**Documentation of my Differential Tutorial Program**

**Aims of the program**

The aim of my program is to teach people how to differentiate linear and nonlinear equations of functions. It will have a section in which will teach the basic of how to differentiate and then a set of questionnaires, with varying difficulty, to which the user can practice.

I also want to include a differentiation calculator. This is where a user can input a linear or nonlinear equation into the differentiation calculator and the derivative will be outputted.

**Requirements for the program**

The program will firstly need a menu in which the user can access the different options available in the program. I will probably use a while loop to achieve this.

The way I want to structure this program out is by first defining the instructions needed for each section of the program. I think this will make the actual building of the program easier and also make the code look nicer.

Underneath the instruction set I will have my while loop with the program’s menu.

There will be 6 sections to the program

**About the program:**

This section of the program will give an introduction to what the program is and what it aims to achieve. It will also have some rules as well as to how to input equations into the program.

**Introduction to differentiation:**

This section of the program is where the rules of differentiation are taught. This will mainly be text, but will discuss differentiation with an example included. After the rules have been read, there will be some practice questions for the user to answer so that they get the process of differentiation correct.

**A questionnaire on easy difficulty:**

This section of the program is where the user can practice their differentiation techniques on easy linear equations e.g. y = 2 x + 5

**A questionnaire on intermediate difficulty:**

This section of the program is where the user can practice their differentiation techniques on harder nonlinear equations e.g. y = 2 x^ 2 + 5 x + 6

**A questionnaire on hard difficulty:**

This section of the program is where the user can practice their differentiation techniques on even harder nonlinear equations e.g. y = 3 x^ 4 + 9 x^ 3 + 20 x + 16

**The differentiation calculator:**

This section of the program is where the user can input a linear or nonlinear equation into the differentiation calculator to which the output will the derivative e.g. y = 2 x^ 3 + 5 x + 6 dy/dx = 6 x^ 2 + 5

