



Cambridge International AS & A Level

COMPUTER SCIENCE

9618/21

Paper 2 Fundamental Problem-solving and Programming Skills

May/June 2022

INSERT

2 hours

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.

Functions

Note: An error will be generated if a function call is not properly formed or if the parameters are of an incorrect type or an incorrect value.

String and Character 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 string "BCD"
LENGTH(ThisString : STRING) RETURNS INTEGER returns the integer value representing the length of string ThisString Example: LENGTH("Happy Days") returns 10
LCASE(ThisChar : CHAR) RETURNS CHAR returns the character representing the lower-case equivalent of ThisChar Alphabetic characters that are not upper case are returned unchanged. Example: LCASE('W') returns 'w'
UCASE(ThisChar : CHAR) RETURNS CHAR returns the character representing the upper-case equivalent of ThisChar Alphabetic characters that are not lower case are returned 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 : <datatype1>) RETURNS <datatype2> returns a string representation of a numeric value. Note: <datatype1> may be REAL or INTEGER, <datatype2> may be CHAR or STRING Example: NUM_TO_STR(87.5) returns "87.5"
STR_TO_NUM(x : <datatype1>) RETURNS <datatype2> returns a numeric representation of a string. Note: <datatype1> may be CHAR or STRING, <datatype2> may be REAL or INTEGER Example: STR_TO_NUM("23.45") returns 23.45

String and Character Functions

IS_NUM(ThisString : <datatype>) RETURNS BOOLEAN

returns the value TRUE if ThisString represents a valid numeric value.

Note: <datatype> may be CHAR or STRING

Example: IS_NUM("-12.36") returns TRUE

ASC(ThisChar : CHAR) RETURNS INTEGER

returns an integer value (the ASCII value) of character ThisChar

Example: ASC('A') returns 65, ASC('B') returns 66, etc.

CHR(x : INTEGER) RETURNS CHAR

returns the character whose integer value (the ASCII value) is x

Example: CHR(65) returns 'A', CHR(66) returns 'B', etc.

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

Date Functions

Note: Date format is assumed to be DD/MM/YYYY unless otherwise stated.

DAY(ThisDate : DATE) RETURNS INTEGER

returns the current day number from ThisDate

Example: DAY(04/10/2003) returns 4

MONTH(ThisDate : DATE) RETURNS INTEGER

returns the current month number from ThisDate

Example: MONTH(04/10/2003) returns 10

YEAR(ThisDate : DATE) RETURNS INTEGER

returns the current year number from ThisDate

Example: YEAR(04/10/2003) returns 2003

DAYINDEX(ThisDate : DATE) RETURNS INTEGER

returns the current day index number from ThisDate where Sunday = 1, Monday = 2, etc.

Example: DAYINDEX(12/05/2020) returns 3

SETDATE(Day, Month, Year : INTEGER) RETURNS DATE

returns a variable of type DATE with the value of <Day>/<Month>/<Year>

TODAY() RETURNS DATE

returns a variable of type DATE with the value set to the current date.

Text File 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.

Operators

Note: An error will be generated if an operator is used with a value or values of an incorrect type.

<code>&</code>	Concatenates (joins) two strings Example: <code>"Summer" & " " & "Pudding"</code> evaluates to <code>"Summer Pudding"</code> Note: May also be used to concatenate a <code>CHAR</code> with a <code>STRING</code>
<code>AND</code>	Performs a logical <code>AND</code> on two Boolean values Example: <code>TRUE AND FALSE</code> evaluates to <code>FALSE</code>
<code>OR</code>	Performs a logical <code>OR</code> on two Boolean values Example: <code>TRUE OR FALSE</code> evaluates to <code>TRUE</code>
<code>NOT</code>	Performs a logical <code>NOT</code> on a Boolean value Example: <code>NOT TRUE</code> evaluates to <code>FALSE</code>
<code>MOD</code>	Finds the remainder when one number is divided by another Example: <code>10 MOD 3</code> evaluates to <code>1</code>
<code>DIV</code>	Finds the quotient when one number is divided by another Example: <code>10 DIV 3</code> evaluates to <code>3</code>

Comparison Operators

<code>=</code>	Used to compare two items of the same type. Returns <code>TRUE</code> if the condition is true, otherwise returns <code>FALSE</code>
<code>></code>	Notes:
<code><</code>	- may be used to compare types <code>REAL</code> and <code>INTEGER</code> - may be used to compare types <code>CHAR</code> and <code>STRING</code> - case sensitive when used to compare types <code>CHAR</code> or <code>STRING</code> - cannot be used to compare two records.
<code>>=</code>	
<code><=</code>	Examples:
<code><></code>	<code>"Program" = "program"</code> evaluates to <code>FALSE</code> <code>count = 4</code> evaluates to <code>TRUE</code> when variable <code>count</code> contains the value <code>4</code>

Permission to reproduce items where third-party owned material protected by copyright is included has been sought and cleared where possible. Every reasonable effort has been made by the publisher (UCLES) to trace copyright holders, but if any items requiring clearance have unwittingly been included, the publisher will be pleased to make amends at the earliest possible opportunity.

To avoid the issue of disclosure of answer-related information to candidates, all copyright acknowledgements are reproduced online in the Cambridge Assessment International Education Copyright Acknowledgements Booklet. This is produced for each series of examinations and is freely available to download at www.cambridgeinternational.org after the live examination series.

Cambridge Assessment International Education is part of Cambridge Assessment. Cambridge Assessment is the brand name of the University of Cambridge Local Examinations Syndicate (UCLES), which is a department of the University of Cambridge.