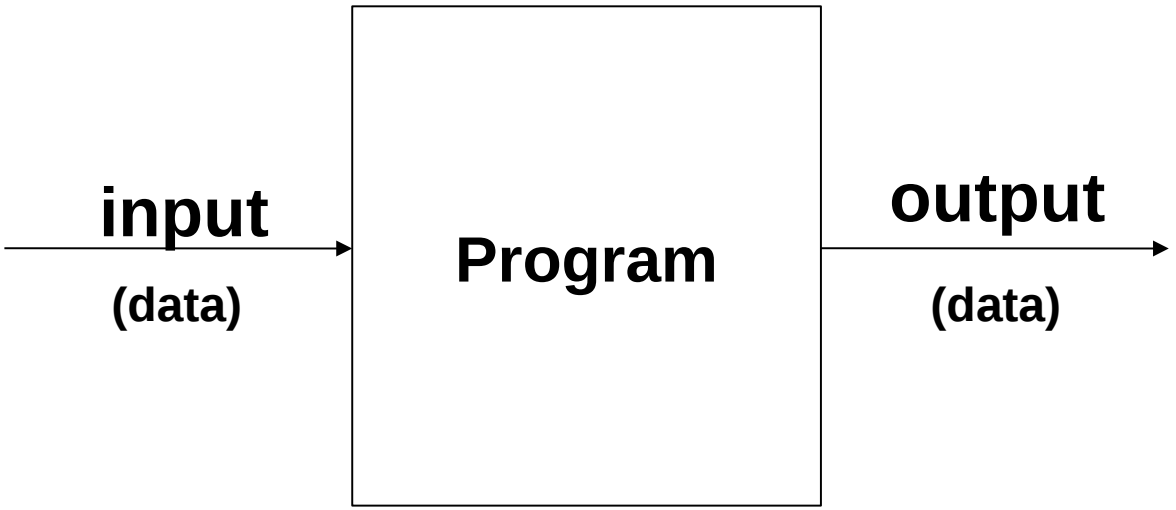
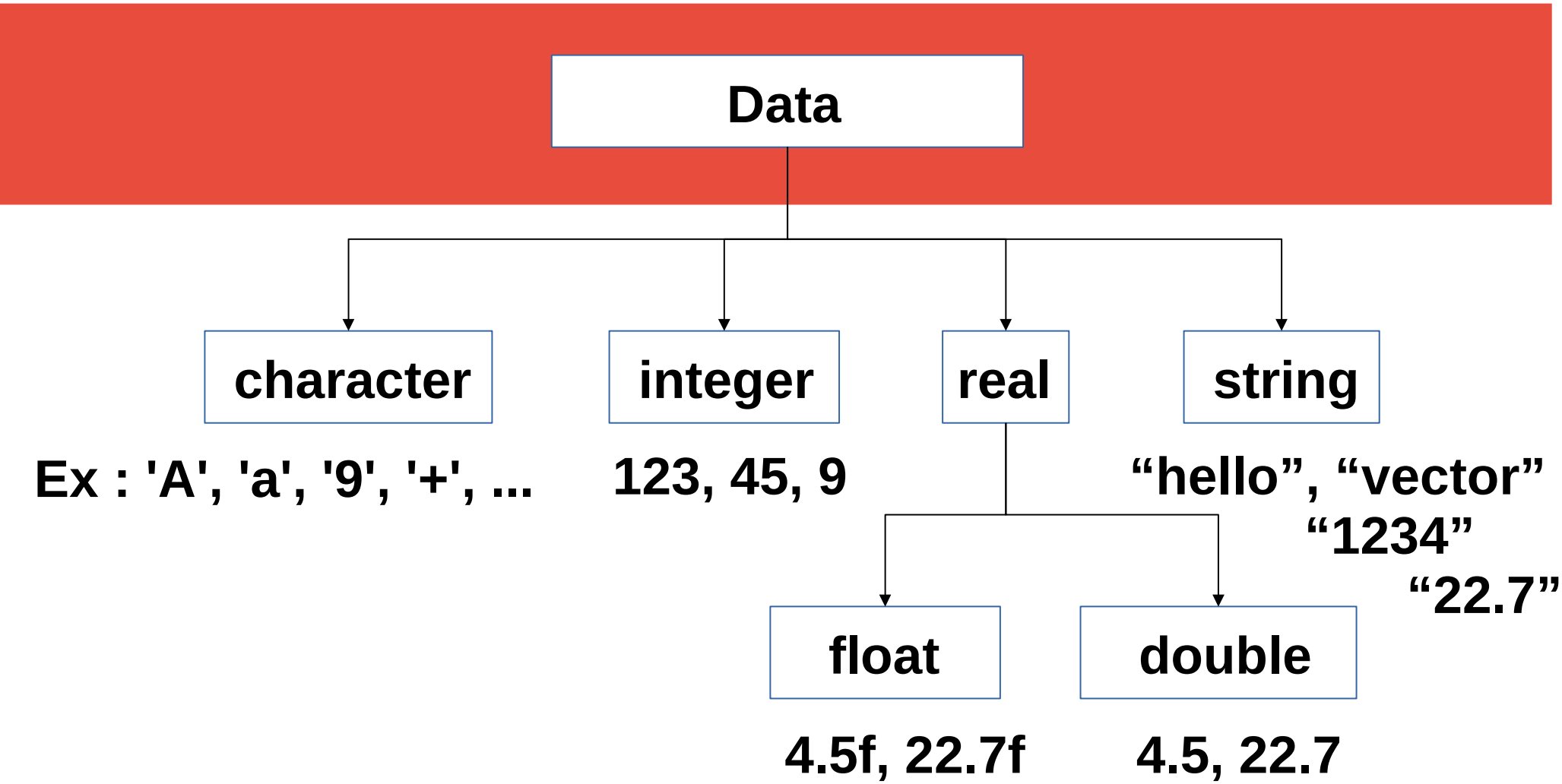
Datatypes


input
(data)
Program
output
(data)

Data types are declarations for variables. This determines the type and size of data associated with variables.

A variable is a name assigned to a memory space that may be used to store a data value.

Syntax : Datatype variable = data.


Data
character
integer
real
string
Ex : 'A', 'a', '9', '+', ...
123, 45, 9
“hello”, “vector”
“1234”
“22.7”
float
double
4.5f, 22.7f
4.5, 22.7

Data types

Pre-defined

- char (1byte) (%c)
- int (4byte) (%d)
- float (4byte) (%f)
- double (8byte) (%lf)

