Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/23**

Paper 2 Fundamental Problem-solving and Programming Skills **October/November 2021**

INSERT **2 hours**

5427399501-I

INFORMATION

- This insert contains all the resources referred to in the questions.
- You may annotate this insert and use the blank spaces for planning. **Do not write your answers** on the insert.

This document has **4** pages.

DC (PQ) 222380

[Turn over

Note: An error occurs if a function call is not properly formed, or if the parameters are incorrect.

STRING Functions

`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"`

`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`

`LCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the lower case equivalent of `ThisChar`
Alphabetic characters that are not upper case are unchanged.

Example: `LCASE('W')` returns `'w'`

`UCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the upper case equivalent of `ThisChar`
Alphabetic characters that are not lower case are unchanged.

Example: `UCASE('a')` returns `'A'`

`TO_UPPER(ThisString : STRING) RETURNS STRING`
returns a string formed by converting all characters of `ThisString` to upper case.

Example: `TO_UPPER("Error 803")` returns `"ERROR 803"`

`TO_LOWER(ThisString : STRING) RETURNS STRING`
returns a string formed by converting all characters of `ThisString` to lower case.

Example: `TO_LOWER("JIM 803")` returns `"jim 803"`

`NUM_TO_STR(x : <data type1>) RETURNS <data type2>`
returns a string representation of a numeric value.

Note: `<data type1>` may be `REAL` or `INTEGER`
`<data type2>` may be `CHAR` or `STRING`

Example: `NUM_TO_STR(87.5)` returns `"87.5"`

`STR_TO_NUM(x : <data type1>) RETURNS <data type2>`
returns a numeric representation of a string.

Note: `<data type1>` may be `CHAR` or `STRING`
`<data type2>` may be `REAL` or `INTEGER`

Example: `STR_TO_NUM("23.45")` returns `23.45`

`IS_NUM(ThisString : STRING) RETURNS BOOLEAN`
returns the value `TRUE` if `ThisString` represents a valid numeric value.

Example 1: `IS_NUM("12.36")` returns `TRUE`
Example 2: `IS_NUM("-12.36")` returns `TRUE`
Example 3: `IS_NUM("12.3a")` returns `FALSE`

`ASC(ThisChar : CHAR) RETURNS INTEGER`
returns an integer value (the ASCII value) of `ThisChar`

Example: `ASC('A')` returns `65`

`CHR(x : INTEGER) RETURNS CHAR`
returns the character whose integer value (the ASCII value) is `x`

Example: `CHR(87)` returns `'W'`

NUMERIC Functions

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

Example: `INT(27.5415)` returns `27`

`RAND(x : INTEGER) RETURNS REAL`
returns a real number in the range `0` to `x` (**not** inclusive of `x`).

Example: `RAND(87)` could return `35.43`

OTHER Functions

`EOF(FileName : STRING) RETURNS BOOLEAN`
returns `TRUE` if there are no more lines to be read from file `FileName`

Note: The function will generate an error if the file is not already open in `READ` mode.

Note: An error occurs if an operator with a value of an incorrect type is used.

OPERATORS

`&`	Concatenates (joins) two strings Example: `"Summer" & " " & "Pudding"` evaluates to "Summer Pudding"
`AND`	Performs a logical `AND` on two Boolean values Example: `TRUE AND FALSE` evaluates to `FALSE`
`OR`	Performs a logical `OR` on two Boolean values Example: `TRUE OR FALSE` evaluates to `TRUE`
`NOT`	Performs a logical `NOT` on a Boolean value Example: `NOT TRUE` evaluates to `FALSE`
`MOD`	Finds the remainder when one number is divided by another Example: `10 MOD 3` evaluates to `1`
`DIV`	Finds the quotient when one number is divided by another Example: `10 DIV 3` evaluates to `3`