code work for one language but not another, check to see if all lines work if all lines should work mark it as working, but this could cause errors in its own right as programs that do not work could get past the filter using this method due to logical errors. another problem with the project was its lack of ability to find logical errors due to the many complex different factors causing them.

for this project the aims of the main objectives were mostly met, the program can successfully find when there are missing end lines and loop errors, it can also point out lacking security features, however these printouts are far from what the finished project should of been able to do with pinpointing the problems source, the project can not provide possible solutions and optimization, eventually these were removed from the program due to all the errors they were causing with getting the rest of the program working correctly whilst they were still present.

the information gained from this project is mostly as expected, that it is possible to use deterministic rule sets to find errors and lacking security features in code, however this does take some time. that it should be possible to use deterministic rule sets combined with a rapidly exploring random tree to find solutions to the errors and possible optimization paths, lastly it should also be possible to use a deterministic rule set to find complex or logical errors in code as long as the rule set is complex and set up correctly.

if this project were to be repeated a few chances to be implemented would be: a much longer deadline at times the project felt rushed, after the literature review the code would start to make sure it was completed with plenty of time left over, lastly get most of the report done very early and only come back to it to finish the abstract, experiment reports, evaluation, conclusion and appendix.

Part VI

**conclusions and
recommendations**

Chapter 25

summary

in summary this project was a partial success it could do part of what it set out to do in minor success. the results proved usefull as they prove that this can be done with much more time, the complexity that this program could be scaled up to would allow for solving many errors in code and finding out how most logical errors come from, however to do this the project would need to be much larger, much better managed and tested on a much larger scale.

the overall summary for this project shows that the project was nearly a complete failure, this was due to expected deadlines being much closer than expected, needing more research and work put in than expected, a near restart in the middle of the project and a lack of motivation after a topic change to reduce the difficulty from AI to deterministic rule sets. this all came together to greatly increase the time and work the project needed put in to it, a complete rewrite to make sure it all followed the correct scientific method was also needed within the last two months. the project also took much longer to program than expected due to needing to make sure no features broke a working feature whilst still coming together to complete all project goals at once.

Chapter 26

conclusion

the project was both a success and a failure.

the program was no were near as complex as it should of been and for the functions it did achieve it preformed them at a normal level,

the project failed as it could not find complex logical errors and preform optimization,

the results show that the program is not completely functioning as intended, this is due to having too many features being worked on at once as well as needing to not compromise the previous features. this lead to last minute errors were some features were removed entirely, and others kept, this helped save the project but also simplified it removing some planed features making it less successful than it should of been.

in conclusion it is defiantly possible to use deterministic rule sets to find logical errors in code, however the code will be vastly complex and take a long time to compute.

Chapter 27

recommendations for further research

after the project was completed recommendations for further research shall be proposed further research is recommended for:

- making the project work for all programming languages,

- making the project much better at finding and pointing out lacking security features.

- making the program completely automatic for finding errors and testing all possible solutions.

- making the program test automatically if the optimizations have the same output as the unoptimized code to check the program is still doing the intended functions after being optimized.

Part VII

Appendices

Part VIII

Declaration of Originality

I, Matthew Lee Cooze declare that I am the sole author of this Project; that all references cited have been consulted; that I have conducted all work of which this is a record, and that the finished work lies within the prescribed word limits.

This work has not previously been accepted as part of any other degree submission.

Signed : Matthew Cooze

Date : 19/11/2021

Part IX

FORM OF CONSENT

I Matthew Lee Cooze hereby consent that my Project, submitted in candidature for the MSC Computer Networks and Cyber Security degree, if successful, may be made available for inter-library loan or photocopying (subject to the law of copyright), and that the title and abstract may be made available to outside organisations.

Signed : Matthew Cooze

Date : 19/11/2021

Part X
reflections

during the project at times i lost track of the main goal and got lost and tried to add more features to the project worrying it would be lacking, this greatly slowed down completion of the project as it meant branching paths kept coming up during research and development, this was bad for the project overall.

another problem during development was that i was unsure if this was an experiment or a development and that i should of clarified that before doing work and changing things as the project developed

the project was greatly lacking in time management skills, the project should of been planned out much more accurately with small targets to achieve by specific dates throughout.

more priority should of been assigned to the MSC project.

more communication with the MSC supervisor, this would of made life a lot easier, for example not changing the project later due to it being deemed too difficult for something else midway through the deadline.

the project should of been much more organised with writing down the subjects in groups as moving them around for grouping later was very time consuming.

keeping track of references was done quite well by putting both the link name and page under the subject paragraph into the document until the submission.

the project was quite successful with finding most of the topics needed for the paper on google scholar, but it was less successful with finding useful papers and all needed heavy screening.

Part XI

references

Christophe Dubach, Timothy M. Jones, Edwin V. Bonilla, Grigori Fursin, and Michael F. P. O’Boyle. (2009). *Portable compiler optimisation across embedded programs and microarchitectures using machine learning*. In *Proceedings of the 42nd Annual IEEE/ACM International Symposium on Microarchitecture (MICRO 42)*. Association for Computing Machinery, New York, NY, USA, 78–88. DOI:<https://doi.org/10.1145/1669112.1669124>, available from https://dl.acm.org/doi/abs/10.1145/1669112.1669124?casa_token=O-INDFpo3bUAAAAA%3A_JoC9VaFsZMY-zM5jh-xgEjOjHwe2C-PMSWLueFvhmTr9bESZdMdoHgqRrQjCN

Rios and Butts WIRED + blackhat (08.09.2018) *A New Pacemaker Hack Puts Malware Directly on the Device*
available from <https://www.wired.com/story/pacemaker-hack-malware-black-hat/>

trustwave + BBC (2021) *How I Hacked A Nuclear Power Plant*
available from <https://www.trustwave.com/en-us/resources/blogs/trustwave-blog/how-i-hacked-a-nuclear-power-plant/>

