Analysis

Background

I am going to create a program which allows the users to learn and perfect their knowledge tree traversals and also Reverse Polish notation (RPN) and also be able to answer questions in a quiz for each topic. The program will target students studying A-level Computer Science who are with the AQA exam board. The program will teach the user about how to: trace trees using pre-order, post-order and in-order; convert simple expressions in infix form to Reverse Polish notation (RPN). In each of these sub topics the program will have animations where appropriate to show how exactly the different traversals and RPN conversion work. As well as this, in each of the sub topics, the animations are linked with key facts and information about the topic and will state what they are all used for and how they’re linked all together. Also, there will be a section where the user can teach themselves the topic by having many examples and also visual representation of how each work with good description alongside it with definitions where appropriate to ensure they learn everything needed according to the specification of the AQA exam board. The user would then be able to take a quiz on each topic whenever they feel like they’re ready.

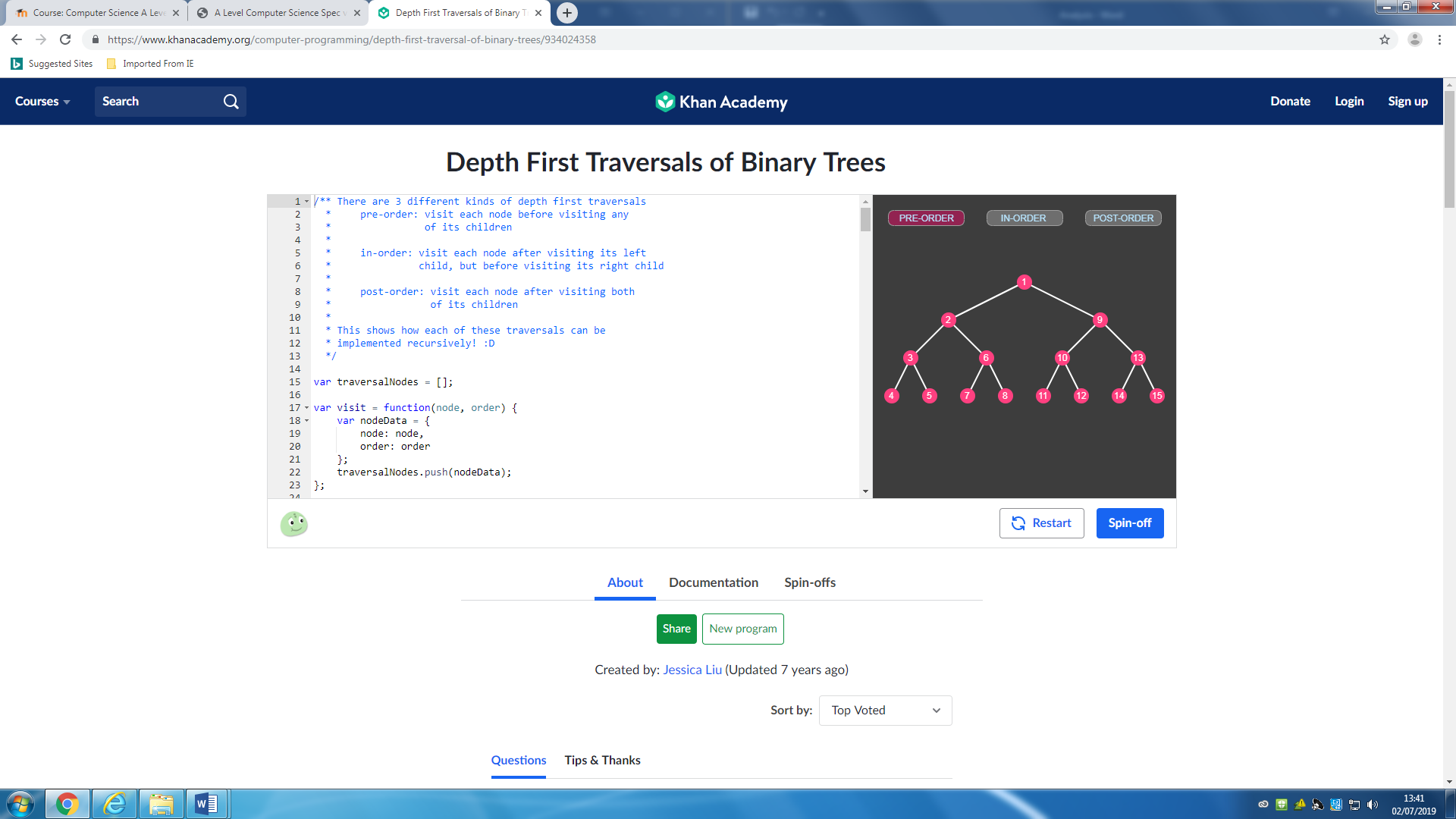
The quiz of each of the topics will generate 10 questions for each topic (questions will be stored in a database alongside their correct answers as well as the answers to choose from). Each question will be a multiple choice where the user has four answers to choose from and they must select the one they think is correct. The user would have 10 minutes to answer all the questions and then the program will mark their work. Once it’s marked, the program stores in a database whether the user got the questions right or wrong. As the user re does the quiz, new questions are generated giving questions based on what the user got wrong and having less questions on what they’ve previously got correct to allow them to improve on their weaker points. The user could choose to re try some more quizzes until they’re confident or they could re look at the revision tools and try the quiz a later time. Finally, the program will also state the total number of questions that the user has got correct so that they know how much more they need to revise. Once all questions are correctly answered the program will just give the questions again which allows the user to come back at a later date to do some extra revision to ensure they still remember everything.