User defined

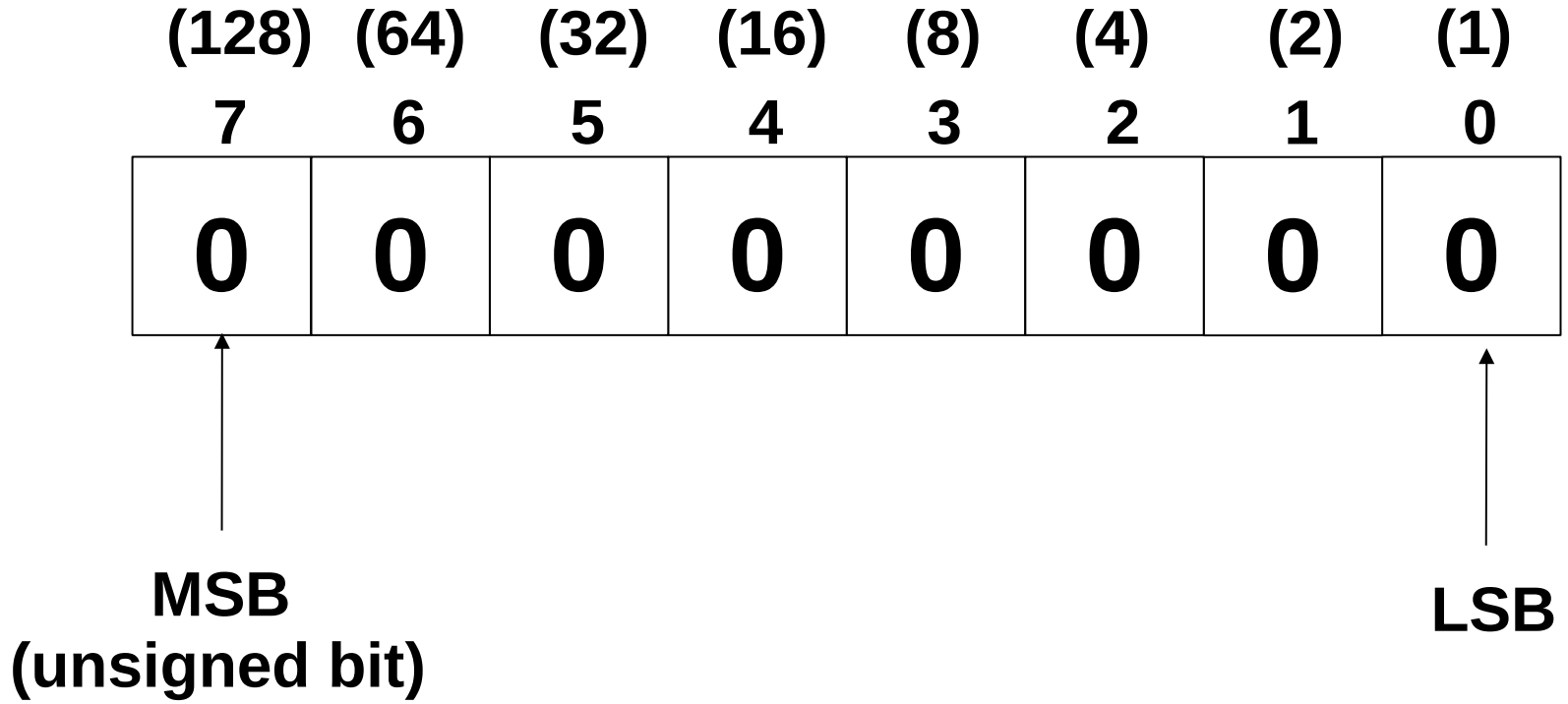
- struct
- union
- enum
- typedef

character

Keyword	**:**	**char**
Size	**:**	**1byte (8 bits)**
Qualifiers	**:**	**signed, unsigned**
Format specifier	**:**	**%c**
Range	**:**	**-128 to 127 (signed char)**
		0 to 255 (unsigned char)

