

This test question isn't complete yet.

## Recursive to Iterative

1. The pseudocode given below converts decimal number to corresponding binary.

```
FUNCTION ToBinaryRecursive(Dec):  
    IF Dec = 0 THEN  
        RETURN 0  
    ELSE:  
        RETURN ToBinaryRecursive(Dec DIV 2)*10 + (Dec MOD 2)
```

i) Write program code to declare the function *ToBinaryRecursive()*.

ii) Write program code for the main program. The main program needs to run all three of the following function calls and output the result of each call:

- *ToBinaryRecursive(7)*
- *ToBinaryRecursive(0)*
- *ToBinaryRecursive(10)*

iii) Rewrite the function *Unknown()* as an iterative function, *ToBinaryIterative()*.

iv) The iterative function needs to be called three times with the same parameters as in part (ii). Amend the main program to perform those calls.

2. The pseudocode given below prints some pattern:

```
```markdown  
FUNCTION PrintPattern(N):  
    IF N < 0 THEN:  
        PRINT "The pattern broke."  
        RETURN  
    IF N = 0 THEN  
        RETURN 0  
    ELSE:  
        PRINT N, Pattern(N - 1), N
```

i) Write program code to declare the function *PrintPattern()*.

ii) Write program code for *IterativePattern()* that prints the same patterns.

iii) Call both versions of function with input 5, -1 and 9.

## Iterative to Recursive

1. The given pseudocode converts a binary number provided as a string to the string representation of the corresponding denary.

```
FUNCTION ToDenaryIterative(BinaryString):  
    Denary <- 0  
    Length <- LENGTH(BinaryString)  
    FOR i <- 0 TO Length - 1 DO  
        Denary <- Denary * 2 + INT(MID(BinaryString, i, i+1))  
    ENDFOR  
    RETURN Denary
```

- i) Write program code defining the function ToDenaryIterative().
- ii) Write program code for recursive version ToDenaryRecursive().
- iii) Test both versions of function with same value.

---

## Questions present below are from past papers.

---

### Recursive to Iterative

Question taken from 2021-Oct-Nov-41 (9608)

1. Given below is a recursively defined function.

```
FUNCTION Recursive(Num1, Num2 : INTEGER) RETURNS INTEGER  
    IF Num1 < 0 OR Num2 < 0 THEN  
        RETURN 1  
    ELSE  
        IF Num1 < Num2 THEN  
            Num1 <- Num1 - 2  
            RETURN 20 + 2 * Recursive(Num1, Num2)  
        ELSE  
            Num2 <- Num2 - 2  
            RETURN 10 + 2 * Recursive(Num1, Num2)  
        ENDIF  
    ENDIF  
ENDFUNCTION
```

- i) Write program corresponding to the above pseudo code.
- ii) Call the function with parameters 5 and 6.
- iii) Write program that defines function Iterative(Num1, Num2) that works similar to the function above but iteratively.
- iv) Call the Iterative() function with parameters 5 and 6.

---