As the program is launched the user will be able to either create a new account or log in to an already existing user. They will have to create a user name that is not already in use and will have to generate a password to keep their account only accessible to them. Each of the user’s information will be stored in a database.

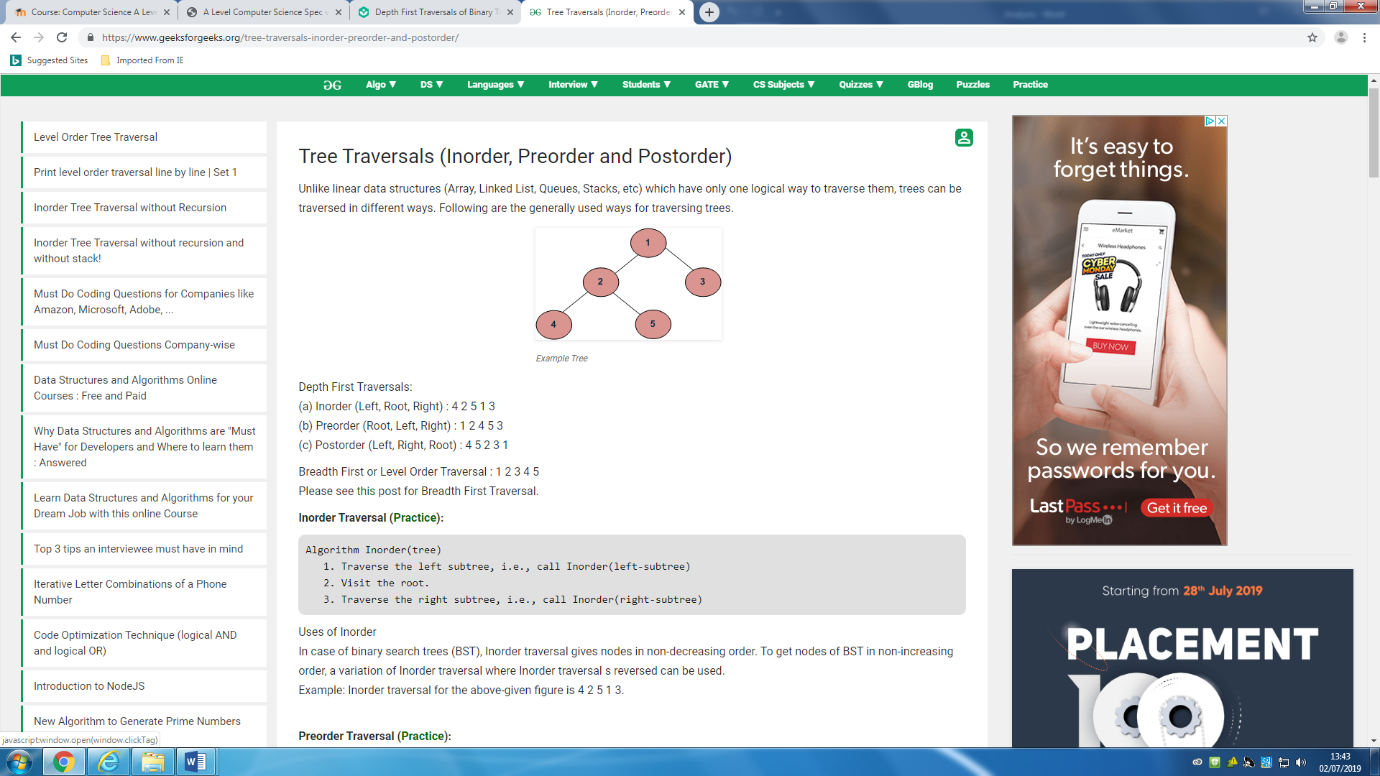
Research

I have found websites which have some concepts that I’d like to include in my project but also have missing things that will help me create a project that allows the user to learn from.

I have found some websites which have part of what I’m planning to create but are also missing things that I wish to include in my program.