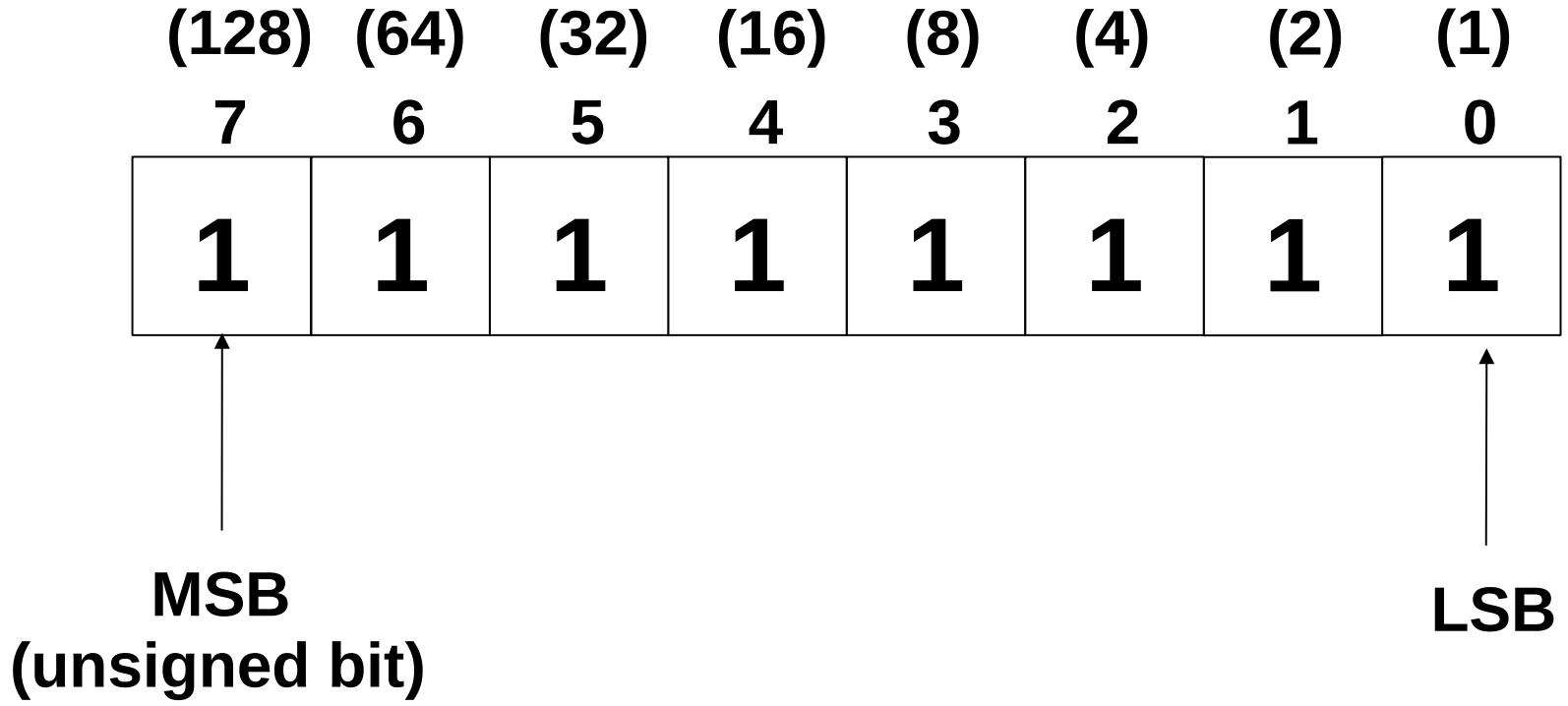
Unsigned Character

Min value ---> 0



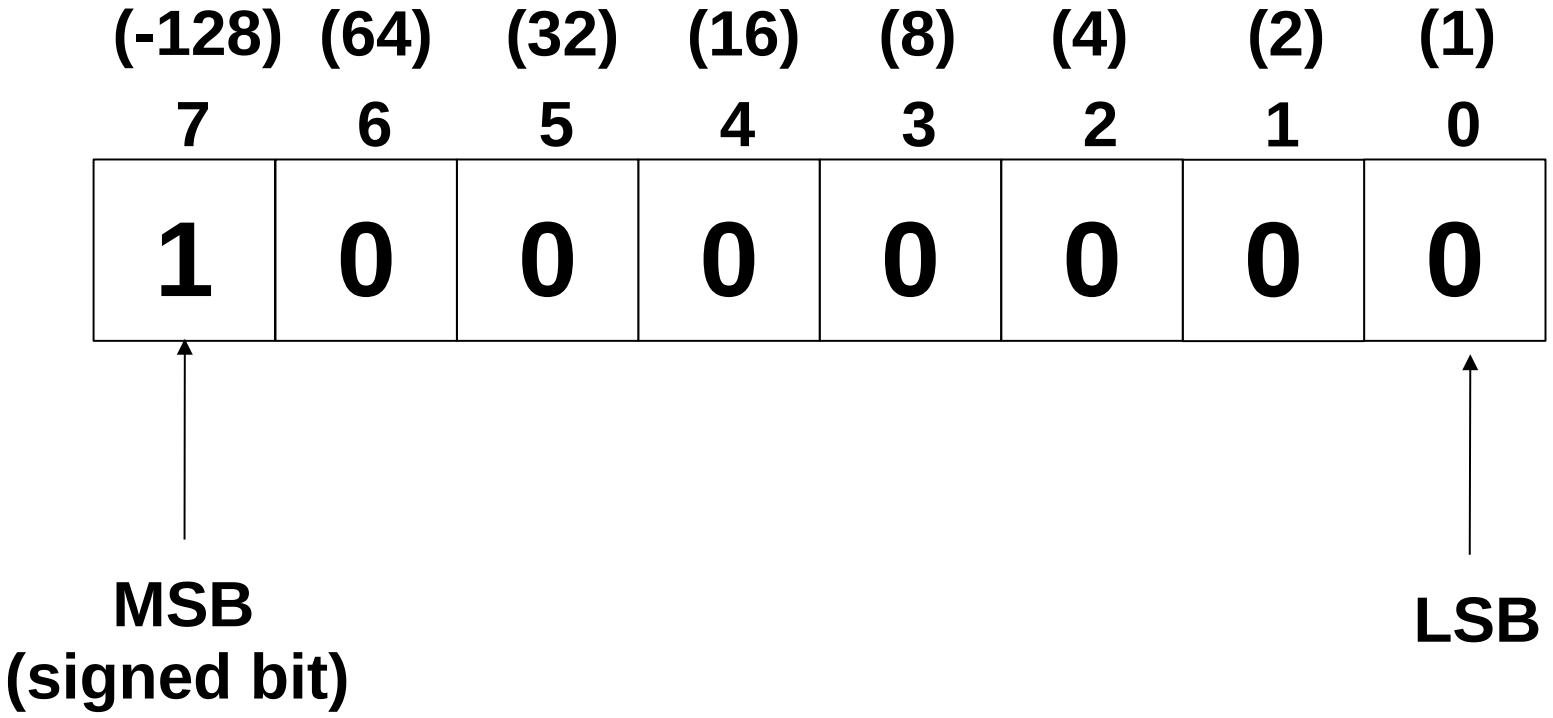
Unsigned Character

Max value ---> 255



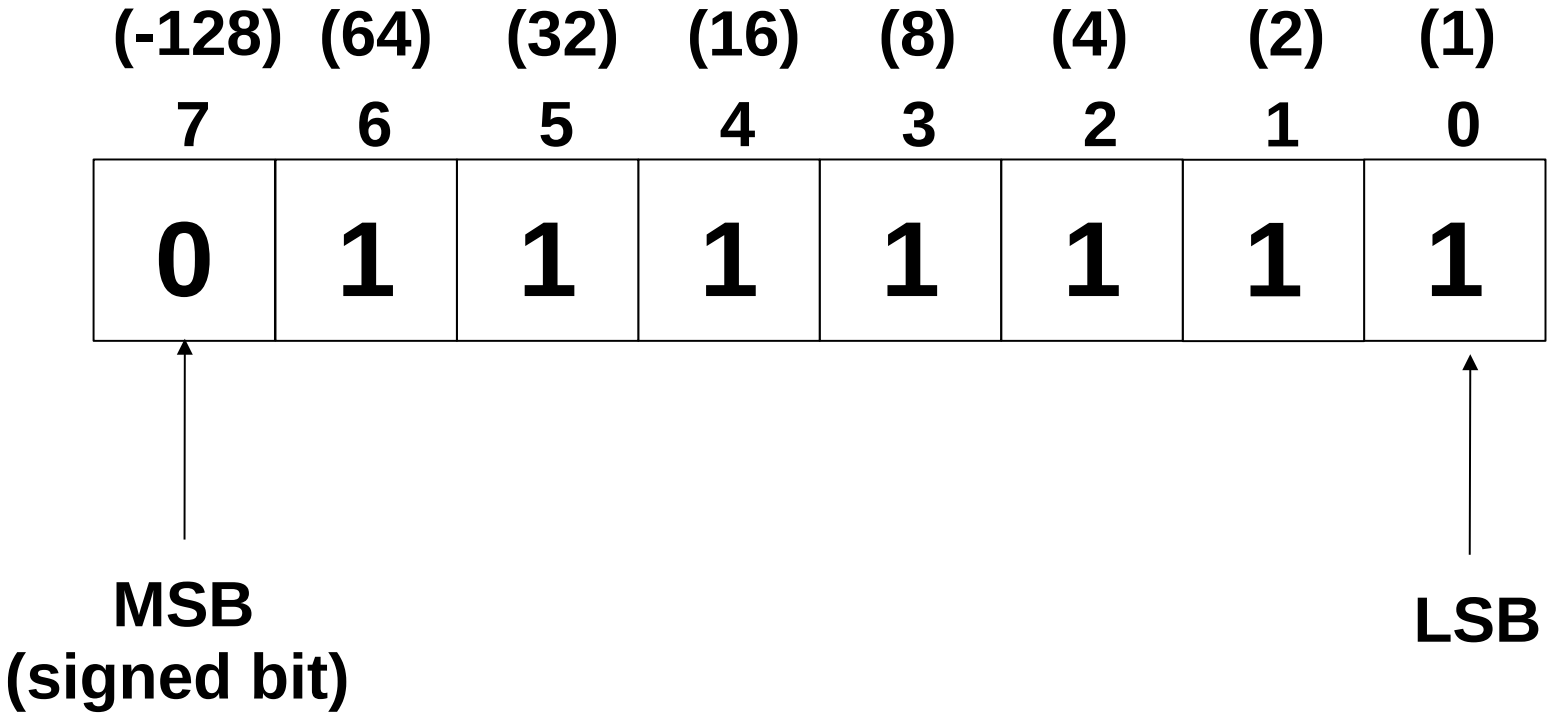
