

# Cambridge International AS & A Level

**COMPUTER SCIENCE**

**9618/21**

Paper 2 Fundamental Problem-solving and Programming Skills

**May/June 2026**

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.

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

- A string of length 1 may be considered to be either of type CHAR or STRING
- A CHAR may be assigned to, or concatenated with, a STRING
- A STRING of length greater than 1 cannot be assigned to a CHAR

|                                                                                                                                                                                                                                                                                                                                                                               |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <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 ThisString</p> <p>Example: LENGTH("Happy Days") returns 10</p>                                                                                                                                                                                                     |
| <p>TO_UPPER(x : &lt;datatype&gt;) RETURNS &lt;datatype&gt;</p> <p>&lt;datatype&gt; may be CHAR or STRING</p> <p>returns an object of type &lt;datatype&gt; formed by converting all characters of x to upper case.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• TO_UPPER("Error 803") returns "ERROR 803"</li> <li>• TO_UPPER('a') returns 'A'</li> </ul>    |
| <p>TO_LOWER(x : &lt;datatype&gt;) RETURNS &lt;datatype&gt;</p> <p>&lt;datatype&gt; may be CHAR or STRING</p> <p>returns an object of type &lt;datatype&gt; formed by converting all characters of x to lower case.</p> <p>Examples:</p> <ul style="list-style-type: none"> <li>• TO_LOWER("JIM 803") returns "jim 803"</li> <li>• TO_LOWER('W') returns 'w'</li> </ul>        |
| <p>NUM_TO_STR(x : &lt;datatype1&gt;) RETURNS &lt;datatype2&gt;</p> <p>returns a string representation of a numeric value.</p> <p>&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>If x is a negative value, the returned value will be a string beginning with the '-' character.</p> |
| <p>STR_TO_NUM(x : &lt;datatype1&gt;) RETURNS &lt;datatype2&gt;</p> <p>returns a numeric representation of a string.</p> <p>&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> <p>If the string begins with the '-' character, the returned value will be negative.</p>                   |

IS\_NUM(ThisString : <datatype>) RETURNS BOOLEAN  
 returns TRUE if ThisString represents a valid numeric value.  
 <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

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'

### 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 35.430729

### Date functions

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

DAY(ThisDate : DATE) RETURNS INTEGER  
 returns the day number from ThisDate  
 Example: DAY(04/10/2003) returns 4

MONTH(ThisDate : DATE) RETURNS INTEGER  
 returns the month number from ThisDate  
 Example: MONTH(04/10/2003) returns 10

YEAR(ThisDate : DATE) RETURNS INTEGER  
 returns the year number from ThisDate  
 Example: YEAR(04/10/2003) returns 2003

DAYINDEX(ThisDate : DATE) RETURNS INTEGER  
 returns the day index number from ThisDate where Sunday = 1, Monday = 2 etc.  
 Example: DAYINDEX(09/05/2023) returns 3

SETDATE(Day, Month, Year : INTEGER) RETURNS DATE  
 returns a value of type DATE with the value of <Day>/<Month>/<Year>  
 Example: SETDATE(26, 10, 2003) returns a date corresponding to 26/10/2003

TODAY() RETURNS DATE  
 returns a value of type DATE corresponding 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  
will generate an error if the file is not already open in READ mode.

## Operators

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

|     |                                                                                                                                                                |
|-----|----------------------------------------------------------------------------------------------------------------------------------------------------------------|
| &   | concatenates (joins) two strings.<br>Example: "Summer" & " " & "Pudding" evaluates to "Summer Pudding"<br>may also be used to concatenate a CHAR with a STRING |
| AND | performs a logical AND on two Boolean values.<br>Example: TRUE AND FALSE evaluates to FALSE                                                                    |
| OR  | performs a logical OR on two Boolean values.<br>Example: TRUE OR FALSE evaluates to TRUE                                                                       |
| NOT | performs a logical NOT on a Boolean value.<br>Example: NOT TRUE evaluates to FALSE                                                                             |
| MOD | finds the remainder when one number is divided by another.<br>Example: 10 MOD 3 evaluates to 1                                                                 |
| DIV | finds the quotient when one number is divided by another.<br>Example: 10 DIV 3 evaluates to 3                                                                  |

## Comparison operators

|    |                                                                                                                                                                                                                                                                                           |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| =  | used to compare two items of the same type.<br>evaluates to TRUE if the condition is true, otherwise evaluates to FALSE                                                                                                                                                                   |
| >  | <b>Notes:</b> <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> <li>case sensitive when used to compare types CHAR and/or STRING</li> <li>cannot be used to compare two records.</li> </ul> |
| <  |                                                                                                                                                                                                                                                                                           |
| >= |                                                                                                                                                                                                                                                                                           |
| <= |                                                                                                                                                                                                                                                                                           |
| <> |                                                                                                                                                                                                                                                                                           |
|    | <b>Examples:</b> <ul style="list-style-type: none"> <li>"Program" = "program" evaluates to FALSE</li> <li>Count = 4 evaluates to TRUE when 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 (Cambridge University Press & Assessment) 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 our 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 International Education is the name of our awarding body and a part of Cambridge University Press & Assessment, which is a department of the University of Cambridge.