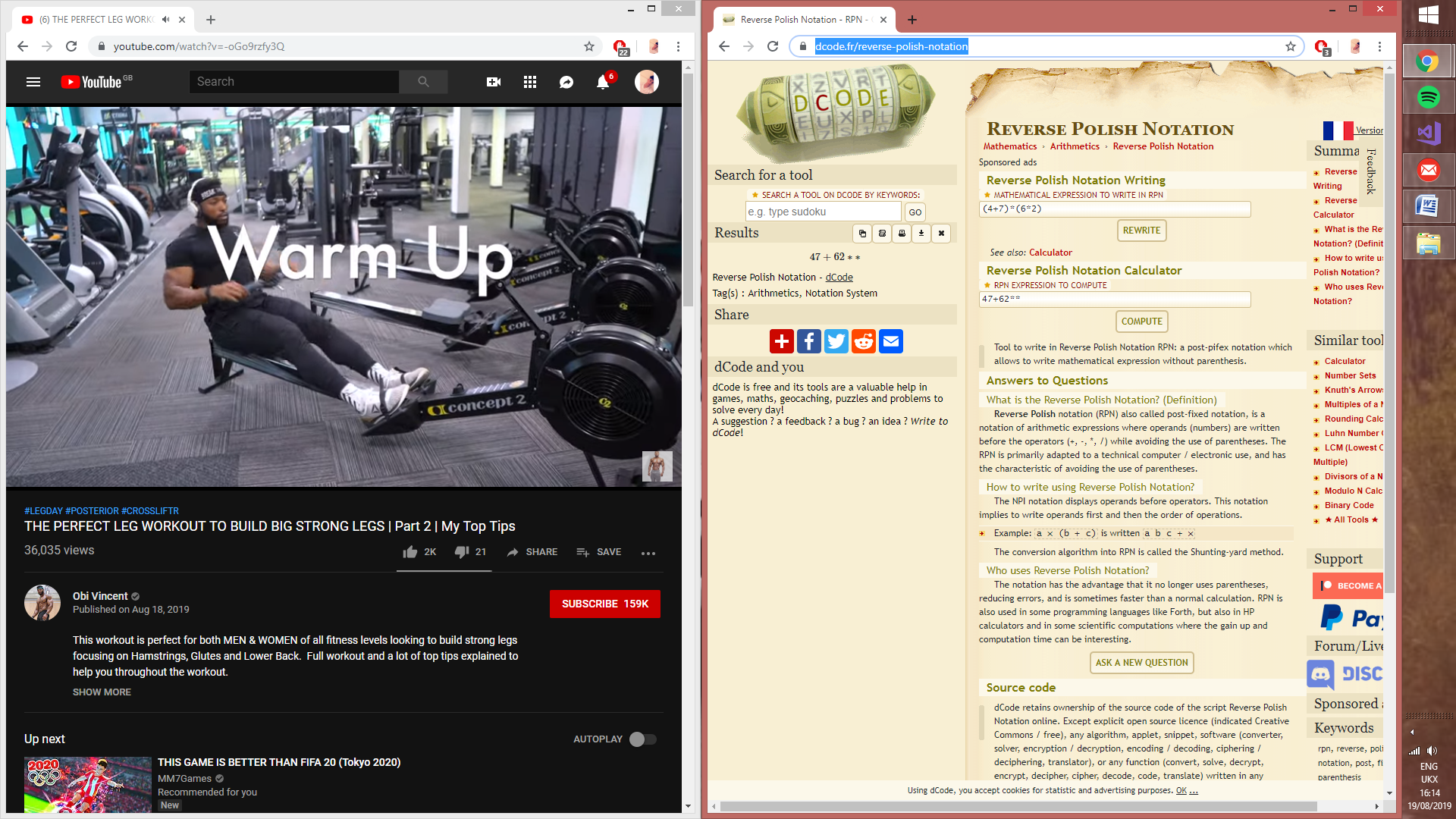


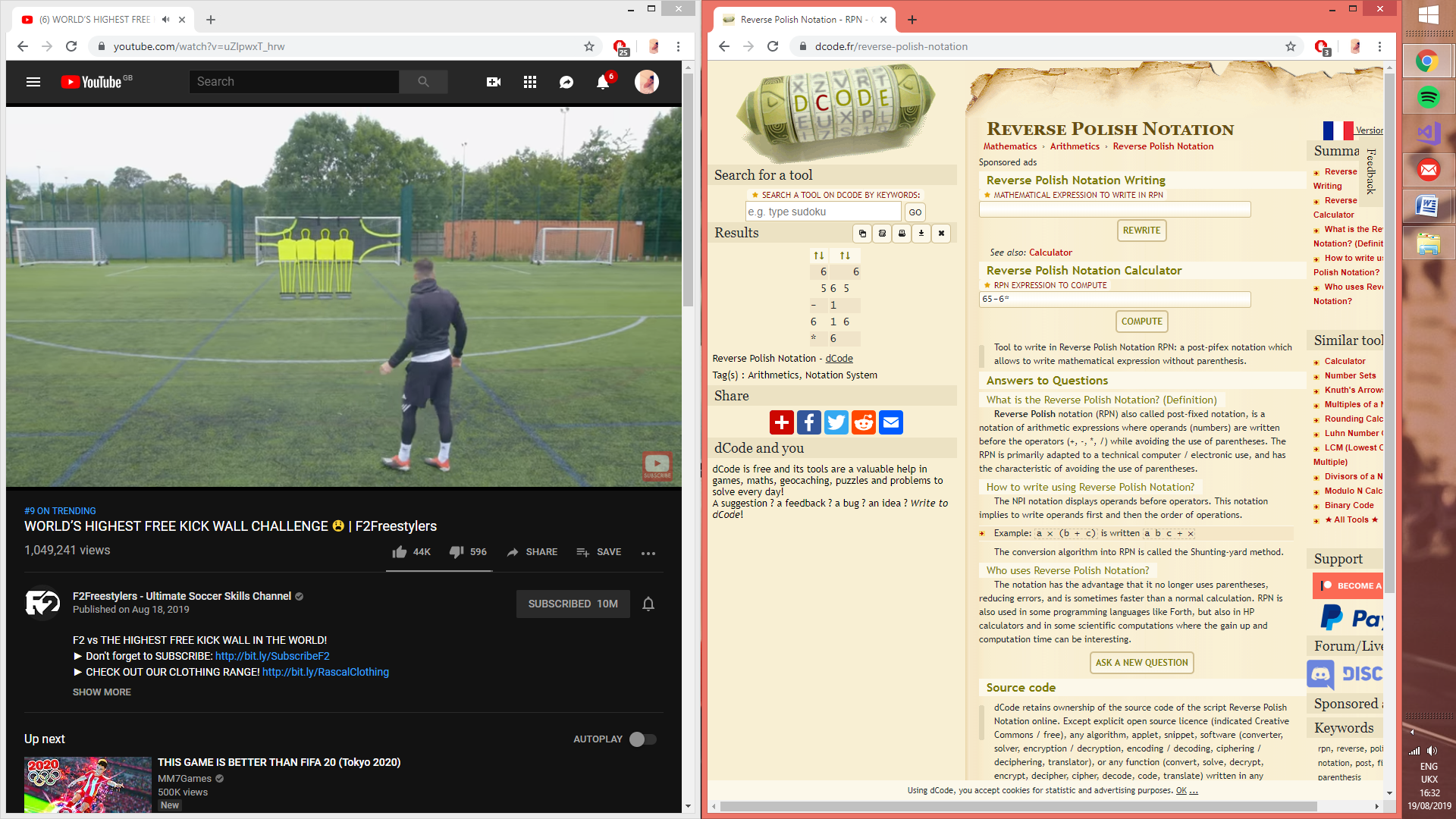
This first website has a section that has a visual animation of each of pre-order, in-order and post-order traversal (Depth First). It allows the user to click on which ever one they’d like and the it will run through the animation showing how that specific traversal works. It highlights each of the dots that are visited when they are traversed and also numbers them so the user can know which order they were traversed in. Also, this website has a section on the page that has the basic algorithms of each of those three traversals stating how the each work. This helps the user understand the visual animation more as they can follow along using the algorithms given. The mix of both the animation and algorithms is helpful for the user to understand the concept and how the traversals works.