signed Character

Min value ---> -128

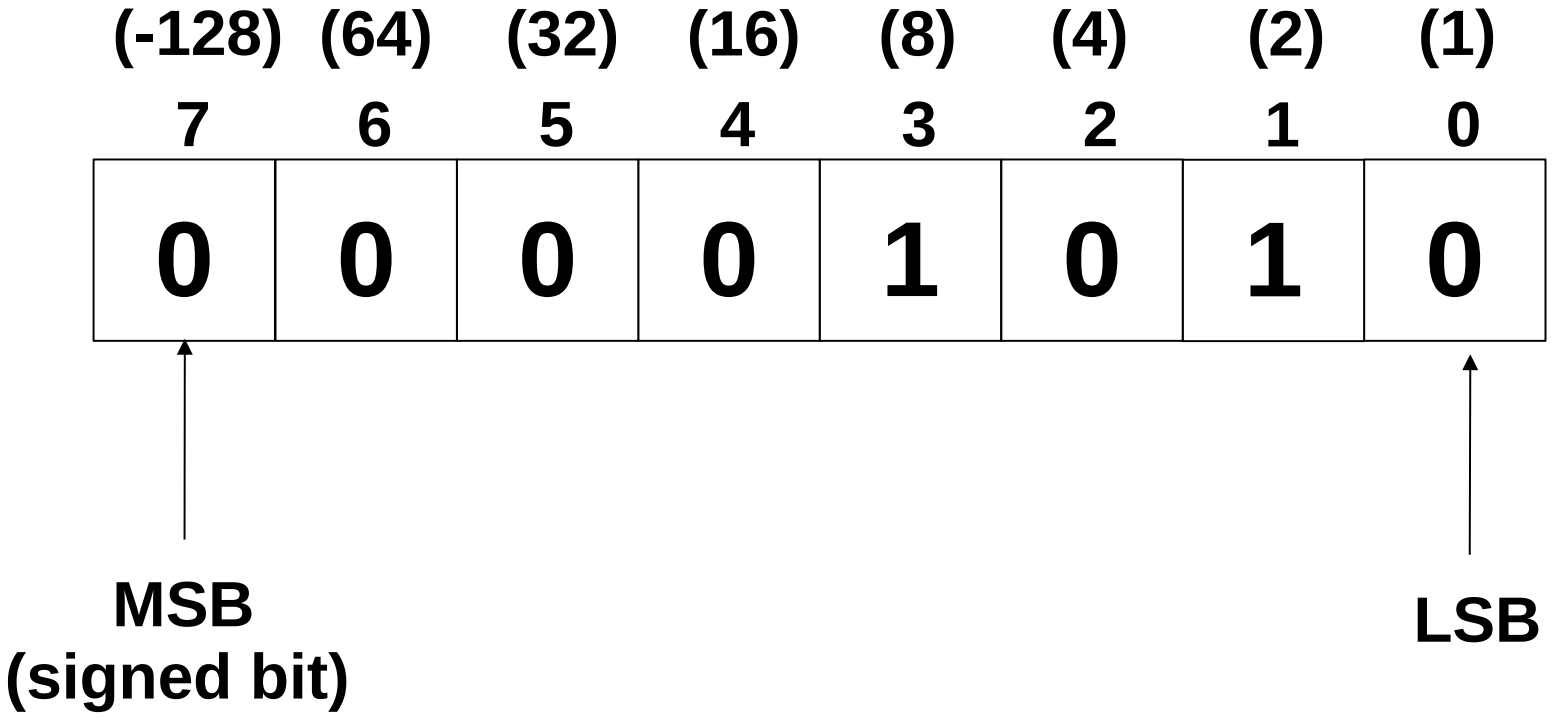


signed Character

Max value ---> 127

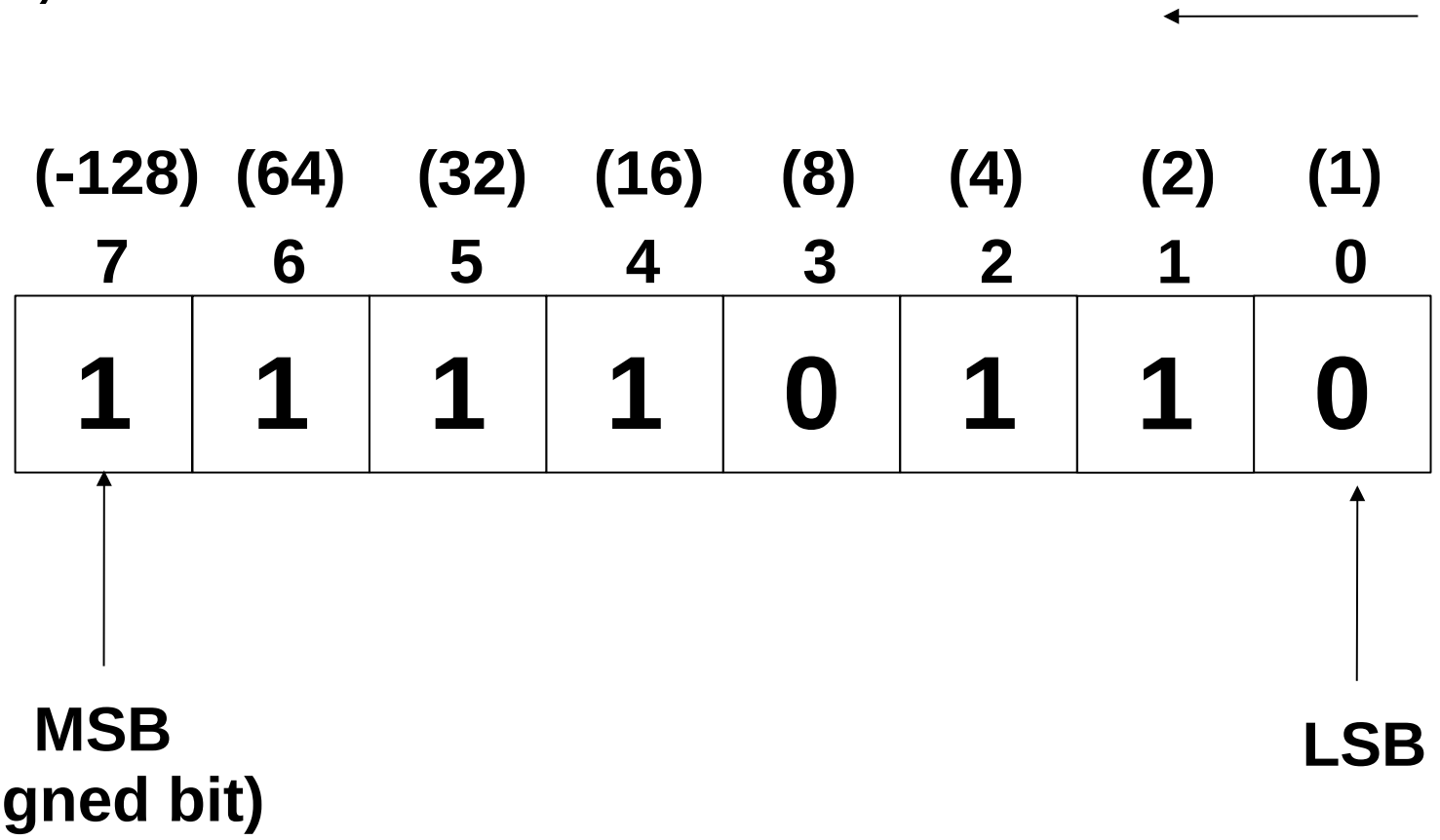


10 binary



-10 binary

2's(x) = -x;



