STRUCTURED ENGLISH : ALGORITHMS

- Generally speaking, there aren't any specific, definite rules that have to be followed while writing an algorithm in structured english.

- However, there are a set of guidelines that should be followed in order to write good structured English.

  They are as follows:

  - It should be very easy to "extract" variable names from the structured English
    That is, it should be easy to look at the structured English and decide what variables would probably be used in the actual program and what their identifier names (variable names) would be.

  - Types of processes should be easily identifiable:

    i) Input and output should be indicated by using words like "enter", "take as input", "print", "display", etc.

    ii) Selection should be indicated by using words like "if", "in the case that...", "choose", etc.

    iii) Any iteration should be indicated using words like "while true, do...", "repeat", etc.

STEPWISE REFINEMENT

- Stepwise refinement refers to recursively breaking down the steps of a structured list to achieve a more detailed and comprehensive list.

- This can be very helpful when writing algorithm for very large and/or complex problems, as it helps in breaking down large problems into smaller problems, which are then easier to solve and work through.

  ↳ It also allows for a more robust algorithm as every detailed and facet of the problem is being concerned

Example of stepwise refinement :

Initial Structured English
1. Enter time taken to run marathon in hours, minutes, and seconds
2. Calculate and store marathon time in seconds
3. Output marathon time in seconds

Breaking down the first step:

1. Enter time taken to run marathon in hours, minutes, and seconds
   - 1.1. Enter the hours
   - 1.2 Enter the minutes
   - 1.3 Enter the seconds
2. Calculate and store marathon time in seconds
3. Output marathon time in seconds

Further breaking down the sub-steps of the first step:

1. Enter time taken to run marathon in hours, minutes, and seconds
   - 1.1. Enter the hours
     - 1.1.1. Input value for hours
     - 1.1.2 Check input in range 2 to 8
     - 1.1.3 Reject if out of range or not a whole number, then reinput(1.1.1)
     - 1.1.4 Accept and store value in hours
   - 1.2 Enter the minutes
     - 1.2.1. Input value for minutes
     - 1.2.2 Check input in range 0 to 59
     - 1.2.3 Reject if out of range or not a whole number, then reinput(1.2.1)
     - 1.2.4 Accept and store value in minutes
   - 1.3 Enter the seconds
     - 1.3.1. Input value for seconds
     - 1.3.2 Check input in range 0 to 59
     - 1.3.3 Reject if out of range or not a whole number, then reinput(1.2.1)
     - 1.3.4 Accept and store value in seconds
2. Calculate and store marathon time in seconds
3. Output marathon time in seconds