NASA’S(1962-2021) *Mariner 1*
available from <https://nssdc.gsfc.nasa.gov/nmc/spacecraft/display.action?id=MARIN1>

alzheimer-europe(Friday 21 August 2009) *Research methods*
available from <https://www.alzheimer-europe.org/Research/Understanding-dementia-research/Types-of-research/Research-methods>

Fact Fiend - With Karl Smallwood, (14 December 2020) YouTube *That Time a Company Paid to Remaster a Game Just... Didn’t (Why This Video ISN’T Sponsored)*.
available from https://www.youtube.com/watch?v=8RbmQ49y9nA&ab_channel=FactFiend-WithKarlSmallwood

Stuff Made Here (4 April 2021) YouTube *My most dangerous robot*
available from https://www.youtube.com/watch?v=ix68oRfI5Gw&ab_channel=StuffMadeHere

Veritasium (2021) YouTube *The Universe is Hostile to Computers*
available from https://www.youtube.com/watch?v=AaZ_RSt0KP8&t=1238s&ab_channel=Veritasium

J. F. Ziegler, (1996) IBM Journal of Research and Development, *Terrestrial cosmic rays*,
available from <https://ve42.co/Ziegler1996>

Stefano Cereda, Gianluca Palermo, Paolo Cremonesi, and Stefano Doni. (2020). Association for Computing Machinery, *A Collaborative Filtering Approach for the Automatic Tuning of Compiler Optimisations*. In *The 21st ACM SIGPLAN/SIGBED Conference on Languages, Compilers, and Tools for Embedded Systems*.
available from DOI:<https://doi.org/10.1145/3372799.3394361>

LaValle, S. M. (1998). *Rapidly-exploring random trees: A new tool for path planning*.
available from <https://www.cs.csustan.edu/~xliang/Courses/CS4710-21S/Papers/06%20RRT.pdf>

Stanford university (2021) *an introduction to philosophy*.
available from <https://web.stanford.edu/~bobonich/glances%20ahead/III.logic.language.html>

Hojjat Adeli,(1995) *ASCI Concurrent Structural Optimization on Massively Parallel Supercomputer*
available from [https://ascelibrary.org/doi/abs/10.1061/\(ASCE\)0733-9445\(1995\)121:11\(1588\)](https://ascelibrary.org/doi/abs/10.1061/(ASCE)0733-9445(1995)121:11(1588))

Saylor Academy (2012) *Principles of Sociological Inquiry: Qualitative and Quantitative Methods*.
available from https://saylordotorg.github.io/text_principles-of-sociological-inquiry-qualitative-and-quantitative-methods/s05-03-inductive-or-deductive-two-dif.html

Berndt, M., Fischer, C., Krummacker, D. (2013) *researchgate Real-Time Task Scheduling and Multi-Vehicle Navigation for Intra-Logistics*.
available from https://www.researchgate.net/profile/Dennis-Krummacker/publication/348431493_Real-Time_Task_Scheduling_and_Multi-Vehicle_Navigation_for_Intra-Logistics/links/5ffec1ab45851553a03dc5eTime-Task-Scheduling-and-Multi-Vehicle-Navigation-for-Intra-Logistics.pdf

Padoin, E.L., Pilla, L.L., Castro, M., Boito, F.Z., Navaux, P.O.A. and Méhaut, J.F. (2015). *IET Computers & Digital Techniques Performance/energy trade-off in scientific computing: the case of ARM big. LITTLE and Intel Sandy Bridge..*
available from https://librarysearch.uwtsd.ac.uk/discovery/openurl?institution=44WHELF_UWTSD&vid=44

Neuman, W. L., Wiegand, B. (2000) US department of justice *Criminal justice research methods: Qualitative and quantitative approaches*.
available from <https://www.ojp.gov/ncjrs/virtual-library/abstracts/criminal-justice-research-methods-qualitative-and-quantitative>

TED-ed (2021) 1 YouTube *The method that can "prove" almost anything - James A. Smith*
available from https://www.youtube.com/watch?v=i60wwZDA1CI&ab_channel=TED-Ed

TED-ed (2021) 2 TED-ed *The method that can "prove" almost anything - James A. Smith*
available from <https://ed.ted.com/lessons/the-method-that-can-prove-almost-anything-james-a-smith#digdeeper>

Veritasium (2016) YouTube *Is Most Published Research Wrong?*
available from https://www.youtube.com/watch?v=42QuXLuH3Q&ab_channel=Veritasium

Cormen, T. H., Leiserson, C. E., Rivest, R. L., Stein, C. (2009).MIT press.

Introduction to algorithms.

available from https://books.google.co.uk/books?hl=en&lr=&id=aefUBQAAQBAJ&oi=fnd&pg=PR5&dq=Algorithm+errors+in+history&redir_esc=y#v=onepage&q=Algorithms&f=false

nick harley (2021) raygun *11 of the most costly software errors in history*

available from <https://raygun.com/blog/costly-software-errors-history/>

Ron Ben Natan (2015) Elsevier digital press *Implementing Database Security and Auditing.*

available from https://books.google.co.uk/books?hl=en&lr=&id=5WIP24cbtSEC&oi=fnd&pg=PP1&dq=Implementing+Database+Security+and+Auditing&redir_esc=y#v=onepage&q=Implementing+Database+Security+and+Auditing&f=false

CodeBeauty (2021) 1 YouTube *What is "using namespace std" and why is it considered a BAD PRACTICE? C++ Programming Questions*

available from https://www.youtube.com/watch?v=etQX4Mme2f4&ab_channel=CodeBeauty

Part XII
ethics form

APPLICATION FOR ETHICAL APPROVAL

In order for research to result in benefit and minimise risk of harm, it must be conducted ethically. A researcher may not be covered by the University's insurance if ethical approval has not been obtained prior to commencement.