-128
64
32
16
4
2

-10

Character set

-> Alphabets

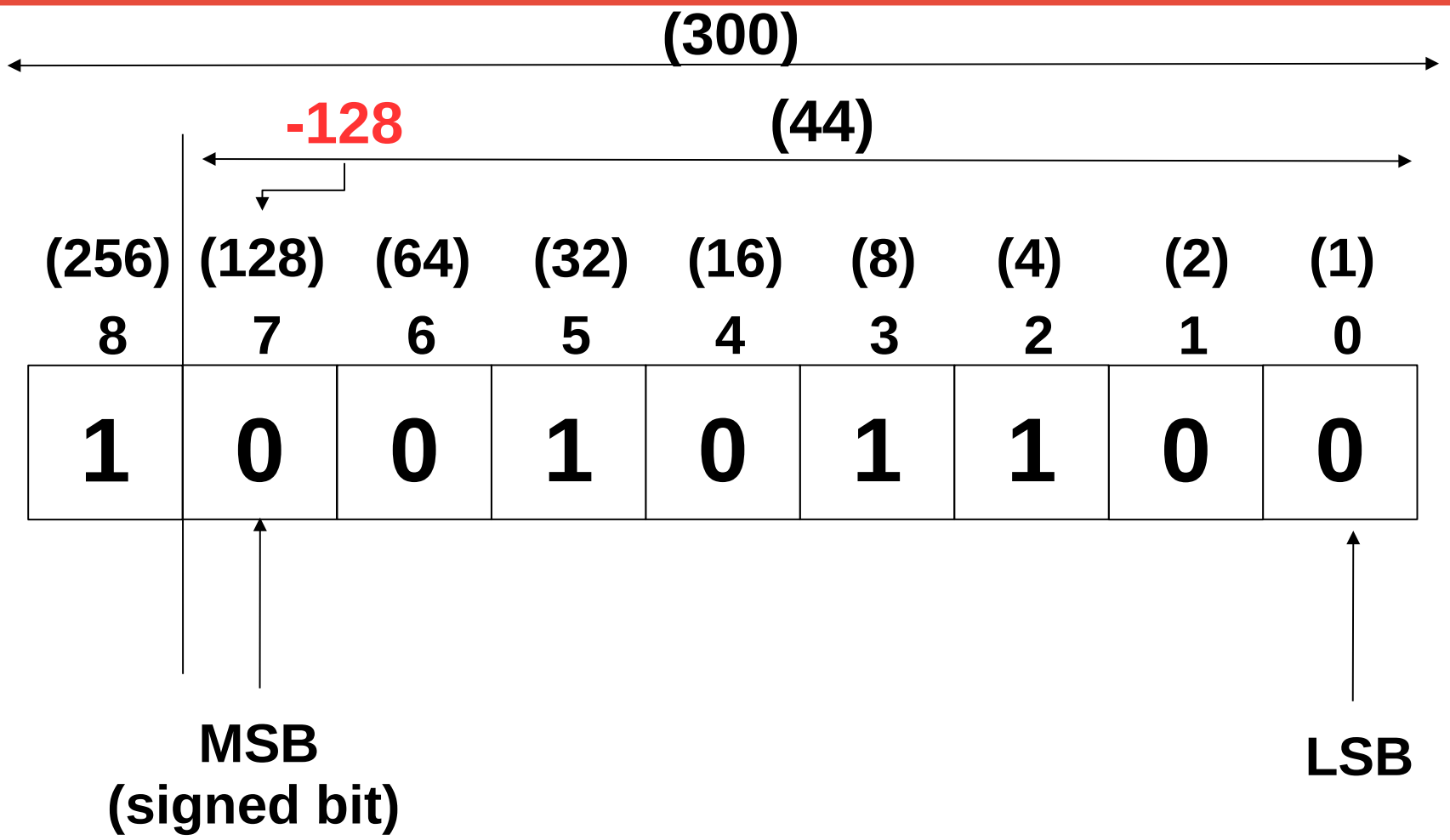
. Uppercase ('A' to 'Z') ---> 65 to 90

. Lowercase ('a' to 'z') ---> 97 to 122

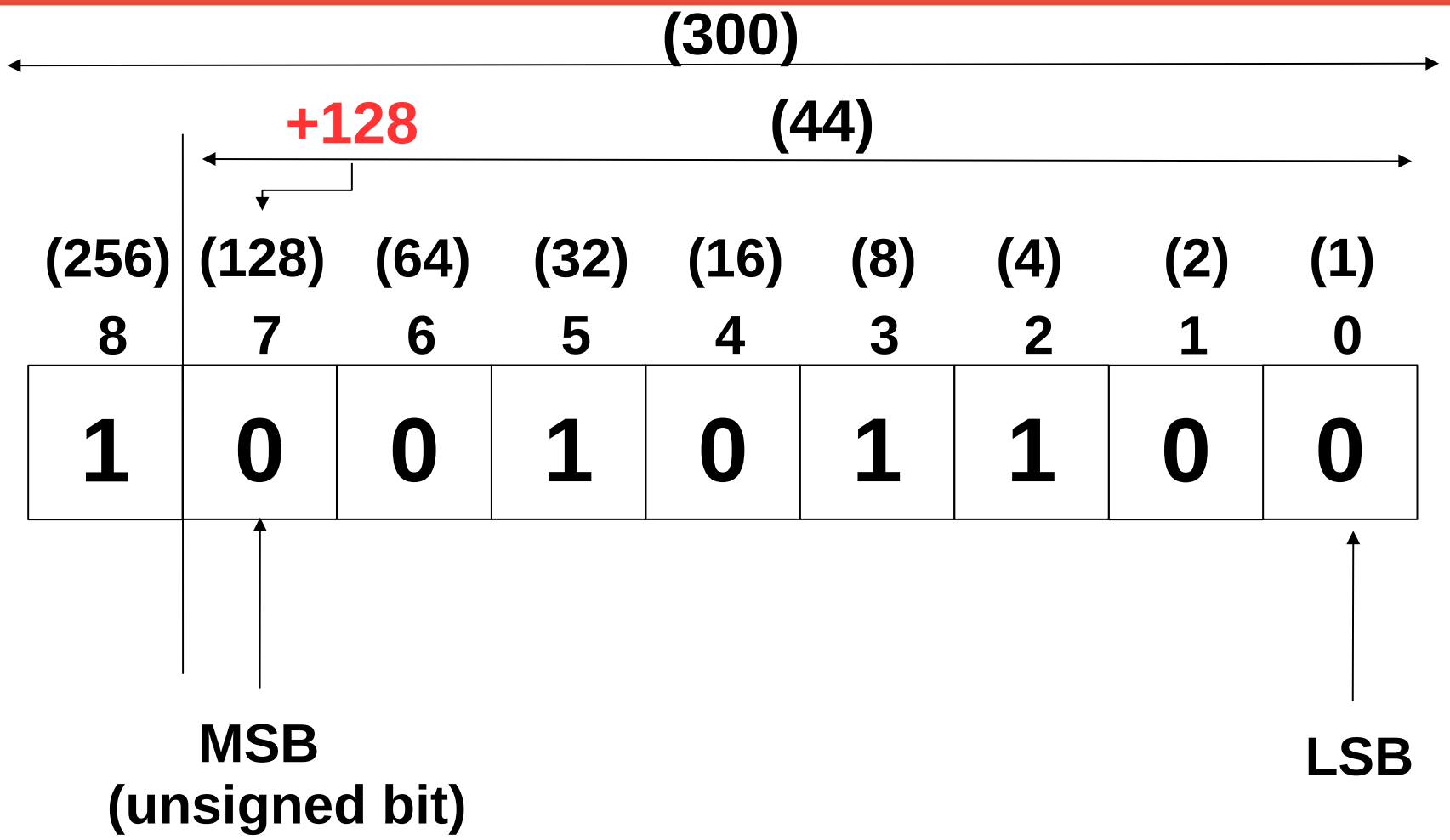
-> Digits ('0' – '9') ---> 48 to 57

-> Special characters ---> ';', '$', '#', ...

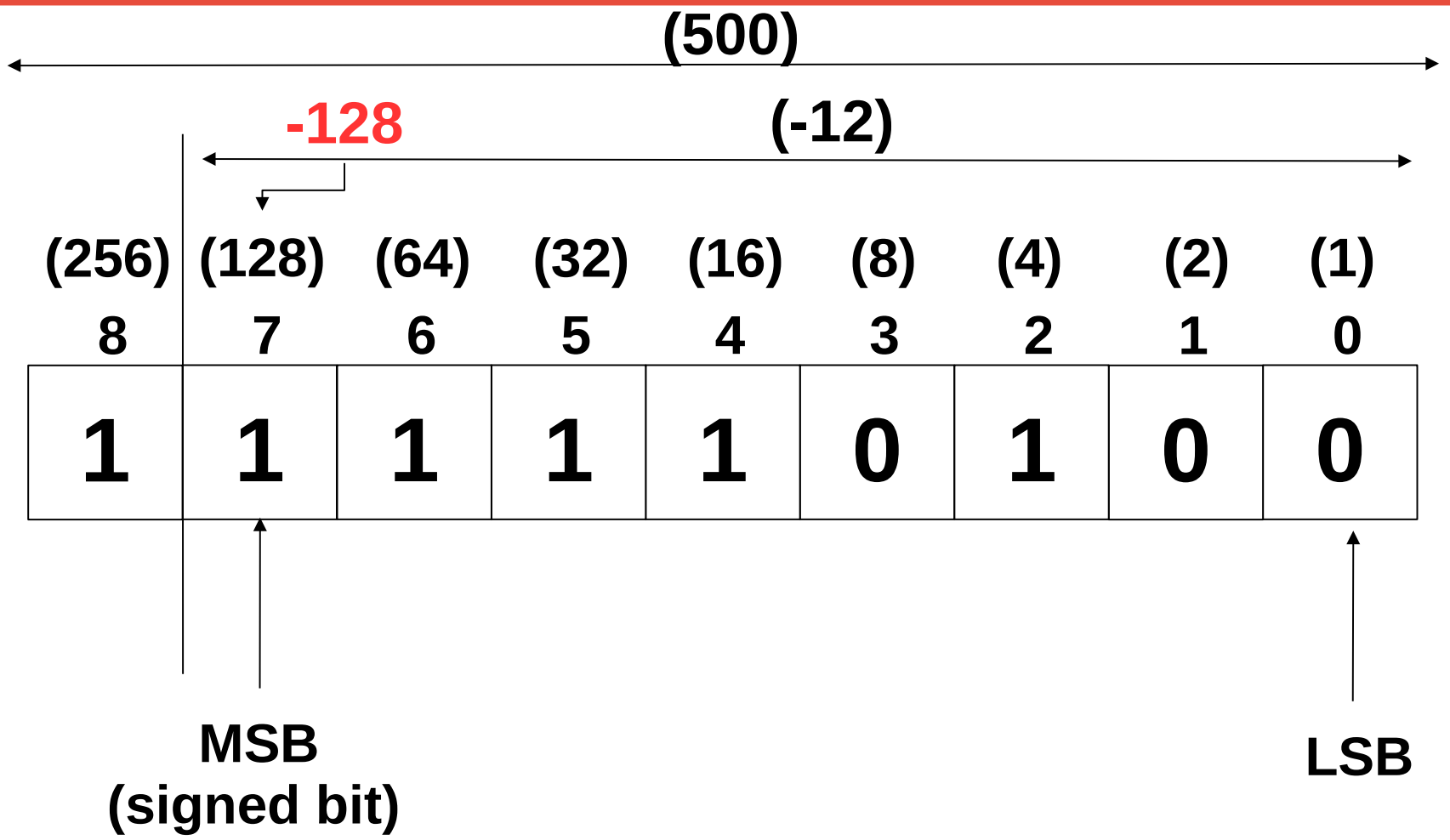
300 binary


(300)
-128
(44)
(256) (128) (64) (32) (16) (8) (4) (2) (1)
8 7 6 5 4 3 2 1 0
1 0 0 1 0 1 1 0 0
MSB
(signed bit)
LSB