In the second website it has diagram of a tree followed by how each depth first and breadth first would be traversed. It has the order the numbers are traversed for each. It then singularly gives an in depth algorithm for each pre-order, in-order and post order traversal to allow the user to follow the tree themselves and then compare with the answers at the top of the website. It also gives a description of what each of the traversals can be used for to allow the user to use a more practical way of using the traversals thus helping them learn how it each works.

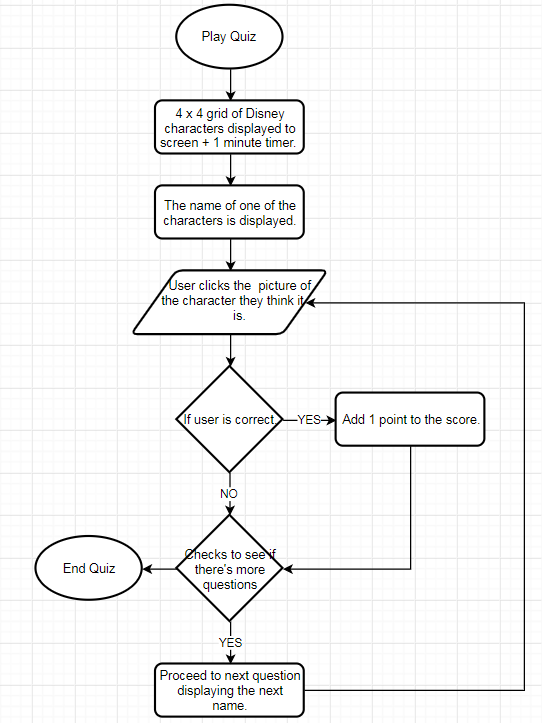
I’d like to use a few ideas from both websites as they each have good points which could be useful in providing the user of my program to learn about how to traverse the different trees in the different ways. I’d like to use the idea of having an animation of the different traversals to allow a visual representation of the traversals but then I’d also want to be able to have in depth algorithms written up for the user to use and to learn how it all works. The mixture of both of these websites will give a good program that can be used for learning as they can chose which way they’d like to learn as it allows more options.





This website allows the user to either convert from infix to prefix or then to calculate what the answer of a prefix notation equation would give. The user can input into the “Reverse Polish Notation Writing” box a mathematical equation (infix) and then in the “Results” section it would output the infix notation in prefix notation. The other box allows the user to input a prefix notation equation and then it would show a stack and how the numbers are inputted and then gives out the answer to the equation.

I would like to include parts of this website into my program which would help the user improve their understanding of Reverse Polish Notation (RPN). I like the ideas of being able to have my program allow the user to input either infix or prefix notation but then instead of just giving the answer in the opposite form, it also shows the steps used to get to the correct answer. On top of this, I would like the program to not only convert but to also have a good explanation how the stacks work to convert followed by how the conversion of infix to prefix makes calculating equation easier.