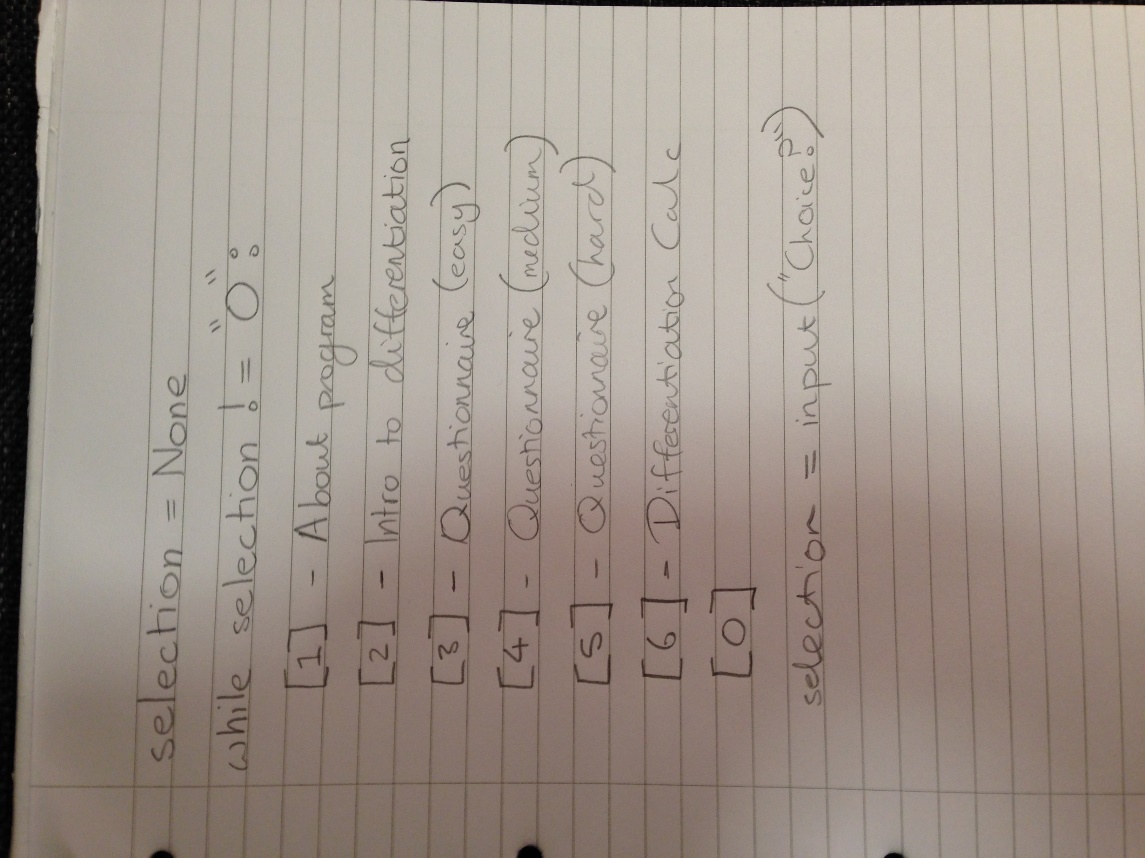
**The design of the program**

The menu

The menu will be made out of a while loop. The while loop will be not equal to 0 (while loop != 0:).

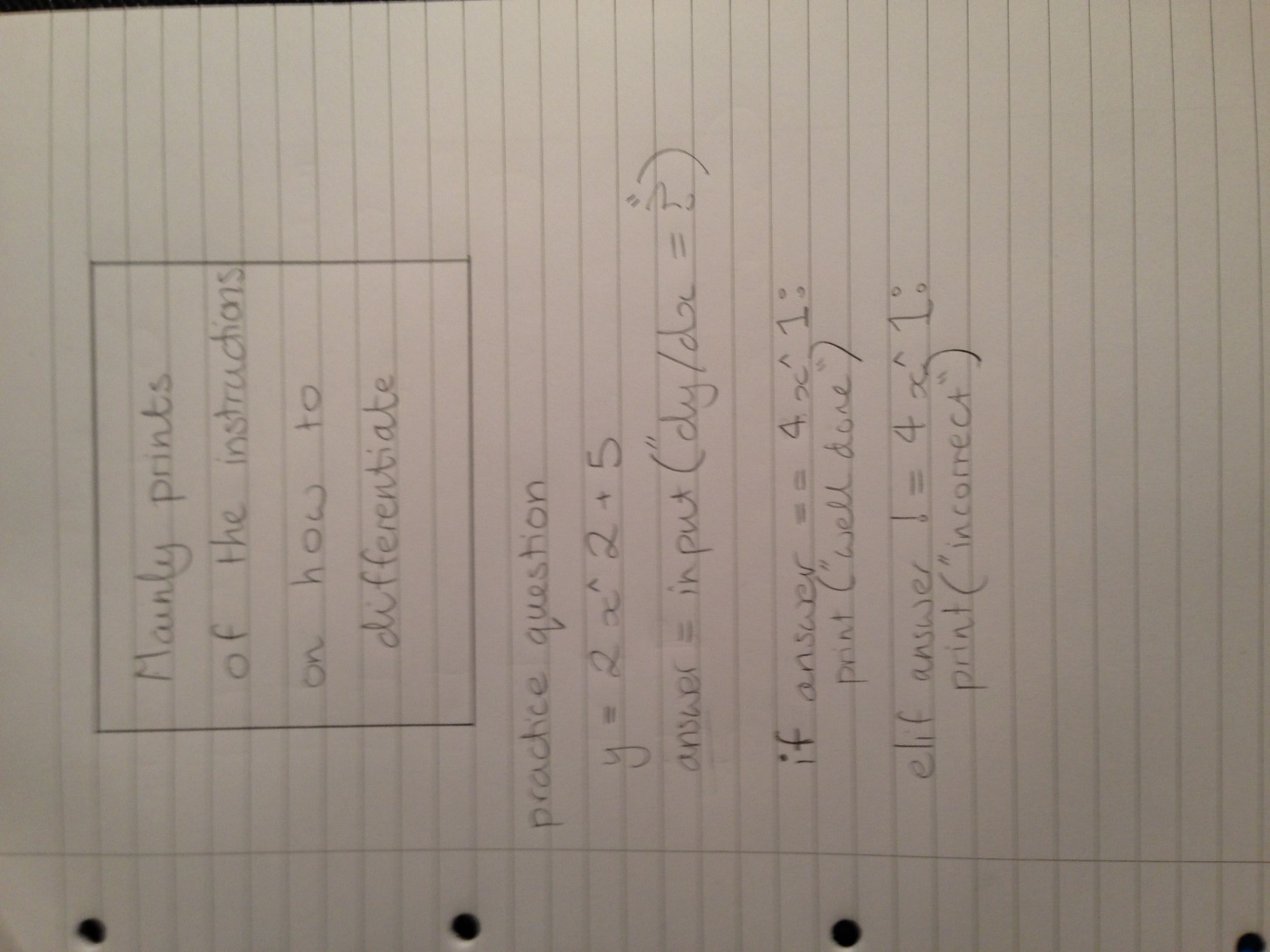
The user will be able to make a selection of what part of the program they would like to use. The options will range from 1 to 6, with 0 being at the end if the user wants to exit the program.

