



## 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

<p>LEFT(ThisString : STRING, x : INTEGER) RETURNS STRING</p> <p>returns leftmost x characters from ThisString</p> <p>Example: LEFT("ABCDEFGH", 3) returns "ABC"</p>
<p>RIGHT(ThisString : STRING, x : INTEGER) RETURNS STRING</p> <p>returns rightmost x characters from ThisString</p> <p>Example: RIGHT("ABCDEFGH", 3) returns "FGH"</p>
<p>MID(ThisString : STRING, x : INTEGER, y : INTEGER) RETURNS STRING</p> <p>returns a string of length y starting at position x from ThisString</p> <p>Example: MID("ABCDEFGH", 2, 3) returns string "BCD"</p>
<p>LENGTH(ThisString : STRING) RETURNS INTEGER</p> <p>returns the integer value representing the length of string ThisString</p> <p>Example: LENGTH("Happy Days") returns 10</p>
<p>LCASE(ThisChar : CHAR) RETURNS CHAR</p> <p>returns the character representing the lower-case equivalent of ThisChar</p> <p>Alphabetic characters that are not upper case are returned unchanged.</p> <p>Example: LCASE('W') returns 'w'</p>
<p>UCASE(ThisChar : CHAR) RETURNS CHAR</p> <p>returns the character representing the upper-case equivalent of ThisChar</p> <p>Alphabetic characters that are not lower case are returned unchanged.</p> <p>Example: UCASE('a') returns 'A'</p>
<p>TO_UPPER(ThisString : STRING) RETURNS STRING</p> <p>returns a string formed by converting all characters of ThisString to upper case.</p> <p>Example: TO_UPPER("Error 803") returns "ERROR 803"</p>
<p>TO_LOWER(ThisString : STRING) RETURNS STRING</p> <p>returns a string formed by converting all characters of ThisString to lower case.</p> <p>Example: TO_LOWER("JIM 803") returns "jim 803"</p>
<p>NUM_TO_STR(x : &lt;datatype1&gt;) RETURNS &lt;datatype2&gt;</p> <p>returns a string representation of a numeric value.</p> <p>Note: &lt;datatype1&gt; may be REAL or INTEGER, &lt;datatype2&gt; may be CHAR or STRING</p> <p>Example: NUM_TO_STR(87.5) returns "87.5"</p>
<p>STR_TO_NUM(x : &lt;datatype1&gt;) RETURNS &lt;datatype2&gt;</p> <p>returns a numeric representation of a string.</p> <p>Note: &lt;datatype1&gt; may be CHAR or STRING, &lt;datatype2&gt; may be REAL or INTEGER</p> <p>Example: STR_TO_NUM("23.45") returns 23.45</p>

## String and Character Functions

<p>IS_NUM(ThisString : &lt;datatype&gt;) RETURNS BOOLEAN</p> <p>returns the value TRUE if ThisString represents a valid numeric value.</p> <p><b>Note:</b> &lt;datatype&gt; may be CHAR or STRING</p> <p><b>Example:</b> IS_NUM("-12.36") returns TRUE</p>
<p>ASC(ThisChar : CHAR) RETURNS INTEGER</p> <p>returns an integer value (the ASCII value) of character ThisChar</p> <p><b>Example:</b> ASC('A') returns 65, ASC('B') returns 66, etc.</p>
<p>CHR(x : INTEGER) RETURNS CHAR</p> <p>returns the character whose integer value (the ASCII value) is x</p> <p><b>Example:</b> CHR(65) returns 'A', CHR(66) returns 'B', etc.</p>

## Numeric Functions

<p>INT(x : REAL) RETURNS INTEGER</p> <p>returns the integer part of x</p> <p><b>Example:</b> INT(27.5415) returns 27</p>
<p>RAND(x : INTEGER) RETURNS REAL</p> <p>returns a real number in the range 0 to x (<b>not</b> inclusive of x).</p> <p><b>Example:</b> RAND(87) could return 35.43</p>

## Date Functions

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

<p>DAY(ThisDate : DATE) RETURNS INTEGER</p> <p>returns the current day number from ThisDate</p> <p><b>Example:</b> DAY(04/10/2003) returns 4</p>
<p>MONTH(ThisDate : DATE) RETURNS INTEGER</p> <p>returns the current month number from ThisDate</p> <p><b>Example:</b> MONTH(04/10/2003) returns 10</p>
<p>YEAR(ThisDate : DATE) RETURNS INTEGER</p> <p>returns the current year number from ThisDate</p> <p><b>Example:</b> YEAR(04/10/2003) returns 2003</p>
<p>DAYINDEX(ThisDate : DATE) RETURNS INTEGER</p> <p>returns the current day index number from ThisDate where Sunday = 1, Monday = 2, etc.</p> <p><b>Example:</b> DAYINDEX(12/05/2020) returns 3</p>
<p>SETDATE(Day, Month, Year : INTEGER) RETURNS DATE</p> <p>returns a variable of type DATE with the value of &lt;Day&gt;/&lt;Month&gt;/&lt;Year&gt;</p>
<p>TODAY() RETURNS DATE</p> <p>returns a variable of type DATE with the value set to the current date.</p>

## 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.

&	Concatenates (joins) two strings Example: "Summer" & " " & "Pudding" evaluates to "Summer Pudding" Note: May also be used to concatenate a CHAR with a STRING
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

## Comparison Operators

=	Used to compare two items of the same type. Returns TRUE if the condition is true, otherwise returns FALSE
>	Notes:
<	<ul style="list-style-type: none"> <li>may be used to compare types REAL and INTEGER</li> <li>may be used to compare types CHAR and STRING</li> </ul>
>=	<ul style="list-style-type: none"> <li>case sensitive when used to compare types CHAR or STRING</li> <li>cannot be used to compare two records.</li> </ul>
<=	Examples:
<>	<ul style="list-style-type: none"> <li>"Program" = "program" evaluates to FALSE</li> <li>count = 4 evaluates to TRUE when variable count contains the value 4</li> </ul>

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](http://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.