**PLANNING A COMPLEX ALGORITHM**

**DESIGN THE ROUTINE**

CHECK PREREQUISITES

Define the problem

*We want to add a Dynamically styled nav that jumps to Anchors on page*

Information the routine will hide

*All previously calculated routines will remain the same and work in the background. Country colour values will be hidden in classes and arrays.*

Inputs to the routine

*Button Press*

Outputs from the routine

*Adds a nav to the top of page.*

Pre-conditions

*All Data is calculated and inputted. The Information is already set up into their appropriate divs, and all DIVs are targetable by either class or id.*

Post-conditions

*Adds a nav element with buttons to jump down to specific anchors at the top of page.*

Name the Routine

*Nav Bar Generation*

Decide how to test the routine

1. *A visual check first: is the navbar there? Does it have Buttons?*
2. *Do the buttons jump to the corresponding element on the webpage?*

Research functionality available in standard libraries

*The majority of this assignment is working with HTML DOM, which is readily available within JS Libraries. I will have to Learn how to generate nav, button and anchor elements*

Think about error handling

*Generally, with this dynamic generation Physical errors are able to be identified and fixed as data will not return on the page. Otherwise all errors will be caught by the console. Since the button will only call the function on click, its possible the webpage will load but will cause errors when the button is pressed.*

Think about efficiency

*Looping will be used to generate most of the data. This approach will avoid the Hard Coding of every single result.*

Research algorithms & data types

*Dom Elements have a different syntax of assigning variables and attributes than regular html. Assigning them to a variable then using functions to assign these attributes seems to be the simplest way of doing it.*

**WRITE PSEUDOCODE**

1. Think about the data
2. Check the pseudocode
3. Try ideas in pseudocode

**CODE THE ROUTINE**

1. Write the declaration
2. Turn pseudocode into comments
3. Fill in code below comments
4. Check if code can be factored

**CHECK THE CODE**

1. Mentally check for errors
2. Step through in Debugger
3. Test the code
4. Remove errors in the code
5. Clean up