**not**

*Operator of integer-valued or logical complement.*

**Syntax:**

*c* **= not** *expr;*

**Arguments:**

*expr* – expression for logical complement or a number for integer-valued complement.

**Description:**

*not* – operator of integer-valued or logical complement. In case of operation of logical complement expression *expr* can comprise any mathematical or logical operations. The result of mathematical or logical operation will be the logical variable featuring value “YES” or “NO”. After operation of logical complement a variable featuring value “YES” will replace it with “NO”, and vice versa. A resulting variable shall feature *boolean* type.

The application of complement operation to the data of integer-valued types brings about a bitwise inversion (complement) corresponding to this number of binary code.

**Result:**

*c* – operation result.

**Example:**

|  |  |
| --- | --- |
|  | **var**  c2:**boolean**,  c3:**boolean**  **end**  x = 78;  c1 = **not** x;  c2 = **not** (x< 10);  c3 = **not** (x > 10); |

As a result, variable *c1* will be assigned value -79. Number 78 occupies 2 bytes in memory and in the binary shape is of the form of: 0000000001001110. In case of fulfilling operation all 0 get replaced with 1, while 1 is replaced with 0, accordingly. After fulfilling operation not 78 we get: 1111111110110001, which corresponds to number -79 in a decimal form.

Variable *c2* will be assigned value “NO”, variable *c3* will be assigned value “YES”.