300 binary


(300)
+128
(44)
(256) (128) (64) (32) (16) (8) (4) (2) (1)
8 7 6 5 4 3 2 1 0
1 0 0 1 0 1 1 0 0
MSB
(unsigned bit)
LSB

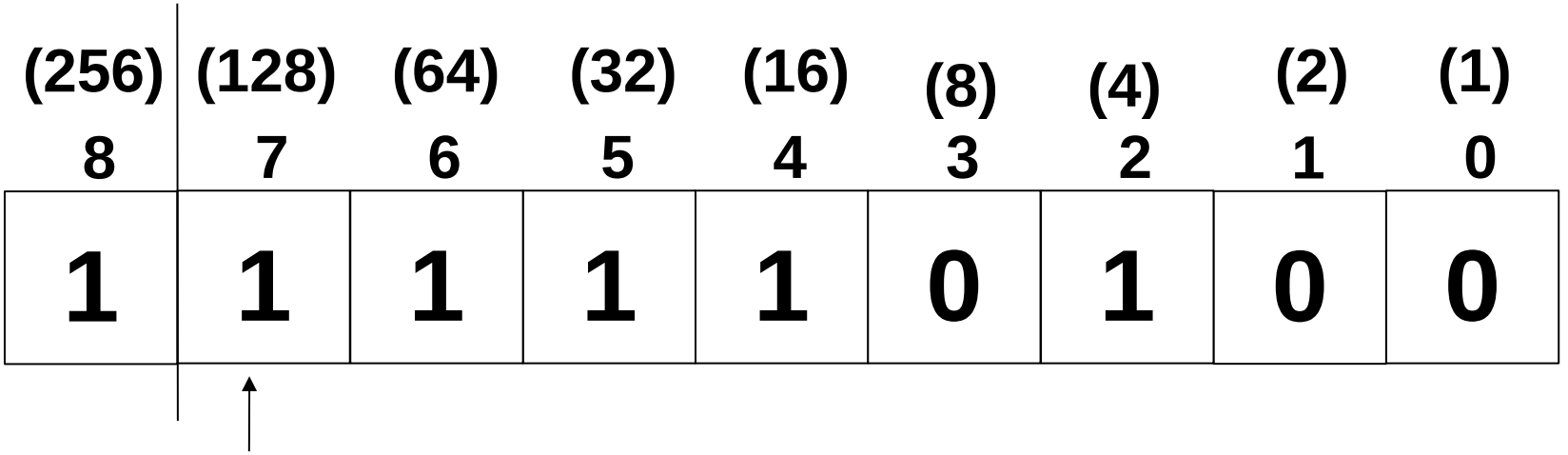
500 binary


(500)
-128
(-12)
(256) (128) (64) (32) (16) (8) (4) (2) (1)
8 7 6 5 4 3 2 1 0
1 1 1 1 1 0 1 0 0
MSB
(signed bit)
LSB

Note

If MSB bit is signed bit, and set with 1, then find 2's complement value and assign '-ve' representation to know what is the value it is.

Ex : 500 value in 1 byte



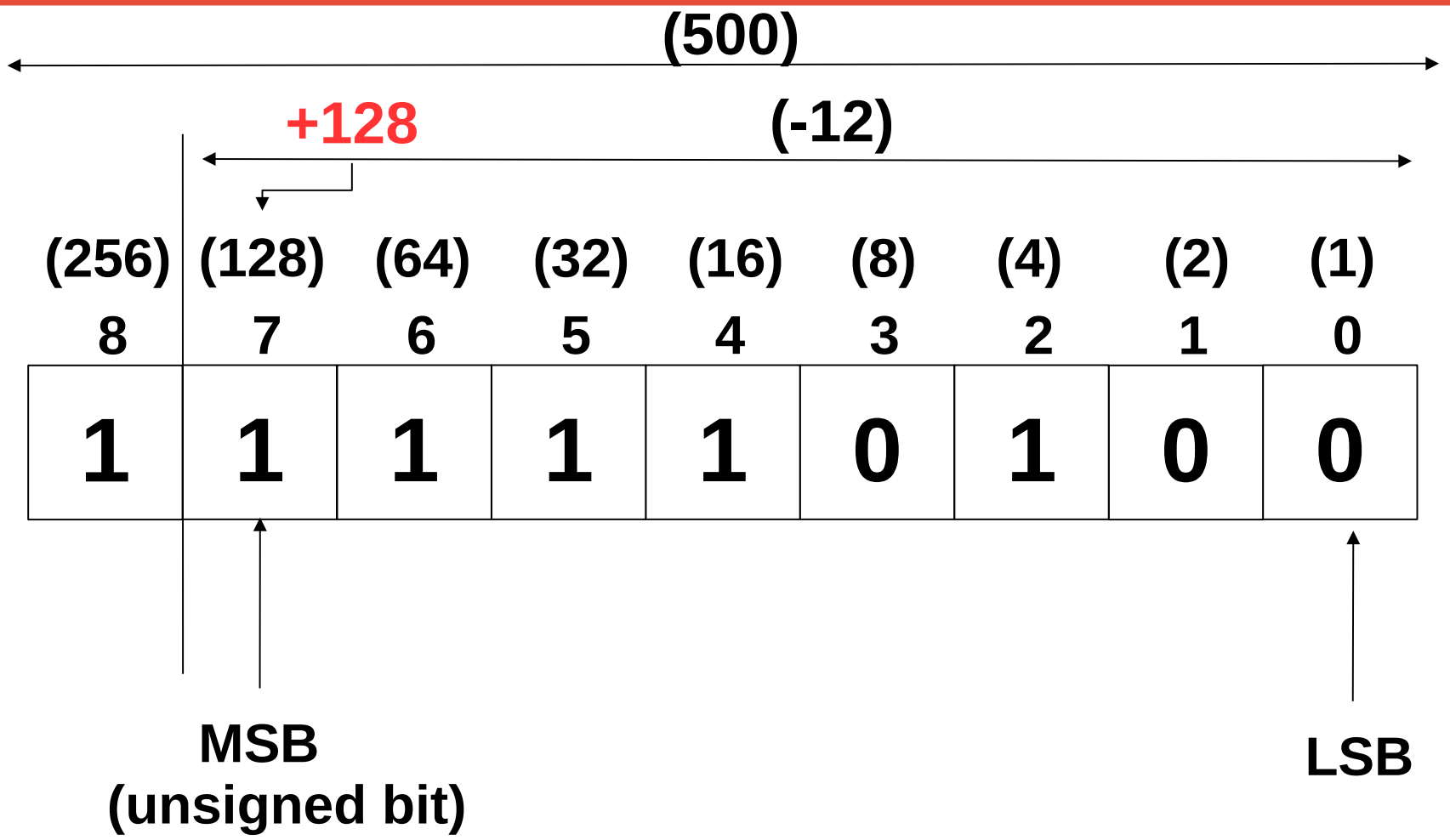
(8)(4)

2's complement for 1st byte is ---> 0 0 0 0 1 1 0 0

(12)

After assign -ve representation --> -12

500 binary


(500)
+128
(-12)
(256) (128) (64) (32) (16) (8) (4) (2) (1)
8 7 6 5 4 3 2 1 0
1 1 1 1 1 0 1 0 0
MSB
(unsigned bit)
LSB

