

1. Study the following pseudocode.

```

PROCEDURE FillTank()

    DECLARE Tries :
    INTEGER DECLARE Full :
    BOOLEAN

    Tries ← 1
    Full
    ←
    ReadSensor("F1") IF

    NOT Full

    THEN
        WHILE NOT Full AND Tries <
            4 CALL TopUp()
            Full ←
            ReadSensor("F1") Tries
                Tries + 1
        ENDWHILE
        IF Tries >
            3 THEN
                OUTPUT "Too many
                attempts" ELSE
                OUTPUT "Tank now full"
            ENDIF
        ELSE
            OUTPUT "Already full"
        ENDIF
    ENDIF

END PROCEDURE

```

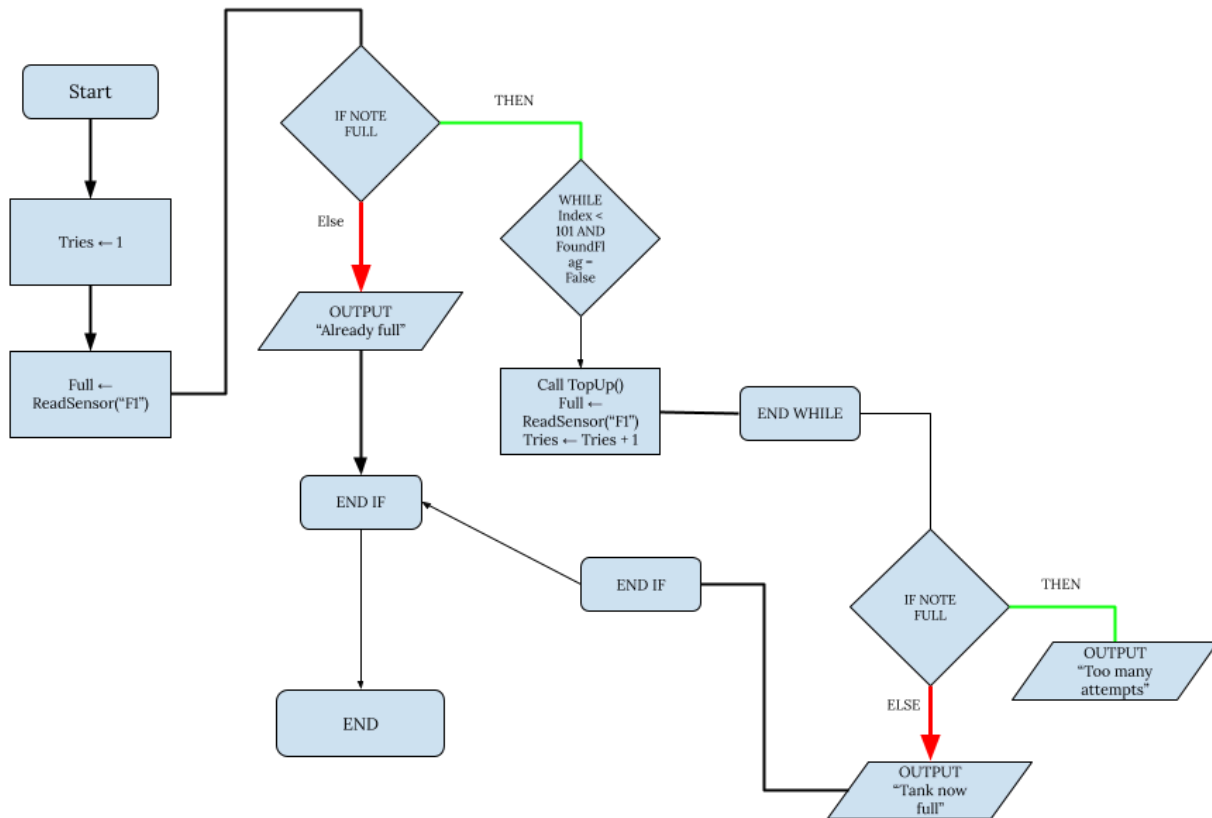
(a) (i) The pseudocode includes features that make it easier to read and understand.

State **three** such features.

Answer for a)

- The statements are each given a separate block (level of indentation) so it's easier to see the code related to each of them
- It uses everyday language rather than a programming language or machine code.
- Descriptive names for variables and the functions

- (ii) Draw a program flowchart to represent the algorithm implemented in the pseudocode. Variable declarations are not required in program flowcharts.



(b) (i) Programming languages support different data types.

Complete the table by giving a suitable data type for each example value.

Example value	Data type
43	
TRUE	
- 273.16	
"- 273.16"	

[4]

Answer for b) i)

- Integer
- Boolean
- Real
- String

(ii) Evaluate each expression in the following table.

If an expression is invalid then write 'ERROR'.

Refer to the **Appendix** on page 18 for the list of built-in functions and operators.

Expression	Evaluates to
RIGHT("Stop", 3) & LEFT("ich", 2)	
MID(NUM_TO_STRING(2019), 3, 1)	
INT(NUM_TO_STRING(-273.16))	
INT(13/2)	

[4]

Answer ii)

- topic
- 920
- "-273.16"
- 6

Appendix

Built-in functions (pseudocode)

Each function returns an error if the function call is not properly formed.

MID(ThisString : STRING, x : INTEGER, y : INTEGER)
 RETURNS STRING
 returns a string of length y starting at position x from ThisString

Example: MID("ABCDEFGH", 2, 3) returns "BCD"

LENGTH(ThisString : STRING) RETURNS INTEGER
 returns the integer value representing the length of ThisString

Example: LENGTH("Happy Days") returns 10

LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING
 returns leftmost x characters from ThisString

Example: LEFT("ABCDEFGH", 3) returns "ABC"

RIGHT(ThisString: STRING, x : INTEGER) RETURNS STRING
 returns rightmost x characters from ThisString

Example: RIGHT("ABCDEFGH", 3) returns "FGH"

INT(x : REAL) RETURNS INTEGER
 returns the integer part of x

Example: INT(27.5415) returns 27

NUM_TO_STRING(x : REAL) RETURNS STRING
 returns a string representation of a numeric value.

Note: This function will also work if x is of type INTEGER

Example: NUM_TO_STRING(87.5) returns "87.5"

STRING_TO_NUM(x : STRING) RETURNS REAL
 returns a numeric representation of a string.

Note: This function will also work if x is of type CHAR

Example: STRING_TO_NUM("23.45") returns 23.45

Operators (pseudocode)

Operator	Description
&	Concatenates (joins) two strings Example: "Summer" & " " & "Pudding" produces "Summer Pudding"
AND	Performs a logical AND on two Boolean values Example: TRUE AND FALSE produces FALSE
OR	Performs a logical OR on two Boolean values Example: TRUE OR FALSE

	produces TRUE
--	---------------