

Scalar Data Types

Booleans

The Boolean Data Type is a data type, having two values (usually denoted true or false), intended to represent the truth values of logic and Boolean Algebra.

Specification :- In Pascal and Ada, the Boolean datatype is considered simply a language-defined enumeration, viz;

`type Boolean = (false, true);`

which both defines the names true and false for the values of the types and defines ordering `false < true`.

Common operations are

`and` : `Boolean x Boolean` \rightarrow `Boolean` (Conjunction)

`or` : `Boolean x Boolean` \rightarrow `Boolean` (inclusive disjunction)

`not` : `Boolean` \rightarrow `Boolean`. (negative or Complement)

Implementation :-

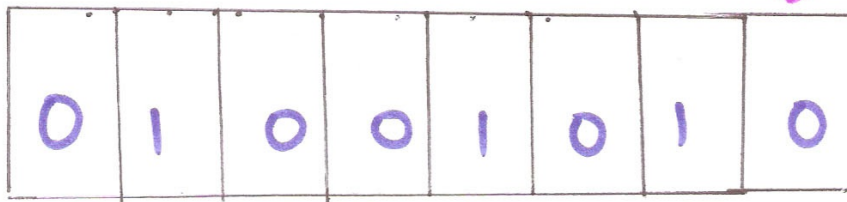
Single bit of storage is provided, no descriptor designated the data type is needed. Because single bits may not be separately addressable in memory which often takes a byte or word to represent it if extended.

Then the values true and false might be represented

In two ways within this storage unit :

- ① → A particular bit is used for the value (often the sign bit of the number representation), with '0 = false', '1 = true', and the rest of the byte or word ignored, or
- ② → A zero value in the entire storage unit represents false, and any other non zero value represents true.

1)

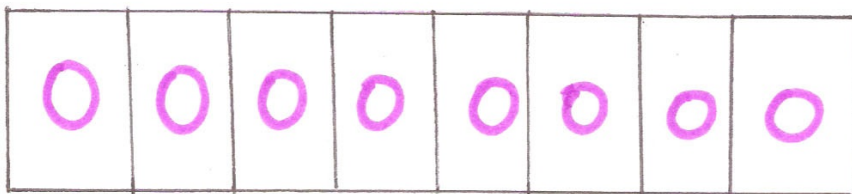


if this is particular bit which is to be considered

0 = false
1 = true

Ignored

2)



0 = false

1 = true