Structured English

Q- What is structured English?

Structured English is use of english language with the syntax of structured programming to communicate the design of comp. program to non-technical users by breaking it down into logical steps using English words.

Q- A programmer wants to write a program to search through a 1-D array of 100 elements.

- Count the number of elements that contains the string "Empty".

- output the number of elements containing "Empty" with a suitable message.

```
[DECLARE Array: ARRAY[1:100] OF STRING]   ← Array is already created, so do not declare.
DECLARE Index, ElementNum : INTEGER

ElementNum = 0
FOR Index = 1 TO 100
    IF Array[Index] = "Empty"
        THEN
            ElementNum = ElementNum + 1
    ENDIF
ENDFOR
PRINT "There are", ElementNum, "element which contain string 'Empty'."
```

```
ElementNum = 0          → Initialize ElementNum to zero
FOR Index = 1 TO 100    → Loop 100 times // Loop through all elements
    IF Array[Index] = "Empty" → Compare an element with 'Empty' in a loop
        THEN
            ElementNum = ElementNum + 1 → Increment ElementNum if condition true in a loop
    END IF
END FOR
PRINT "There are", ElementNum, "element which contain string 'Empty'."
```

↳ Output a message together with Elementnum outside the loop.

Q- A programmer wants to search through 1000 elements in an unsorted array to find 56 and output either the index position or the message "Not Found"

- Initialize count to 0
- loop through all the array
- Compare an element of array with 56 in the loop.
- If condition is true, then output its index position in the loop.
- Increment value of count if condition is true in a loop.
- Compare the variable count with 0, outside the loop.
- If condition is true, output 'not found'

- Unsorted means that data can be or can not be present in multiple elements of an array

```
Count=0
FOR Index=1 TO 1000
    IF A[Index]=56
    THEN
        Output Index
        Count = Count+1
    END IF
ENDFOR
IF Count=0
    THEN
        PRINT 'Not Found'
END IF
```

2 A team is designing a software system to monitor temperature in a process. To do this, the system needs to sample the temperature repeatedly. If the temperature exceeds a given threshold value, an alarm will sound.

The system is to be software-based. It will include a subroutine, `SampleTemp`, which samples the temperature and sets the alarm state to either ON or OFF.

The initial design stage will produce a prototype of `SampleTemp` with a user interface.
The structured English for this is:

1. IF the temperature does not exceed threshold value, SET alarm state to OFF
2. INPUT threshold value (to two decimal places)
3. INPUT sensor value (a whole number in the range 0 to 100)
4. MULTIPLY sensor value by conversion factor 1.135 to give temperature
5. IF temperature exceeds threshold value SET alarm state to ON
6. IF temperature exceeds threshold value OUTPUT message “Temperature Alarm”
7. IF temperature does not exceed threshold value OUTPUT message “Temperature OK”

```
AlarmState = False
PRINT "Enter threshold value correct to two decimal places: "
INPUT Threshold
PRINT "Enter Sensor value in range of 0 to 100: "
INPUT Sensor
Temperature = 1.135 * Sensor
IF Temperature > Threshold
    THEN
```

```
        *Alarm state = True
        PRINT "Temperature Alarm"
    ELSE
        *Alarm state = False
        PRINT "Temperature OK"
END IF
```

* (ON, OFF) ✗

- (True, False) ✓