

DEFINING AND CALLING A FUNCTION

```
FUNCTION <identifier>(<param1>:<datatype>, <param2>:<datatype>...) RETURNS <data type>
    <statements>
ENDFUNCTION
```

```
FUNCTION SumSquare (Number1:INTEGER, Number2:INTEGER) RETURNS INTEGER
    RETURN Number1 * Number1 + Number2 * Number2
ENDFUNCTION
```

```
OUTPUT "Sum of squares = ", SumSquare(10, 20)
```

DEFINING AND CALLING A PROCEDURE

```
PROCEDURE <identifier>(<param1>:<datatype>, <param2>:<datatype>...)  
    <statements>  
ENDPROCEDURE
```

```
PROCEDURE DefaultLine  
    CALL LINE(60)  
ENDPROCEDURE  
  
PROCEDURE Line(Size : INTEGER)  
    DECLARE Length : INTEGER  
    FOR Length ← 1 TO Size  
        OUTPUT '-'  
    NEXT Length  
ENDPROCEDURE  
  
IF MySize = Default  
    THEN  
        CALL DefaultLine  
    ELSE  
        CALL Line(MySize)  
ENDIF
```

FILE HANDLING

- Create or open a file
 - **OPENFILE** <File identifier> FOR <File mode>
 - The file identifier will be the name of the file with data type string. The following file modes are used:
 - READ for data to be read from the file
 - WRITE for data to be written to the file. A new file will be created and any existing data in the file will be lost
- Data is read from the using the READFILE command as follows:
 - **READFILE** <File Identifier>, <Variable>
- Data is written into the file using the WRITEFILE command as follows:
 - **WRITEFILE** <File identifier>, <Variable>
- Files should be closed when they are no longer needed using the CLOSEFILE command as follows:
 - **CLOSEFILE** <File identifier>

```
DECLARE LineOfText : STRING
OPENFILE FileA.txt FOR READ
OPENFILE FileB.txt FOR WRITE
READFILE FileA.txt, LineOfText
WRITEFILE FileB.txt, LineOfText
CLOSEFILE FileA.txt
CLOSEFILE FileB.txt
```

QUESTION

- Write a pseudocode to make sure the value input for a measurement meets the following rules:
 - the value is a positive number
 - a value is always input
 - the value is less than 1000

QUESTION

```
01 Flag ← FALSE
02 REPEAT
03     Total ← 0
04     FOR Counter ← 1 TO 4
05         OUTPUT "Enter a digit ", Counter
06         INPUT Number[Counter]
07         Total ← Total + Number * Counter
08         IF Number[Counter] = 0
09             THEN
10                 Flag ← TRUE
11         ENDIF
12     NEXT Counter
13     IF NOT Flag
14         THEN
15             Number[5] ← MOD(Total, 10)
16             FOR Counter ← 0 TO 5
17                 OUTPUT Number[Counter]
18             NEXT
19         ENDIF
20 UNTIL Flag
```

• Give the line number(s) for the statements showing

1. Totaling
2. Selection
3. Count-controlled loop
4. Post-condition loop
5. Logical operator

QUESTION

- The variable Saying is used to store string data in a program.
- (a) Write the pseudocode statement to declare the variable Saying
- (b) Write the pseudocode statements to:
 - allow a string to be input to the variable Saying
 - store the content of the variable Saying in a text file named "Quotations.txt"
 - make sure the text file is closed at the end of the algorithm

QUESTION

Pseudocode description

A loop that will iterate at least once.

A conditional statement to deal with many possible outcomes.

A loop that will iterate a set number of times.

A conditional statement with different outcomes for true and false.

Pseudocode statement

FOR...TO...NEXT

IF...THEN...ELSE...ENDIF

WHILE...DO...ENDWHILE

CASE...OF...OTHERWISE...ENDCASE

REPEAT...UNTIL