The University follows the OECD Frascati manual definition of **research activity**: "creative work undertaken on a systematic basis in order to increase the stock of knowledge, including knowledge of man, culture and society, and the use of this stock of knowledge to devise new applications". As such this covers activities undertaken by members of staff, postgraduate research students, and both taught postgraduate and undergraduate students working on dissertations/projects.

The individual undertaking the research activity is known as the "principal researcher".

Ethical approval is not required for routine audits, performance reviews, quality assurance studies, testing within normal educational requirements, and literary or artistic criticism.

Please read the notes for guidance before completing ALL sections of the form.

This form must be completed and approved prior to undertaking any research activity. Please see Checklist for details of process for different categories of application.

Delete the Guidance Notes at the end of the form BEFORE submitting your application

SECTION A: About You (Principal Researcher)

1	Full Name:	Matthew Lee Cooze			
2	Tick all boxes which apply:	Member of staff:	☐	Honorary research fellow:	☐
3	Undergraduate Student	☐	Taught Postgraduate Student	☒	Postgraduate Research Student
4	Institute/Academic Discipline/Centre:	University of Wales Trinity Saint David			
5	Campus:	SA1 SWANSEA WATERFRONT IQ building			
6	E-mail address:	1911017@student.uwtsd.ac.uk			
7	Contact Telephone Number:	Home phone 01792 510050 (preferred) Mobile 07496 152006,			
For students:					
8	Student Number:	1911017			
9	Programme of Study:	Computer Networks and Cyber Security (MSc)			
10	Director of Studies/Supervisor:	supervisor: Tim Bashford			

SECTION B: Approval for Research Activity

1	Has the research activity received approval in principle? (please check the Guidance Notes as to the appropriate approval process for different levels of research by different categories of individual)	YES	☐	NO	☐
					Date
2	If Yes, please indicate source of approval (and date where known):	Research Degrees Committee	☐		
3	Approval in principle must be obtained from the relevant source prior to seeking ethical approval	Institute Research Committee	☐		
4		Other (Taught masters research Ethical Approval Forms are considered within Institutes)	☐		

SECTION C: Internal and External Ethical Guidance Materials

	Please list the core ethical guidance documents that have been referred to during the completion of this form (including any discipline-specific codes of research ethics, and also any specific ethical guidance relating to the proposed methodology). Please tick to confirm that your research proposal adheres to these codes and guidelines.
1	UWTSD Research Ethics & Integrity Code of Practice ☒
2	UWTSD Research Data Management Policy ☒
3	☐

SECTION D: External Collaborative Research Activity

1	Does the research activity involve collaborators outside of the University?	YES	☐	NO	☒	
2	If Yes, please provide the name of the external organisation and name and contact details for the main contact person and confirmation this person has consented to their personal data being shared as part of this collaboration.					
3	Institution					
4	Contact person name					
5	Contact person e-mail address					
6	Has this individual consented to sharing their details on this form?	YES	☐	NO	☐	
7	Are you in receipt of a KESS scholarship?	YES	☐	NO	☐	
8	Is your research externally funded	YES	☐	NO	☐	
9	Are you specifically employed to undertake this research in either a paid or voluntary capacity?	Voluntary	YES	☐	NO	☐
10		Employed	YES	☐	NO	☐
11	Is the research being undertaken within an existing UWTSD Athrofa Professional Learning Partnership (APLP)	If YES then the permission question below does not need to be answered.	YES	☐	NO	☐
12	Permission to undertake the research has been provided by the partner organisation	(If YES attach copy) If NO the application cannot continue	YES	☐	NO	☐

Where research activity is carried out in collaboration with an external organisation

13	Does this organisation have its own ethics approval system?	YES	☐	NO	☐
14	If Yes, please attach a copy of any final approval (or interim approval) from the organisation				

SECTION E: Details of Research Activity

1	Indicative title:	MSC Research Proposal Programming a deterministic rule set to Find coding errors		
2	Proposed start date:	Monday, 19 th April, 2021	Proposed end date:	Thursday, 19 th November, 2021
3	Introduction to the Research (maximum 300 words) Ensure that you write for a <u>Non-Specialist Audience</u> when outlining your response to the three points below: • Purpose of Research Activity • Proposed Research Question • Aims of Research Activity • Objectives of Research Activity Demonstrate, briefly, how <u>Existing Research</u> has informed the proposed activity and explain • What the research activity will add to the body of knowledge • How it addresses an area of importance.			
4	Purpose of Research Activity Key deterministic rule set (in deterministic models, the output of the model is fully determined by the parameter values and the initial values) rule set, is a group of code specifically made to find or target a specific cause or issue. The purpose of this research is to see if a deterministic rule set can find complex coding errors by scanning through the source code. right now most software and interpreters are design to find small errors in code, if the research is successful it would allow for complex problems to be found before they become a problem. it will also allow for optimisation and removal of bugs pre-release for software, with an added bonuses of extra security measures being in place as it will be able to remind people of silly errors people forget about when rushing e.g global variables, taking up much more memory than needed etc. this research will add to the body of knowledge of coding and errors more specifically google scholar (deterministic rules for code errors) with 190,000 results most of these results are using it to manage resources, algorithms or firewall and security.			
5	Research Question Can a deterministic rule set to Find complex coding errors ? (this box should expand as you type)			
6	Aims of Research Activity			

