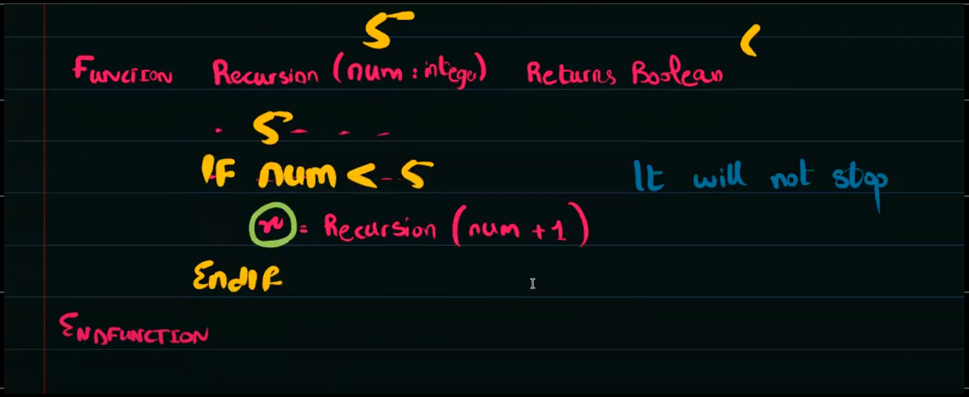
Recursion

Q- What is meant by recursion?

- When a procedure/function calls itself



```
                         5                        (
Function  Recursion (num : integer)  Returns Boolean
            . 5 - - -
            IF num < 5                          It will not stop
               n = Recursion (num + 1)
            EndIF
ENDFUNCTION
```

Q- What are the features of Recursion?

- General Case

- Base Case

- It calls itself

Note: The data from Recursion gets stored in stacks

If the function calls itself too many times, it runs out of stack space, stack overflow

Base Case: It is a conditional statement which is used to stop the recursion from continuing forever.

General Case: Is the place where recursive calls are made and it eventually reduces to the base case

Problem

Recursion	Iteration
Base Case	Loops
General Case	

Q- What is meant by recursive algorithm?

* see papersdock notes for past paper questions

- Has a base case
- Work towards a base case/general case
- Calls itself

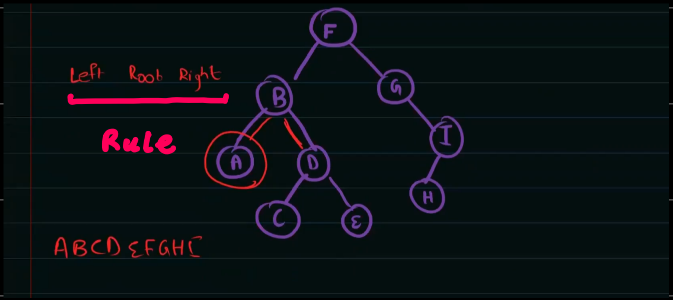
Unwinding: When the base case is reached, all the data stored in the stack gets popped out from top of stack

Q- Explain what a compiler has to do to implement Recursion?

- When the recursive call is made, all the values are put on the stack
- When the base case is met
- The algorithm unwinds
- The last set of values are taken off the stack in reverse order.

Traverse Tree

- Traverse Tree operation outputs the data item in alphabetic order



*Pseudocode

```
PROCEDURE TraverseTree ( RootPointer: INTEGER)
    IF Tree[ Rootpointer]. LeftPointer <> -1
      THEN
          CALL TraverseTree ( Tree [RootPointer]. LeftPointer)
    END IF
    OUTPUT Tree [Root]. Data
    IF Tree [ Root]. RightPointer <> -1
      THEN
          CALL TraverseTree ( Tree [Root]. RightPointer)
    END IF
END PROCEDURE
```

File 1: Q2, 9, 17, 39, 47

File 2: Q20, 28, 37