Keyword	**:**	**int or long int**
Size	**:**	**4 bytes (32 bits) for GCC**
Qualifiers	**:**	**signed, unsigned, short, long**
Format specifier	**:**	**%d, %u**
Range	**:**	**-2,147,483,648 to 2,147,483,647 or -2G to 2G (signed int)** **0 to 4,294,967,295 or 0 to 4G (unsigned int)**

int datatype

signed int --> 4bytes --> %d --> -2G to +2G

unsigned int -> 4bytes--> %u --> 0 to 4G

short signed int --> 2bytes --> %hd --> -32768 to +32767

short unsigned int --> 2bytes --> %hu --> 0 to 65535

long signed int --> 4bytes --> %ld --> -2G to +2G

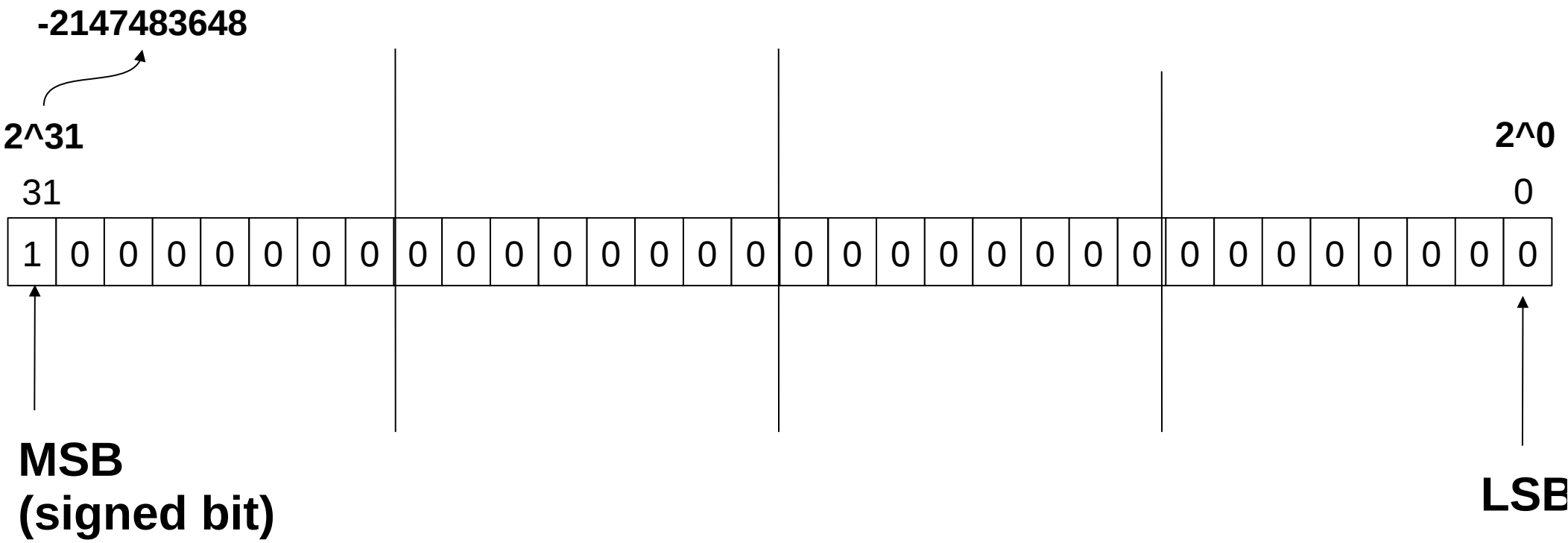
long unsigned int --> 4bytes --> %lu --> 0 to 4G

long long signed int --> 8bytes --> %lld

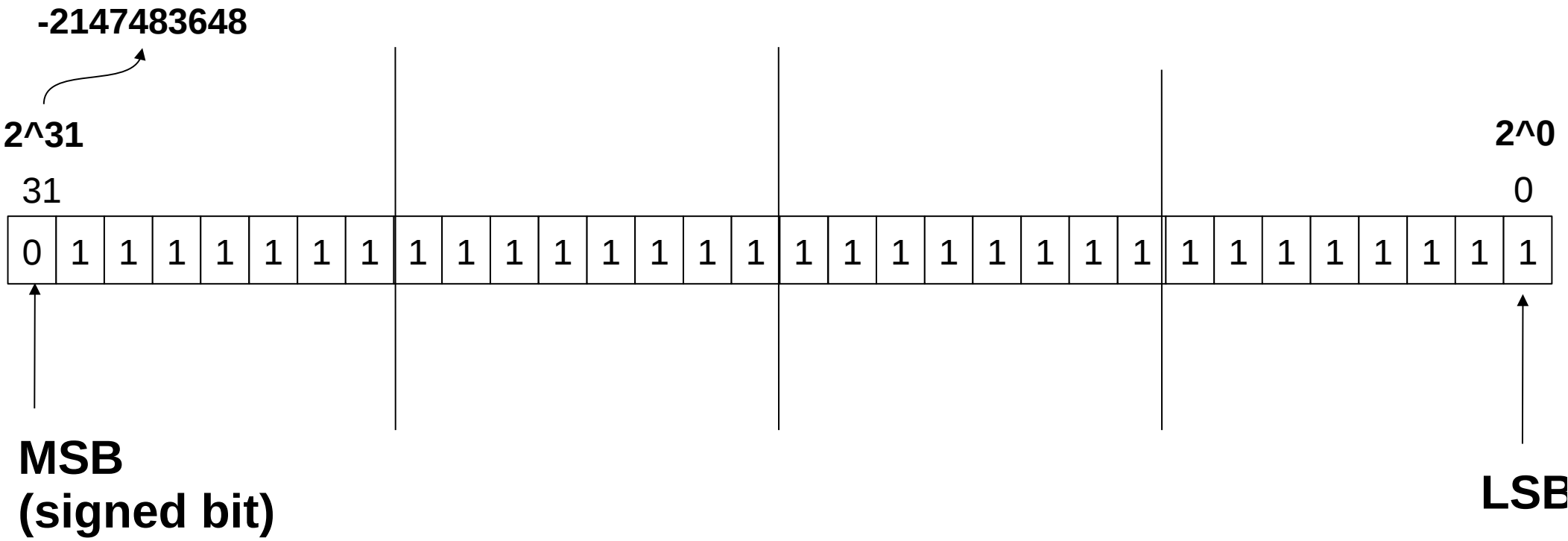
long long unsinged int --> 8bytes --> %llu