	Project Aim To build a program using deterministic nature and rule based searching, code will then be input through a text file and searched to find unsecure/ bad coding practices e.g memory leaks, heap corruption etc }
7	Objectives of Research Activity <pre> \begin{enumerate} \item Research if rule sets can be used to find complex computing issues. \item study if deterministic nature can be added to the rule set to improve the code. \item find what causes a lot of computer errors. \item study if the errors are man made or due to the way computers work. \item Research how effectively the program can find programming errors. \end{enumerate} </pre>
8	Proposed methods (maximum 600 words) Provide a brief summary of all the methods that may be used in the research activity, making it clear what specific techniques may be used. If methods other than those listed in this section are deemed appropriate later, additional ethical approval for those methods will be needed.
9	<pre> \begin{frame}[t]{Research Methodology } \vspace{10pt} The three main sources of research will be: \\ Literature review will be for finding all needed information for research and programming, will also be covering finding the OSI and MIT license approved code for testing with Data collection will be for comparing the code between modification and how it has improved at finding faults in the scanned code (testing) Analysis of collected data. This will be analysing the results of all data collected it will use methods researched in literature review and data from the tests run \vspace{0.5em} The chosen research methodology will be Quantitative this is for the logical reason as it works better with code \vspace{0.5em} When measuring the Success of the program the two best ways of comparing are. \vspace{0.5em} \\ The number of faults found over the number of files scanned. \vspace{0.5em} \\ How well it is able to find the source of the error in the files scanned. \\ \end{frame} </pre>

10	Location of research activity Identify all locations where research activity will take place.		
11	Swansea, home – my PC <i>(this box should expand as you type)</i>		
12	Research activity outside of the UK If research activity will take place overseas, you are responsible for ensuring that local ethical considerations are complied with and that the relevant permissions are sought. Specify any local guidelines (e.g. from local professional associations/learned societies/universities) that exist and whether these involve any ethical stipulations beyond those usual in the UK (provide details of any licenses or permissions required). Also specify whether there are any specific ethical issues raised by the local context in which the research activity is taking place, for example, particular cultural and/or legal sensitivities or vulnerabilities of participants.		
13	<i>(this box should expand as you type)</i>		

14	Use of documentation not in the public domain: Are any documents NOT publicly available?	NO	☒
		YES	☐
15	If Yes, please provide details here of how you will gain access to specific documentation that is not in the public domain and that this is in accordance with prevailing data protection law of the country in question and England and Wales. <i>(this box should expand as you type)</i>		

SECTION F: Scope of Research Activity

1	Will the research activity include:	YES	NO
2	Use of a questionnaire or similar research instrument?	☐	☒
3	Use of interviews?	☐	☒
4	Use of diaries?	☐	☒
5	Participant observation with their knowledge?	☐	☒
6	Participant observation without their knowledge?	☐	☒
7	Use of video or audio recording?	☐	☒
8	Access to personal or confidential information without the participants' specific consent?	☐	☒
9	Administration of any questions, test stimuli, presentation that may be experienced as physically, mentally or emotionally harmful / offensive?	☐	☒
10	Performance of any acts which may cause embarrassment or affect self-esteem?	☐	☒
11	Investigation of participants involved in illegal activities?	☐	☒
12	Use of procedures that involve deception?	☐	☒
13	Administration of any substance, agent or placebo?	☐	☒
14	Working with live vertebrate animals?	☐	☒

15	Other primary data collection methods, please explain in this box For example, 'focus groups'. Please indicate the type of data collection method(s) in this box and tick the accompany box.	☐	☐
16	Details of any other primary data collection method: (this box should expand as you type)		

If NO to every question, then the research activity is (ethically) low risk and **may** be exempt from **some** of the following sections (please refer to Guidance Notes).

If YES to any question, then no research activity should be undertaken until full ethical approval has been obtained.

SECTION G: Intended Participants

1	Who are the intended participants:	YES	NO
2	Students or staff at the University?	☐	☒
3	Adults (over the age of 18 and competent to give consent)?	☐	☒
4	Vulnerable adults?	☐	☒
5	Children and Young People under the age of 18? (Consent from Parent, Carer or Guardian will be required)	☐	☒
6	Prisoners?	☐	☒
7	Young offenders?	☐	☒
8	Those who could be considered to have a particularly dependent relationship with the investigator or a gatekeeper?	☐	☒
9	People engaged in illegal activities?	☐	☒
10	Others (please identify specifically any group who may be unable to give consent) please indicate here and tick the appropriate box.	☐	☒
11	Other – please indicate here: (this box should expand as you type)		

12	Participant numbers and source Provide an estimate of the expected number of participants. How will you identify participants and how will they be recruited?	
13	How many participants are expected?	0 – 1 myself as the researcher (this box should expand as you type)
14	Who will the participants be?	Myself for modifying the code and running tests (this box should expand as you type)
15	How will you identify the participants?	Myself (this box should expand as you type)

16	Information for participants:	YES	NO	N/A
17	Will you describe the main research procedures to participants in advance, so that they are informed about what to expect?	☐	☐	☒
18	Will you tell participants that their participation is voluntary?	☐	☐	☒
19	Will you obtain written consent for participation?	☐	☐	☒
20	Will you explain to participants that refusal to participate in the research will not affect their treatment or education (if relevant)?	☐	☐	☒
21	If the research is observational, will you ask participants for their consent to being observed?	☐	☐	☒
22	Will you tell participants that they may withdraw from the research at any time and for any reason?	☐	☐	☒
23	With questionnaires, will you give participants the option of omitting questions they do not want to answer?	☐	☐	☒
24	Will you tell participants that their data will be treated with full confidentiality and that, if published, it will not be identifiable as theirs?	☐	☐	☒
25	Will you debrief participants at the end of their participation, in a way appropriate to the type of research undertaken?	☐	☐	☒
26	If NO to any of above questions, please give an explanation			
27	<i>(this box should expand as you type)</i>			