When 0 is inputted, the while loop will break and the program will end.



Introduction to differentiation

This section will mainly be prints, if and elif statements. This part of the program aims to teach the initial differentiation technique and because the equations are not being randomly generated, there is no need for complexity.



Questionnaire on easy difficulty

The questionnaire will be 10 questions long and will ask the user to find the derivative of a randomly generated linear equation.

This section will require code to randomly generate a linear equation. The user will need to input the derivative correctly to get the right answer.

This means I will also have to develop a code which find the derivative of the equation so that it matches the answer the user puts in. If the equation the user puts in matches the one solved by the code then they’ve got it correct, otherwise they’ve got it wrong.

I also want to put in a score system which counts how many correct answers the user gets out of 10 questions.

Questionnaire on intermediate difficulty

The questionnaire will be 10 questions long and will ask the user to find the derivative of a randomly generated nonlinear equation.

This section will require code to randomly generate a nonlinear equation. The user will need to input the derivative correctly to get the right answer.

This means I will also have to develop a code which find the derivative of the equation so that it matches the answer the user puts in. If the equation the user puts in matches the one solved by the code then they’ve got it correct, otherwise they’ve got it wrong.

I also want to put in a score system which counts how many correct answers the user gets out of 10 questions.

This will require more variables than the easy questionnaire as the user will be working with more terms.

Questionnaire on hard difficulty

The questionnaire will be 10 questions long and will ask the user to find the derivative of a randomly generated a more complex nonlinear equation.

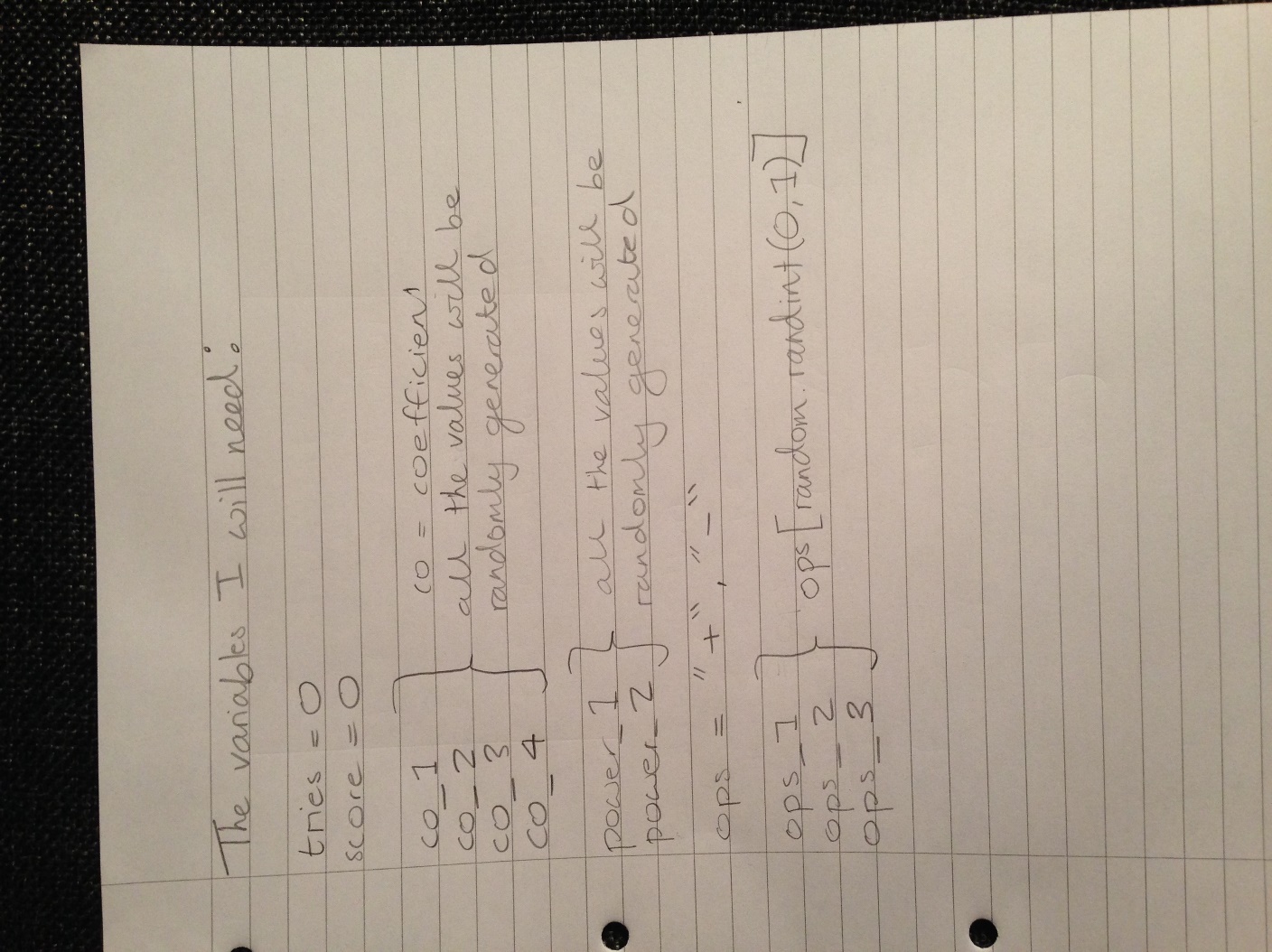
This section will require code to randomly generate a nonlinear equation. The user will need to input the derivative correctly to get the right answer.

