

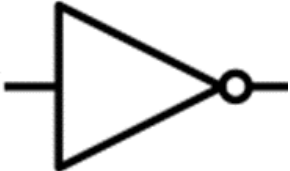


Test tables						
<p>Tests are laid out in a testing table, which indicates:</p> <ul style="list-style-type: none"> The test number A description of what the test intends to check The test data being used The type of test - normal, boundary or erroneous Expected outcome Actual outcome <p>Ideally, a programmer should run as many tests as is sensible. Many large programs, especially games, contain bugs simply because it may not be possible to test every possible input or action.</p>	Example					
	#	Description	Data	Test type	Expected	Actual
	1	Number accepted?	5	Normal	Accept	Accept
	2	Test low boundary	1	Boundary	Accept	Accept
	3	Test upper boundary	10	Boundary	Accept	Accept
	4	Test < boundary	-5	Erroneous	Reject	Reject
5	Test > boundary	20	Erroneous	Reject	Reject	

Validation

- Range check** - the input must fall within a specified range. This is usually applied to numbers and dates, but can apply to characters. For example, when making a payment to someone, the amount to be entered might be set to be greater than zero and not greater than the funds available.
- Length check** - the input must not be too long or too short. For example, a surname will require at least one letter, but is unlikely to require more than 40.
- Presence check** - a data value must be entered. For example, entering a quantity when placing an order.
- Format check** - the data must be in the correct format, such as entering a date in the format DD/MM/YYYY.
- Type check** - the data must be of a specified data type, such as an integer when specifying a quantity.

Boolean Logic																	
<p>AND</p> <p>Both inputs need to be 1s in order for the output to be a 1. Otherwise the output is 0</p>		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Output</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>0</td></tr> <tr><td>1</td><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Output	0	0	0	0	1	0	1	0	0	1	1	1
A	B	Output															
0	0	0															
0	1	0															
1	0	0															
1	1	1															
<p>OR</p> <p>At least 1 of the inputs should be 1 in order for the output to be a 1. Otherwise the output is 0</p>		<table border="1"> <thead> <tr> <th>A</th> <th>B</th> <th>Output</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td><td>0</td></tr> <tr><td>0</td><td>1</td><td>1</td></tr> <tr><td>1</td><td>0</td><td>1</td></tr> <tr><td>1</td><td>1</td><td>1</td></tr> </tbody> </table>	A	B	Output	0	0	0	0	1	1	1	0	1	1	1	1
A	B	Output															
0	0	0															
0	1	1															
1	0	1															
1	1	1															
<p>NOT</p> <p>The output is the opposite to the input. So if a 1 is input, a 0 will be output, and if a 0 is input the output will be 1.</p>		<table border="1"> <thead> <tr> <th>Input</th> <th>Output</th> </tr> </thead> <tbody> <tr><td>0</td><td>0</td></tr> <tr><td>1</td><td>1</td></tr> </tbody> </table>	Input	Output	0	0	1	1									
Input	Output																
0	0																
1	1																

Input sanitation	The process of checking entered data and removing dangerous inputs which could otherwise be used to cause damage to a program
Authorisation	Verifying the identify of the user
Validation	Checking input data is sensible and in the right format
Maintainability	The process of ensuring that a program is easy to understand, modify and update
Naming conventions	Choose either CamelCase or under_score conventions for variables and stick to one.
Comments	Annotations using a # in Python. Should be used to explain what a section of code intends to do
Iterative Testing	Tests carried out while a program is being developed. Step by step.
Final/ Terminal testing	A test carried out when all parts of a program are complete
Syntax Error	Error in a program resulting from code not following syntax rules governing how to write statements in a programming language.
Logic Error	Error in a program which does not cause a program to crash but causes unexpected results.
Test Data	Data entered into a program to test if it is working
Normal Data	Data entered into a program that should produce a positive result.
Boundary Data	Data entered into a program at the edge of its acceptable range
Invalid Data	Data entered into a program that should produce a negative result
Erroneous Data	Data that a program cannot process and should not accept.