28	Information for participants:	YES	NO	N/A
29	Will participants be paid?	☐	☐	☒
30	Is specialist electrical or other equipment to be used with participants?	☐	☐	☒
31	Are there any financial or other interests to the investigator or University arising from this study?	☐	☐	☒
32	Will the research activity involve deliberately misleading participants in any way, or the partial or full concealment of the specific study aims?	☐	☐	☒
33	If YES to any question, please provide full details			
34	<i>(this box should expand as you type)</i>			

SECTION H: Anticipated Risks

1	Outline any anticipated risks that may adversely affect any of the participants, the researchers and/or the University, and the steps that will be taken to address them.		
	If you have completed a full risk assessment (for example as required by a laboratory, or external research collaborator) you may append that to this form.		
2	Full risk assessment completed and appended?	Yes	☐
		No	☒
3	Risks to participants For example: emotional distress, financial disclosure, physical harm, transfer of personal data, sensitive organisational information		
4	Risk to Participant: 0 participants <i>(this box should expand as you type)</i>	How will you mitigate the Risk to Participant <i>(this box should expand as you type)</i>	

5	If research activity may include sensitive, embarrassing or upsetting topics (e.g. sexual activity, drug use) or issues likely to disclose information requiring further action (e.g. criminal activity), give details of the procedures to deal with these issues, including any support/advice (e.g. helpline numbers) to be offered to participants. Note that where applicable, consent procedures should make it clear that if something potentially or actually illegal is discovered in the course of a project, it may need to be disclosed to the proper authorities			
	<i>(this box should expand as you type)</i>			
6	Risks to investigator For example: personal safety, physical harm, emotional distress, risk of accusation of harm/impropriety, conflict of interest			
	Risk to Investigator: Downloading a virus masked as code, <i>(this box should expand as you type)</i>	How will you mitigate the Risk to Investigator: Use anti-virus, anti-malware and virtual machines were possible to test code <i>(this box should expand as you type)</i>		
7	University/institutional risks For example: adverse publicity, financial loss, data protection			
	Risk to University: Plagiarism Failure to meet legal and professional objectives	How will you mitigate the Risk to University: Make sure all work read and quoted from is referenced. make sure to read up on GDPR and British data protection acts before commencing any downloads of code and testing with them, make sure the code is referenced with link attached in the references. With licence at time of download in case that changes later		
8	Disclosure and Barring Service			
9	If the research activity involves children or vulnerable adults, a Disclosure and Barring Service (DBS) certificate must be obtained before any contact with such participants.	YES	NO	N/A
10	Does your research require you to hold a current DBS Certificate?	☐	☐	☒

SECTION I: Feedback, Consent and Confidentiality

1	Feedback What de-briefing and feedback will be provided to participants, how will this be done and when?
	None (no participants, only code) <i>(this box should expand as you type)</i>
2	Informed consent Describe the arrangements to inform potential participants, before providing consent, of what is involved in participating. Describe the arrangements for participants to provide full consent before data collection begins. If gaining consent in this way is inappropriate, explain how consent will be obtained and recorded in accordance with prevailing data protection legislation.
	By uploading code to both Dockerhub and GitHub using an open source licence(OSI) in public mode you are giving consent for others to download and play around with your code. open source licence(OSI) Massachusetts Institute of Technology license (MIT)

	Only code with MIT License or OSI License will be used as both are open source and are open to be used, edited and run legally There are also ways to not share you code on both platforms(private mode) and the ability to use different licenses that are not open source. "Open source licenses are licenses that comply with the Open Source Definition — in brief, they allow software to be freely used, modified, and shared. To be approved by the Open Source Initiative (also known as the OSI)" \href{ https://opensource.org/licenses }open source initiative } <i>(this box should expand as you type)</i>
3	Confidentiality / Anonymity Set out how anonymity of participants and confidentiality will be ensured in any outputs. If anonymity is not being offered, explain why this is the case.
	All code to be tested will be from public sites where people have uploaded codes for others to test and play around with: e.g. Dockerhub, GitHub, people on these sites tend to use made up account names and as such start with anonymity, there code will be put through the tester to see what problems can be found if any, <i>(this box should expand as you type)</i>

SECTION J: Data Protection and Storage

In completing this section refer to the University's Research Data Management Policy and the extensive resources on the University's Research Data Management web pages (<http://uwtsd.ac.uk/library/research-data-management/>).

1	Does the research activity involve personal data (as defined by the General Data Protection Regulation 2016 "GDPR" and the Data Protection Act 2018 "DPA")?	YES	NO
	"Personal data" means any information relating to an identified or identifiable natural person ('data subject'). An identifiable natural person is one who can be identified, directly or indirectly, in particular by reference to an identifier such as a name, an identification number, location data, an online identifier or to one or more factors specific to the physical, physiological, genetic, mental, economic, cultural or social identity of that natural person.	☐	☒
2	If YES, provide a description of the data and explain why this data needs to be collected:		
	No personal data, A link to all the code used with reference to codes original purpose and what licenses it comes under, with a screenshot of the licence at time of download. <i>(this box should expand as you type)</i>		
3	Does it involve special category data (as defined by the GDPR)?	YES	NO
	"Special category data" means sensitive personal data consisting of information as to the data subjects' – (a) racial or ethnic origin, (b) political opinions, (c) religious beliefs or other beliefs of a similar nature, (d) membership of a trade union (within the meaning of the Trade Union and Labour Relations (Consolidation) Act 1992), (e) physical or mental health or condition, (f) sexual life,	☐	☒