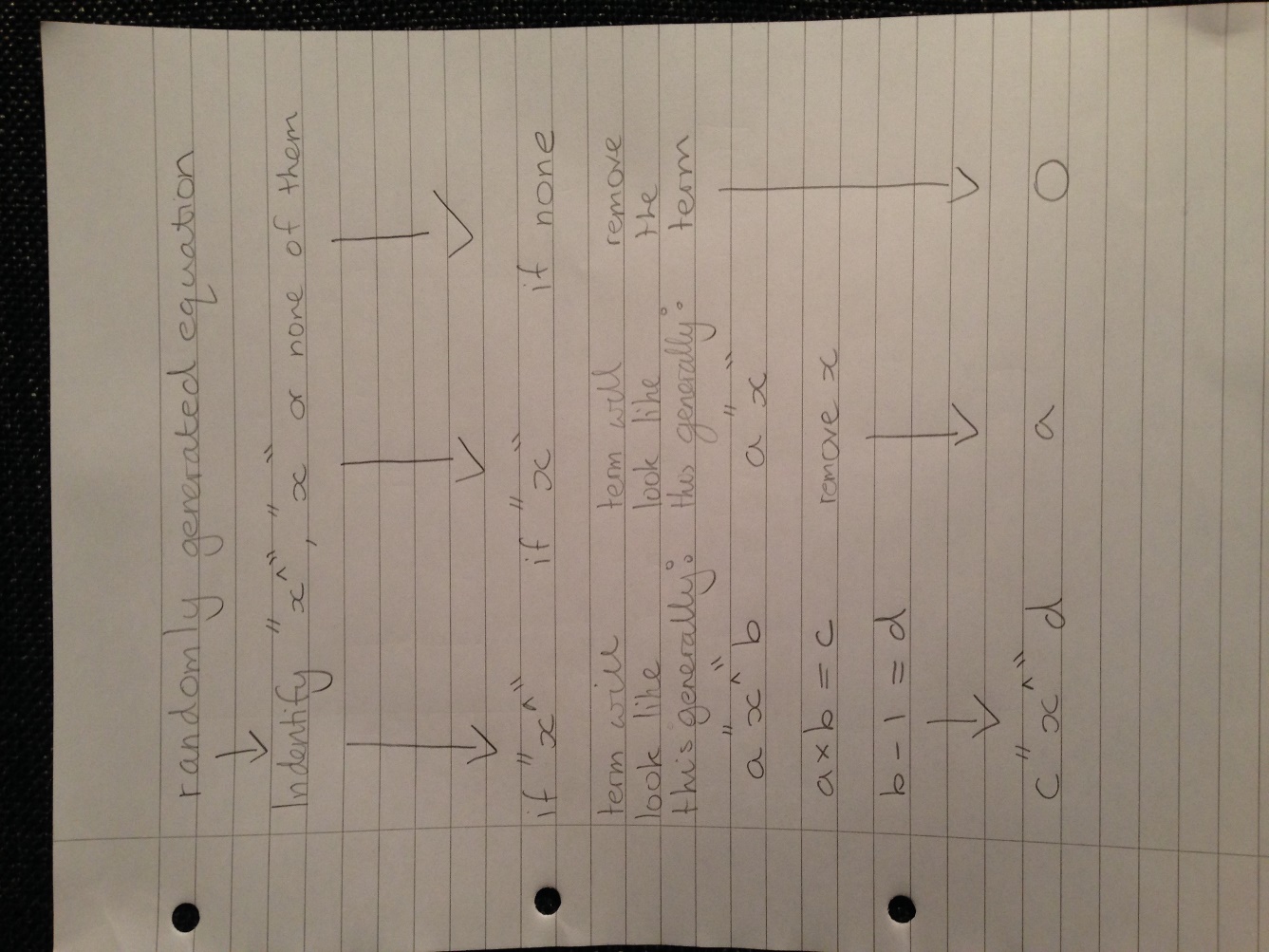
This means I will also have to develop a code which find the derivative of the equation so that it matches the answer the user puts in. If the equation the user puts in matches the one solved by the code then they’ve got it correct, otherwise they’ve got it wrong.

