Cambridge Assessment International Education

Cambridge International AS & A Level

COMPUTER SCIENCE **9618/02**

Paper 2 Fundamental Problem-solving and Programming Skills **For examination from 2021**

SPECIMEN INSERT **2 hours**

0123456789

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. Blank pages are indicated.

Pseudocode Functions

An error is generated if the syntax is incorrect, or if a parameter type is incorrect.

`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"`

`LCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the lower case equivalent of `ThisChar`

If `ThisChar` is not an upper-case alphabetic character, it is returned unchanged.

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

`UCASE(ThisChar : CHAR) RETURNS CHAR`
returns the character value representing the upper case equivalent of `ThisChar`

If `ThisChar` is not a lower-case alphabetic character, it is returned unchanged.

Example: `UCASE('a')` will return `'A'`

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

Example: `TO_UPPER("Disk Error 27")` returns `"DISK ERROR 27"`

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

Non-alphabetic characters are unchanged.

Example: `TO_LOWER("ERROR - Password Invalid")` returns `"error - password invalid"`

`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("87.5")` returns `87.5`

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

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

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

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

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

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

`MOD(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER`
returns the integer value representing the remainder when `ThisNum` is divided by `ThisDiv`

Example: `MOD(10,3)` returns `1`

`DIV(ThisNum : INTEGER, ThisDiv : INTEGER) RETURNS INTEGER`
returns the integer value representing the whole number part of the result when `ThisNum` is divided by `ThisDiv`

Example: `DIV(10,3)` returns `3`

Pseudocode Operators

Operator	Description
`&`	Used to concatenate (join) two strings Example: `"Summer" & " " & "Pudding"` produces `"Summer Pudding"`
`AND`	Used to perform a logical `AND` on two Boolean values Example: `TRUE AND FALSE` produces `FALSE`
`OR`	Used to perform a logical `OR` on two Boolean values Example: `TRUE OR FALSE` produces `TRUE`
`NOT`	Used to perform a logical `NOT` on a Boolean value Example: `NOT FALSE` produces `TRUE`

BLANK PAGE