	(g) genetics, (h) biometric data (as used for ID purposes),		
4	If YES, provide a description of the special category data and explain why this data needs to be collected:		
	(this box should expand as you type)		
5	Will the research activity involve storing personal data and/or special category data on one of the following:	YES	NO
6	Manual files (i.e. in paper form)?	☐	☒
7	University computers?	☐	☒
8	Private company computers?	☐	☒
9	Home or other personal computers?	☐	☒
10	Laptop computers/ CDs/ Portable disk-drives/ memory sticks?	☐	☒
11	"Cloud" storage or websites?	☐	☒
12	Other – specify:	☐	☒
13	For all stored data, explain the measures in place to ensure the security of the data collected, data confidentiality, including details of password protection, encryption, anonymisation and pseudonymisation:		
	A link to all the code used with reference to codes original purpose and what licenses it comes under, with a screenshot of the licence at time of download., as anonymising the usernames would not work with a link to the code next to them on a website with the username publicly available. (this box should expand as you type)		
14	All Data Storage		
15	Will the research activity involve any of the following activities:	YES	NO
16	Electronic transfer of data in any form?	☒	☐
17	Sharing of data with others at the University?	☐	☒
18	Sharing of data with other organisations?	☐	☒
19	Export of data outside the European Union or importing of data from outside the UK?	☒	☐
20	Use of personal addresses, postcodes, faxes, emails or telephone numbers?	☐	☒
21	Publication of data that might allow identification of individuals?	☐	☒
22	Use of data management system?	☐	☒
23	Data archiving?	☐	☒
24	If YES to any question, please provide full details, explaining how this will be conducted in accordance with the GDPR and DPA (and/or any international equivalent):		
	as the files can be stored on a server in any country in the world, and will be transferred electronically.		

	I will use Github as it is GDPR compliant and I will stick to EU countries to make sure I am within the law when it comes to code I download, all code downloaded will have MIT and OSI licensing. <i>(this box should expand as you type)</i>		
25	List all who will have access to the data generated by the research activity: Myself, supervisor, markers <i>(this box should expand as you type)</i>		
26	List who will have control of, and act as custodian(s) for, data generated by the research activity: Myself <i>(this box should expand as you type)</i>		
27	Give details of data storage arrangements, including security measures in place to protect the data, where data will be stored, how long for, and in what form. Will data be archived – if so how and if not why not. Data stored will be kept on a hard drive or in a latex document Data will be a link to open source code with reference of its original purpose, and screenshot of licence at time of download The code to be tested will also be kept for the duration of the project This data will not have any special security measures in place other than encryption due to it being open source and freely available on the internet, data will be stored for the duration of the MSC. Then deleted as required by GDPR <i>(this box should expand as you type)</i>		
28	Please indicate if your data will be stored in the UWTSD Research Data Repository (see https://researchdata.uwtsd.ac.uk/). If so please explain. <i>(Most relevant to academic staff)</i> No <i>(this box should expand as you type)</i>		
29	Confirm that you have read the UWTSD guidance on data management (see https://www.uwtsd.ac.uk/library/research-data-management/)	YES	☒
		NO	☐
30	Confirm that you are aware that you need to keep all data until after your research has completed or the end of your funding	YES	☒
		NO	☐

SECTION K: Declaration

31	The information which I have provided is correct and complete to the best of my knowledge. I have attempted to identify any risks and issues related to the research activity and acknowledge my obligations and the rights of the participants. In submitting this application I hereby confirm that I undertake to ensure that the above named research activity will meet the University's Research Ethics and Integrity Code of Practice which is published on the website: https://www.uwtsd.ac.uk/research/research-ethics/		
	Signature of applicant:	Matthew Cooze	Date: 19/11/2021

For STUDENT Submissions:

32	Director of Studies/Supervisor:		Date:
33	Signature:		

For STAFF Submissions:

34	Academic Director/ Assistant Dean:		Date:
35	Signature:		

Checklist: Please complete the checklist below to ensure that you have completed the form according to the guidelines and attached any required documentation:

☒	I have read the guidance notes supplied before completing the form.
☒	I have completed ALL RELEVANT sections of the form in full.
☐	I confirm that the research activity has received approval in principle
☐	I have attached a copy of final/interim approval from external organisation (where appropriate)
☐	I have attached a full risk assessment (and have NOT completed Section H of this form) (where appropriate) ONLY TICK IF YOU HAVE ATTACHED A FULL RISK ASSESSMENT
☒	I understand that it is my responsibility to ensure that the above named research activity will meet the University's Research Ethics and Integrity Code of Practice.
☒	I understand that before commencing data collection all documents aimed at respondents (including information sheets, consent forms, questionnaires, interview schedules etc.) must be confirmed by the DoS/Supervisor, module tutor or Academic Director.
☒	I have deleted the guidance notes before submitting the PG2 for consideration

RESEARCH STUDENTS AND STAFF ONLY

All communications relating to this application during its processing must be in writing and emailed to pgresearch@uwtsd.ac.uk, with the title 'Ethical Approval' followed by your name. You will be informed of the outcome of your claim by email; therefore it is important that you check your University and personal email accounts regularly.

STUDENTS ON UNDERGRADUATE OR TAUGHT MASTERS PROGRAMMES should submit this form (and receive the outcome) via systems explained to you by the supervisor/module leader.

This form is available electronically from the Academic Office web pages:
<https://www.uwtsd.ac.uk/academic-office/appendices-and-forms/>

Application Process

All staff research projects and all research students must submit the Ethical Approval Form to the **University Research Ethics Committee** via the Academic Office (pgresearch@uwtsd.ac.uk). Staff research directly in relation to personal study for taught undergraduate or Masters programmes should be submitted via the Institute procedures explained below.