I also want to put in a score system which counts how many correct answers the user gets out of 10 questions.

This will require more variables than the easy questionnaire and the intermediate questionnaire as the user will be working with more terms.

**Below is the general plan for the questionnaire sections**

****

****

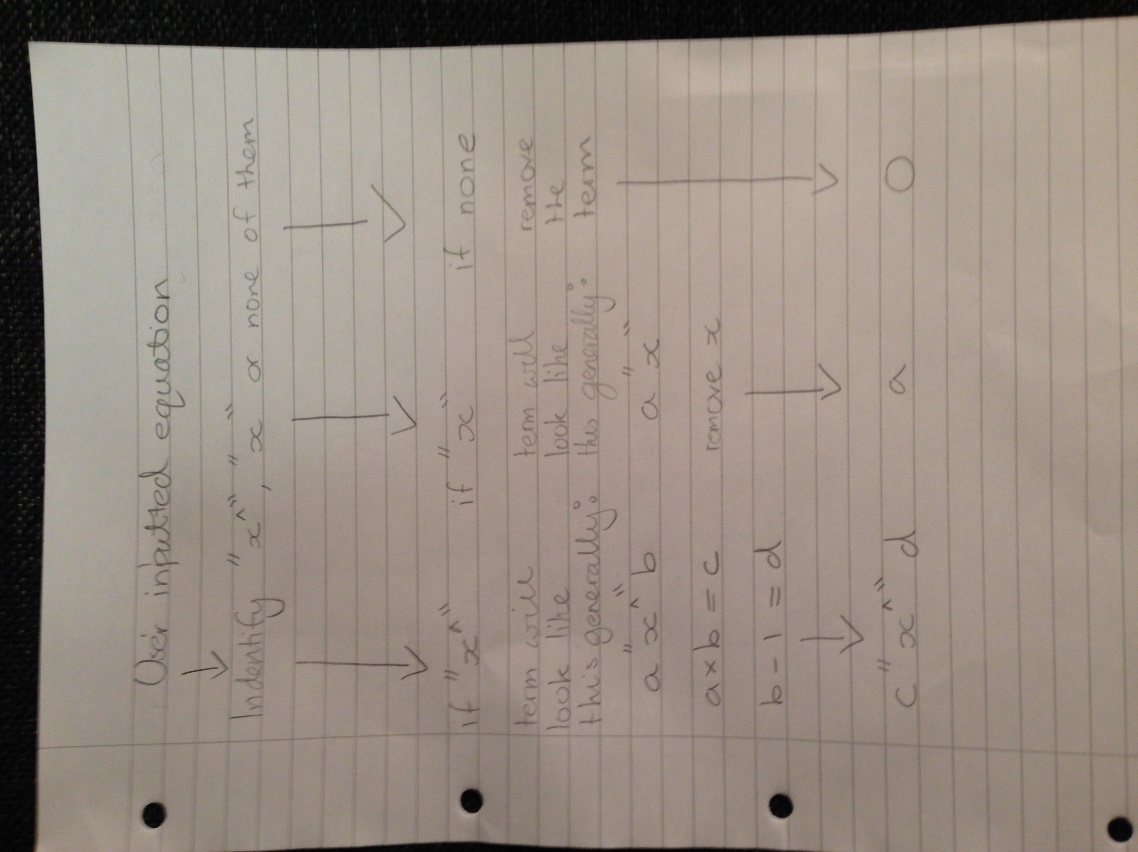
Differentiation calculator

The differentiation calculator will be able to find the derivative of a linear or nonlinear equation inputted by the user.

This will require a way of recognising particular terms which were inputted so that the right operation can be performed.

This would mean finding the coefficient of the x term, the x term itself, and the power of the x term. There will probably be the use of many if and elif statements so that particular terms can be recognised and the right operation is performed on them e.g. doing different operation on 5 x^ 3 compared to 5 x.

I was thinking of going about this by creating code which acts like a parsing tree. The code will identify the operators in the equation and then split up the equation so that we have parts instead. Each part will end up being an x term or a constant because we are splitting the string up between each operator. This will turn the string into a list.

 Then the if and elif statements will become active as they go through the list, identifying each term and then performing the right operation on it and then appending the part answer into another list.

This other list will then be converted back into a string, and present the derivative of the equation inputted by the user.

This is similar to the questionnaire code, except generating a random equation, the user has to put in the equation.

**Testing**

|  |  |  |  |
| --- | --- | --- | --- |
| **Section** | **Task** | **Does it work?** | **Screenshot** |
| While loop | Does the while loop work? | Yes |  |
| Menu | Does the 1st choice execute the right instruction | Yes |  |
| Does the 2nd choice execute the right instruction | Yes |  |
| Does the 3rd choice execute the right instruction | Yes |  |
| Does the 4th choice execute the right instruction | Yes |  |
| Does the 5th choice execute the right instruction | Yes |  |
|  | Does the 6th choice execute the right instruction | Yes |  |
| Does the input “0” break the while loop | Yes |  |
| How to use this program | Does the print code work | Yes |  |
| Introduction  To  Differentiation | Do all print functions work | Yes |  |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Do the practice equations work | Yes |  |
| Questionnaire  Easy | Does the random equation generator work | Yes |  |
|  | Does user input work correctly | Yes |  |
| Questionnaire  Intermediate | Does the random equation generator work | Yes |  |
| Does user input work correctly | Yes |  |
| Questionnaire  Hard | Does the random equation generator work | Yes |  |
| Does user input work correctly | Yes |  |
| Differentiation  Calculator | Can it solve ‘x^ ‘terms | Yes |  |
| Can it solve ‘x’ terms | Yes |  |
| Can it solve constants | Yes |  |
| Can it solve a complex nonlinear equation | Yes |  |
| Can it solve x terms with negative coefficients | Yes |  |
| Can it solve floating point values | Yes |  |

**What went well and what didn’t go so well**

What went well

Overall, I was pleased with the outcome of my program. It seems to function very well.

I believe with this program, someone could pick up the basic of differentiation and become quite good at it.

I’m particularly pleased with the differentiation calculator I created. It can do potentially and infinite amount of terms. It can also do floating points as well as integers. It can also solve x terms with negative coefficients.

The code use to create the differentiation calculator helped me with creating the questionnaires also. It uses the same base code, with a few modifications so that a random equation is generated rather than user-inputted one.

I think the layout is also quite nice as I used \n and \t multiple times for prints. Text is nicely separated to prevent it looking like blocks.

What didn’t go so well

I was a bit disappointed that I couldn’t get the differentiation code to solve x terms with negative powers. I developed a code which could handle x terms with negative powers, but implementing it in entire program became problematic so I left it out.

I would of also have liked to make the tutorial part of the section more detailed, as it seems a bit basic. One of the way I would improve this is by importing images of graphs to visually demonstrate differentiation.