Numerical Data Types

Integers: The most primitive numerical data type is integers.

Specification: -
C has four different integer specifications:
int, short, long, and char

But maximal and minimal values depend upon the what
bit architecture of hardware is basically, and in some
languages these values represented as defined constants.
Like in Pascal: max int

Types of Operations on Integers

1. Arithmetic Operation: -
   It is of basically two types
   - Binary operation
     Binop: integer x integer \rightarrow integer
     Eg: addition (+), subtraction (-),
         multiplication (*), division (/),
         remainder (mod)
   - Unary operation
     UnaryOp: integer \rightarrow integer
     Negative (-), or
     Identify (+), or
     ABS value
2) Relational Operations:-
Signature is
Relop : integer x integer $\rightarrow$ Boolean

Where Relop may be equal, notequal, less than,
greater than, less-than-or-equal, greater-than-or-equal

Relational operation compares the values of its two
arguments data values and returns a Boolean (true or false
value) data object as its result.

3) Assignment Operations:-
Signature
assignment : integer x integer $\rightarrow$ integer

and
assignment : integer x integer $\rightarrow$ integer

4) Bit operations:-
In C, integers also play the role of Boolean values.
Therefore additional bit operation are also defined.

Signature:-
Binop : integer x integer $\rightarrow$ integer

Operator ($\&$) for and the bits together
Operator ($|$) for or the bits together
Operator ($<<<<$) for shift the bit among others.
Implementation of Integers

Most often using the hardware-defined integer storage representation and a set of hardware arithmetic and relational operations on integers.

**NO DESCRIPTOR**

**DESCRIPTOR STORED IN SEPARATE WORD**

**DESCRIPTOR STORED IN SAME WORD**

**THREE STORAGE REPRESENTATIONS FOR INTEGERS**
Numeric Data Types

Sub Ranges of an Integer

Specification :-
A Subrange of an integer data type is a subtype of the integer data type and consists of a sequence of integer values within some restricted range.

Eg
Integers in Range

\[ 1 \text{ to } 10 \]

\[ -500 \text{ to } 1000 \]

Declaration in Pascal
A: 1..10

Declaration in ADA
A : integer range 1..10

Implementation :- Its implementation basically has two advantages

(i) Smaller Storage Requirement :- As a smaller range of values,
a subrange value can usually be stored in fewer bits than a general integer value.
2) Better type checking:

More precise type checking to be performed on the value assigned to that variables

\[ \text{Eg} \quad \text{if variable Month is: } \text{Month} : 1..12 \]

then the assignment

\[ \text{Month} := 0 \]

is invalid and can be detected at compile time.

If we use assignment

\[ \text{Month} : \text{Month} + 1 \]

at runtime, Compiler checks for range limit that should not be exceeded.