Taught masters and taught undergraduate research Ethical Approval Forms are considered **within Institutes**. Institutes will provide details of the specific processes for this. Where the Ethical issues within any single ethical application are of particular concern the Institute will refer these to the University Research Ethics Committee. Any student activity that involves the collection of primary data needs to undergo Ethical approval, this includes assignment work as well as dissertations.

Part XIII

logbook

day(date)	description
day 1	produce research proposal
day 2	declaration of originality
day 3	form of consent
day 4	introduction, aim, objectives
day 5	introduction, structure of the project
day 6	literature review methodology
day 7	literature review optimization flags
day 8	literature review algorithms
day 9	literature review ECC memory
day 10	literature review deterministic rule sets
day 11	research methodology introduction, methodology
day 12	research methodology research overview
day 13	research methodology research design
day 14	research methodology triangulation
day 15	research methodology conclusion
day 16	design and implementation introduction
day 17	design and implementation problem formulation
day 18	design and implementation design
day 19	design and implementation implementation
day 20	design and implementation conclusion
day 21	testing and evaluation testing
day 22	testing and evaluation testing
day 23	testing and evaluation evaluation
day 24	testing and evaluation evaluation
day 25	conclusion and recommendations summary
day 26	conclusion and recommendations conclusion
day 27	conclusion and recommendations recommendations
day 28	reflections
day 29	bibliography/references
day 30	glossary/key
day 31	rewrite entire project due to bad paragraphing
day 32	create test tables and description
day 33	ethics form, logbook glossary/key
day 34	abstract

Part XIV
other material

Chapter 28

main code

```

#include <iostream>
#include <thread>
#include <chrono>
#include <iostream>
#include <fstream>
#include <string>

using std::cout;
using std::endl;
using std::string;
using std::ifstream;

class MSCcode
{
public:

    MSCcode(); //constructor
    void display(void); //display options
    bool checkcode(void);

};

void checkcode()
{
    bool std = false; // this is used for checking if the program has good security
practices in place.

    int opens = 0;
    // creates a variable called opens used for checking how many opening loops
there are
    int closes = 0;
    // creates a variable called closes used for checking how many closing loops
there are
    // these two variables can later be compared to check if loops are opening and
closing correctly.

    int lines = 0;
    // lines is used to see the total number of lines in the program
    int endlines = 0;
    // endlines is used to store the interger of all ; used to end lines

    int blanklines = 0;
    // a variables called blanklines to work out how much white lines are in the
program

    int hashtags = 0;
    // the above interger will be used to count the number of included headerfiles
in the program.

    string myText;
    // creates a variables called mytext to store and output the code being read in
for testing

    // name of the testcode file ready for future testing.
    ifstream MyReadFile("testcode.txt");

    // Use a while loop together with the getline() function to read the file line
by line
    while (getline(MyReadFile, myText)) {

```

```

std::string s(myText);

for (char& c : s)
{
    // std::cout << "One character: " << c << "\n";
    // the above code was for displaying the characters during
testing
    if (c == 59)
    {
        endlines++; // counts the number of end lines
    }

    // 123 is the character code for {
    else if (c == 123)
    {
        opens++; // counts how many code brackets were opened
    }

    // 125 is the character code for }
    else if (c == 125)
    {
        closes++; // counts how many code brackets were closed
    }

    // counts how many includes were used in the project
    else if (c == 35)
    {
        hashtags++;
    }
    if (myText == "using namespace std;")
    {
        std = true;
    }

    c = '*';
}

// Output the text from the file and end each line as appropriate to
make sure the output is correct
cout << myText << endl;

lines++; // counts the number of total lines in the code

}
cout << "number of code brackets opened " << opens << endl;
cout << "number of code brackets closed " << closes << endl;
blanklines = lines - endlines - closes - opens - hashtags;
cout << "total number of headers in tested code " << hashtags << endl;
cout << "total number of blanklines in tested code " << blanklines << endl;
cout << "number of end lines " << endlines << endl;
cout << "total number of lines in tested code " << lines << endl;

if (opens == closes)
{
    cout << "all loops are working correctly " << endl;
}
else if (closes > opens)

```

```

{
    cout << "error found in loop opening " << endl;
}
else
{
    cout << "error found in loop closing " << endl;
}

if (std == true)
{
    cout << endl;
    cout << "please do not use using namespace std;" << endl;
    cout << "it is bad practice from a security standpoint due to all the
packages it contains" << endl;
    cout << "try using std::cout; instead this means other types will" <<
endl;
    cout << "not be used by hackers in weakspots later as they will not be
available" << endl;

}

if (hashtags >= 8)
{
    cout << endl;
    cout << "please delete some headers if possible;" << endl;
    cout << "as the number increased so does the security risk" << endl;

}

// Close the file
MyReadFile.close();
std::this_thread::sleep_for(std::chrono::milliseconds(200000));
}

void display()
//draw puzzle as it is now
{
    system("cls"); // blanks the screen its clear screen
    cout << "the program will now start looking for errors please wait" << endl;
    //cout << "please press 2 for code optimization" << endl;
    //cout << "please press 3 to exit the program" << endl;
    //cout << endl;
    //cout << "please choose a valid option " << endl;
    cout << endl;
    // this will display the options available for the main menu
}

int main(void)
{

    //main menu for code security error finder
    while (true) //go forever
    {
        display(); //display menu options
        checkcode(); // checks if puzzle is solved

    }
    return 0;
}

```

```
}//the end
```

Chapter 29

code for testing other programs

the first 3 codes were used at the start of the project to check that reading code and finding missing brackets and end lines worked.

the next 3 codes were used to make sure security vulnerabilities could be found and that all code could be tracked throughout the program.

