**PLANNING A COMPLEX ALGORITHM**

**DESIGN THE ROUTINE**

CHECK PREREQUISITES

Define the problem

Create a function to compare the app guess number and user’s response

Information the routine will hide

1. The process of comparing guess number and user’s response

Inputs to the routine

1. User’s response
2. The guess number of app

Outputs from the routine

A message representing app’s reply

Pre-conditions

1. The app must exist
2. The app should have a counter
3. The app has guessed a number
4. The user should have given response

Post-conditions

A message is displayed on the page

Name the Routine

compareNumber

Decide how to test the routine

Test the routine with different user’s responses 5 times.

Research functionality available in standard libraries

Literal templates

<https://developer.mozilla.org/en-US/docs/Web/JavaScript/Reference/Template_literals>

Think about error handling

1. The message doesn’t show
2. The counter shows a wrong number

Think about efficiency

The routine can be used unlimited times

Research algorithms & data types

Algorithm:

1. The counter adds 1
2. Analyse user’s response
   1. Create an empty string as the app’s reply
   2. If the user’s response is ‘Correct!’

The app’s reply is the time it takes

* 1. Else calculate a new guess number based on the user’s reply
  2. Use the new guess number to check whether the user’s reply is honest
     1. If the user is honest, the app’s reply is the new guess number
     2. Else the app’s reply is a complain
  3. Return the app’s reply

1. The html page shows the app’s reply

**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