This is a flow chart of a website which has a quiz similar to what I’d like to include in my program. It starts by showing multiple picture of Disney characters and also displaying the name of one of the characters and the user must select which character they think it is. If the user chooses correctly their score is updated by one. Once every character name has been displayed the quiz stops and allows the user to have another go but using different characters. I’d like to include something similar into my program using the multiple choice options popping up but instead of using pictures it would be words/ statements which the user can choose from. Also, when the users finishes, the program can generate another quiz but instead of having a pre made quiz the program generates one based off questions the users has not seen or has got incorrect in previous attempts.

User Requirements

Revision Section:

1. User selects which topic they’d wish to revise.
   1. User clicks either the “Tree Traversing” or the “Reverse Polish Notation” button.
      1. If “Tree Traversing” is pressed a new form loads up for the section on tree traversals.
      2. If “Reverser Polish Notation” is pressed a new for loads up for the section on RNP.
      3. The user now either decides to do a quiz or then do some revision by clicking on the suitable button which takes the user to the form which it’s held in.

Tree Traversal Revision:

1. On this form there will be a section describing all the information needed with examples.
2. There will also be a section where the user can physically see how each pre-order, in-order and post-order traverse.
   1. Pre-order algorithm:
      1. Visit root node.
      2. Recursively traverse left subtree.
      3. Recursively traverse right subtree.
   2. In-order algorithm:
      1. Recursively traverse left subtree.
      2. Visit root node.
      3. Recursively traverse right subtree.
   3. Post-order algorithm:
      1. Recursively traverse left subtree.
      2. Recursively traverse right subtree.
      3. Visit root node.
3. As well as this, they can input a list of numbers or words and then the program will do each algorithm for pre-order, in-order and post-order and output the answer for each.

Reverse Polish Notation (RPN) Revision:

1. On this form there will be a section describing all the information needed with examples.
2. There is also a section where the user can input a string of numbers and operands in either infix notation or prefix notation and then the program will calculate the answer in the opposite form.
   1. If the user wishes to calculate the answer of a postfix notation number then a stack is used to help calculate the answer to the equation.
      1. A number just means to push this number onto the stack.
      2. If it is an operand that means to pop 2 elements from the stack, does the operation, and the pushes the result back.
   2. If the user wishes to convert from infix to postfix notation then an expression string and operator stack is used to convert it.
      1. The program reads through the string and every time it encounters a number it’s added to the expression string.
      2. When it then encounters an operator it will get push onto the operator stack.
      3. It then repeats until all the numbers and operands are used.
      4. It then removes two numbers from the expression string and then pops one operand from the operator stack.
      5. It then pop two numbers again from the expression string, if there’s less than two then it pops what’s remaining.
      6. The process repeats until all numbers and operands are outputted.

User Log in or Register:

1. User writes their username and password into the correct spot.
   1. The program then reads the Login database.
   2. It then checks to see if the username inputted by the user is stored in the database.
      1. If the username is stored, it then checks to see if the password inputted by the user matches the password stored in the database.
         1. If the username and password both match then it allows the user to login.
         2. If the username and password don’t match then the program makes a message box pop up saying “INCORRECT USERNAME OR PASSWORD” and allows the user to retry.
      2. If the username inputted by the user doesn’t match, then the program makes a message box pop up saying “INCORRECT USERNAME OR PASSWORD” and allows the user to retry.

Quiz Section:

1. The user can either choose to do multiple choice questions for tree traversal or reverse polish notation.
2. In each quiz attempt the user gets 10 minutes to answer 10 questions which are all stored in a database.
3. The program selects 10 questions stored inside the database that the user has got incorrect.
   1. If there are less than 10 wrong questions remaining to answer, then the program would select questions that have previously been answered correct to fill up the 10 questions.
4. The program marks whether the user got the answers correct or incorrect.
   * 1. The program then stores the answers that they got correct so next time the user re does a quiz on that topic it only gives questions they’ve got incorrect or never seen so that they can improve their knowledge in areas they’re not good in.
5. Each user has their scores saved inside a database so when they log in they can keep their progress and continue from where they stopped.