29.1 code 1

```
#include <iostream>

using namespace std;

int main() {

    int i = -1;

    do {

        i++;

        cout << i << "\n";

    } while (i < 50);

    cout << i << "\n"

    int i = -1;

    while (i < 5) {

        i++;

        cout << i << "\n";

    }

    return 0;

    while (true) //go forever

    {

    }

}
```

29.2 code 2

```
#include <iostream>

using namespace std;

int main() {

    int i = -1;

    do {

        i++;

        cout << i << "\n";

    } while (i < 50);

    cout << i << "\n"

    int i = -1;

    while (i < 5) {

        i++;

        cout << i << "\n";

    }

    return 0;

    while (true) //go forever

    {

    }

}
```

29.3 code 3

```
#include <iostream>

using namespace std;

int main() {

    int i = -1;

    do {

        i++;

        cout << i << "\n";

    } while (i < 50);

    cout << i << "\n"

    int i = -1;

    while (i < 5)

        i++;

    cout << i << "\n";

}

return 0;

    while (true) //go forever

    {

    }

}
```

29.4 code 4

```
#include <iostream>

using namespace std;

int main() {

    int i = -1;

    do {

        i++;

        cout << i << "\n";

    } while (i < 50);

    cout << i << "\n";

    int p = -1;

    while (p < 5) {

        p++;

        cout << p << "\n";

    }

    return 0;

    while (true) //go forever

    {

    }

}
```

29.5 code 5

```
#include <iostream>

#include <thread>

#include <chrono>

using namespace std;

int main() {

    int i = 37;

    do {

        i--;

        cout << "there are " << i << " days till xmas" << "\n";

    } while (i != 0);

    cout << i << "\n";


    std::this_thread::sleep_for(std::chrono::milliseconds(200000));


    return 0;

    while (true) //go forever

    {

    }

}
```

29.6 code 6

```
#include <iostream>

#include <thread>

#include <chrono>

using namespace std;

int main() {

    int i = 37;

    do {

        i--;

        cout << "there are " << i << " days till xmas" << "\n";

    } while (i != 0);

    cout << i << "\n";


    std::this_thread::sleep_for(std::chrono::milliseconds(200000));


    return 0;

    while (true) //go forever

    {

    }

}
```

29.7 code 7

```
#include <iostream>

#include <thread>

#include <chrono>

using namespace std;

int main() {

    int i = 37;

    do {

        i--;

        cout << "there are " << i << " days till xmas" << "\n";

    } while (i != 0);

    cout << i << "\n";

    int i = 37;

    do {

        i--;

        cout << "there are " << i << " days till xmas" << "\n";

    } while (i != 0);

    cout << i << "\n";

    int i = 37;

    do {

        i--;

        cout << "there are " << i << " days till xmas" << "\n";

    } while (i != 0);

    cout << i << "\n";

    int i = 37;

    do {

        i--;
```

```
        cout << "there are " << i << " days till xmas" << "\n";  
    } while (i != 0);  
    cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {
```

```
        i--;  
        cout << "there are " << i << " days till xmas" << "\n";  
    } while (i != 0);  
    cout << i << "\n";
```

```
int i = 37;  
  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);
```

```
cout << i << "\n";
```

```
int i = 37;
```

```
do {
```

```
    i--;
```

```
    cout << "there are " << i << " days till xmas" << "\n";
```

```
} while (i != 0);
```

```
cout << i << "\n";
```

```
int i = 37;
```

```
do {
```

```
    i--;
```

```
    cout << "there are " << i << " days till xmas" << "\n";
```

```
} while (i != 0);
```

```
cout << i << "\n";
```

```
int i = 37;
```

```
do {
```

```
    i--;
```

```
    cout << "there are " << i << " days till xmas" << "\n";
```

```
} while (i != 0);
```

```
cout << i << "\n";
```

```
int i = 37;
```

```
do {
```

```
    i--;
```

```
    cout << "there are " << i << " days till xmas" << "\n";
```

```
} while (i != 0);
```

```
cout << i << "\n";
```

```
int i = 37;
```

```
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";
```

```
} while (i != 0);  
cout << i << "\n";  
  
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;
```

```
        cout << "there are " << i << " days till xmas" << "\n";  
    } while (i != 0);  
    cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;  
do {  
    i--;  
    cout << "there are " << i << " days till xmas" << "\n";  
} while (i != 0);  
cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
int i = 37;

do {

    i--;

    cout << "there are " << i << " days till xmas" << "\n";

} while (i != 0);

cout << i << "\n";
```

```
std::this_thread::sleep_for(std::chrono::milliseconds(200000));
```

```
return 0;
```

```
while (true) //go forever
```

```
{
```

```
}
```

```
}
```

Chapter 30

link to demo

the following is a link to the quick YouTube demonstration.
<https://youtu.be/fxl35vDL2To>

Part XV

glossary/key

Heap corruption is when a program sometimes does not release all the memory it was using or worse released memory and then tried to use it conflicting with other programs.

logic errors: when the code technically is working just not doing what you want

runtime errors: the code can be perfect on your machine but if the server or a customers pc is running slightly different it can break the code, another way of causing this error is by inputting something slightly the wrong way than expected.

arithmetic errors: are errors when the code is doing maths

resource errors: this is when the computer tries to get more resources than the computer has available causing errors. memory leaks are a type of resource error where the computer does not free up memory once it is no longer needed

interface errors: these happen when the end user misunderstood how the code was designed to be run, compared to how the developer expects it to be run.

quantitative data is mathematical data this is data that can be described very accurately with numbers,

qualitative data is words observations interviews etc.

deterministic rule set (in deterministic models, the output of the model is fully determined by the parameter values and the initial values)

rule set, is a group of code specifically made to find or target a specific cause or issue.