-2G value --> -2147483648   +2G value --> 2147483648

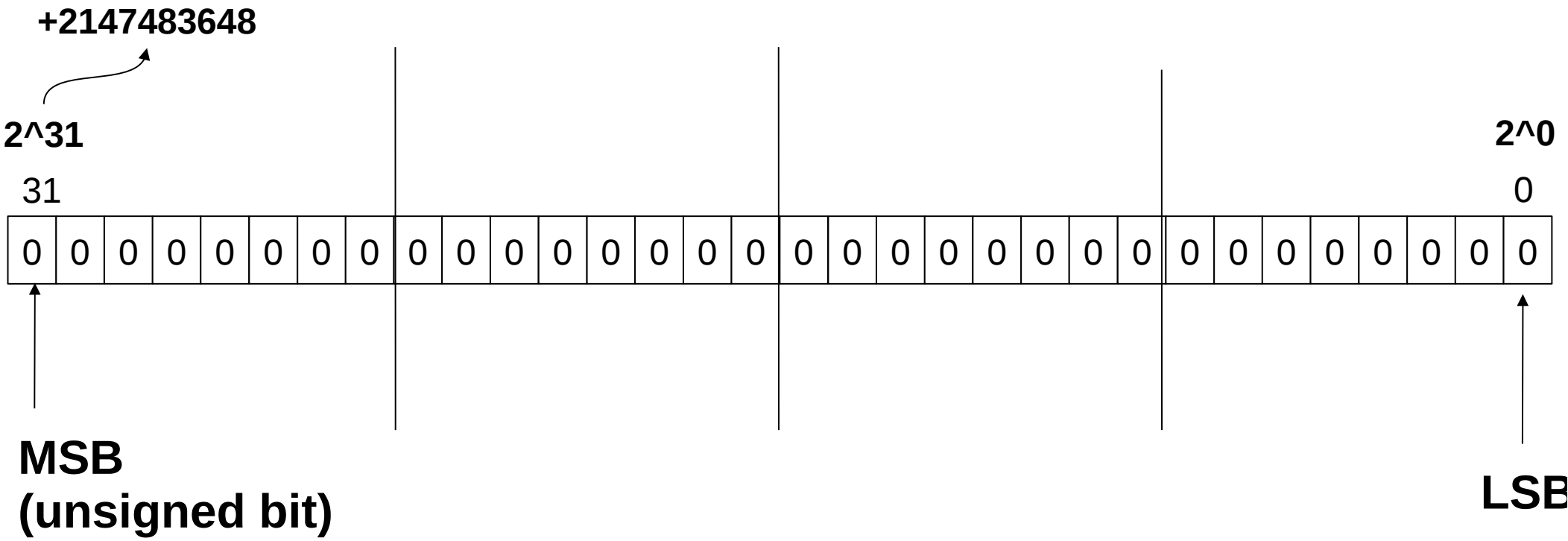
Singed int min value --> -2147483648



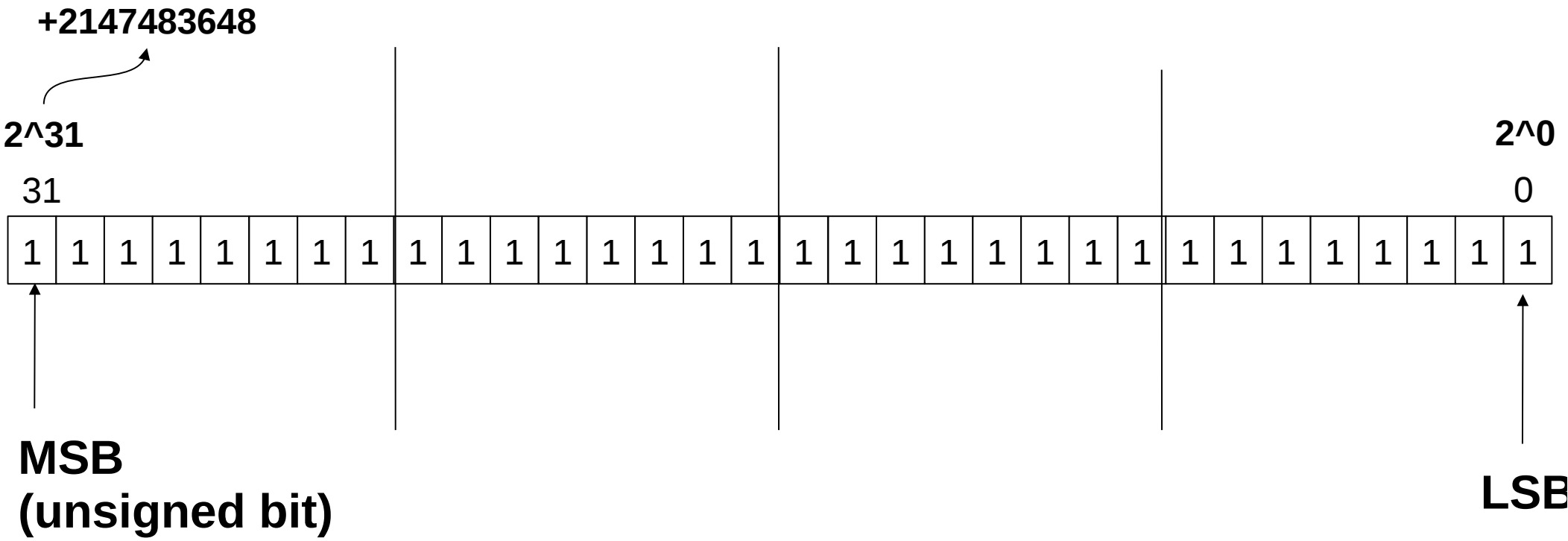
Singed int max value --> +2147483647



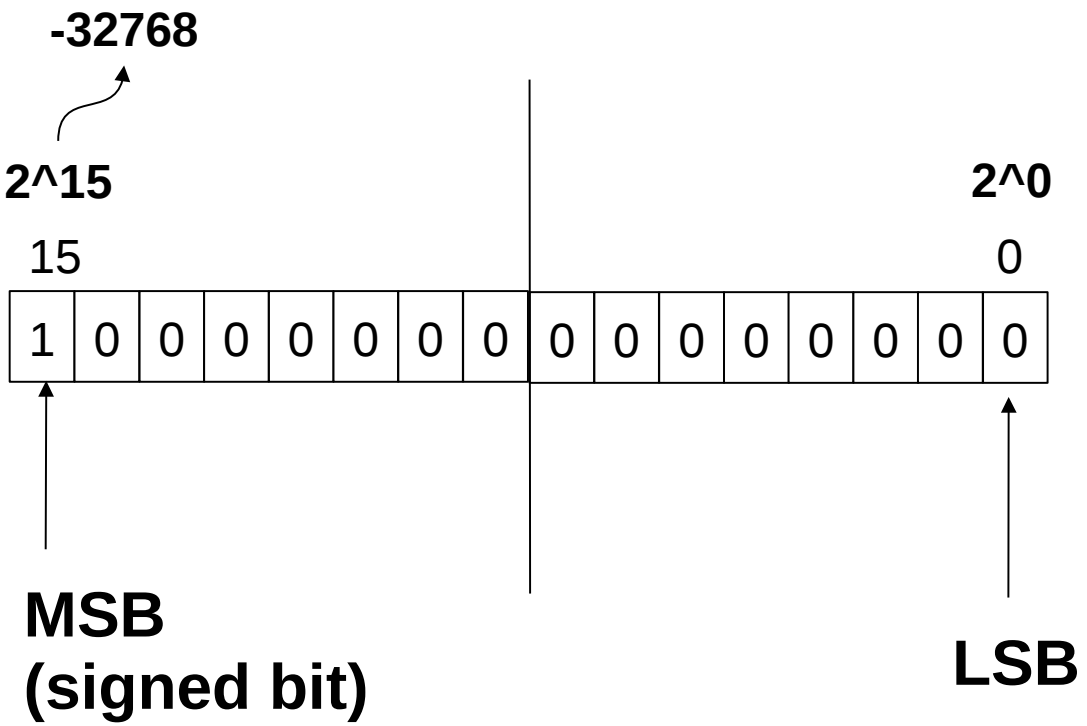
unsinged int min value --> 0



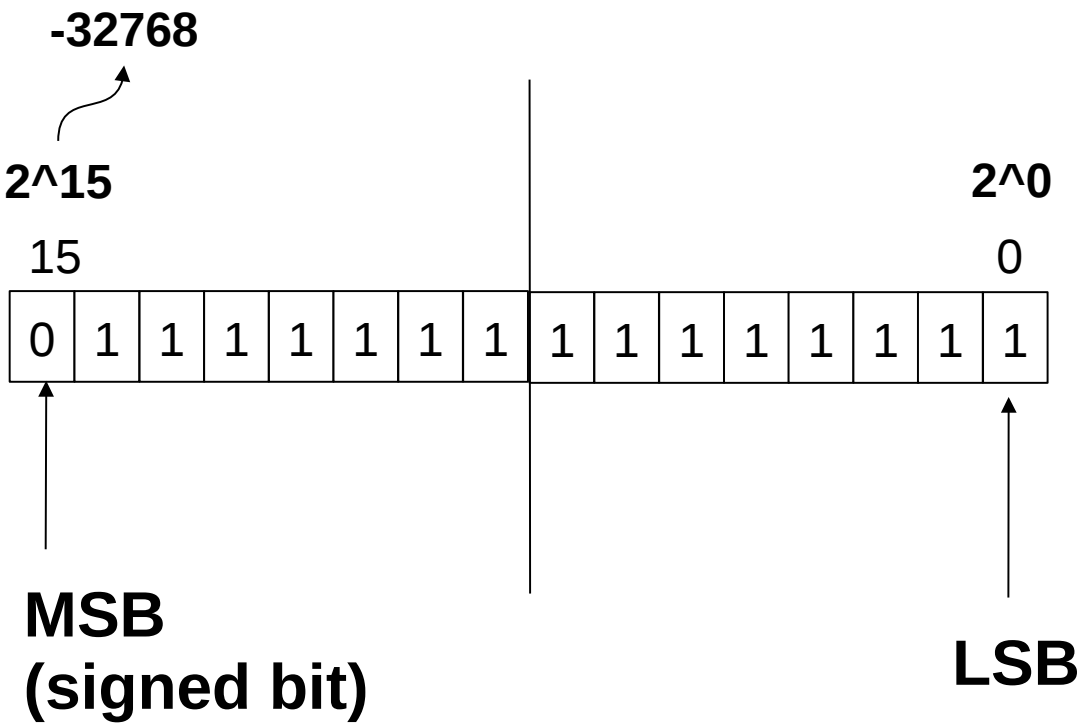
unsinged int max value --> 4294967295



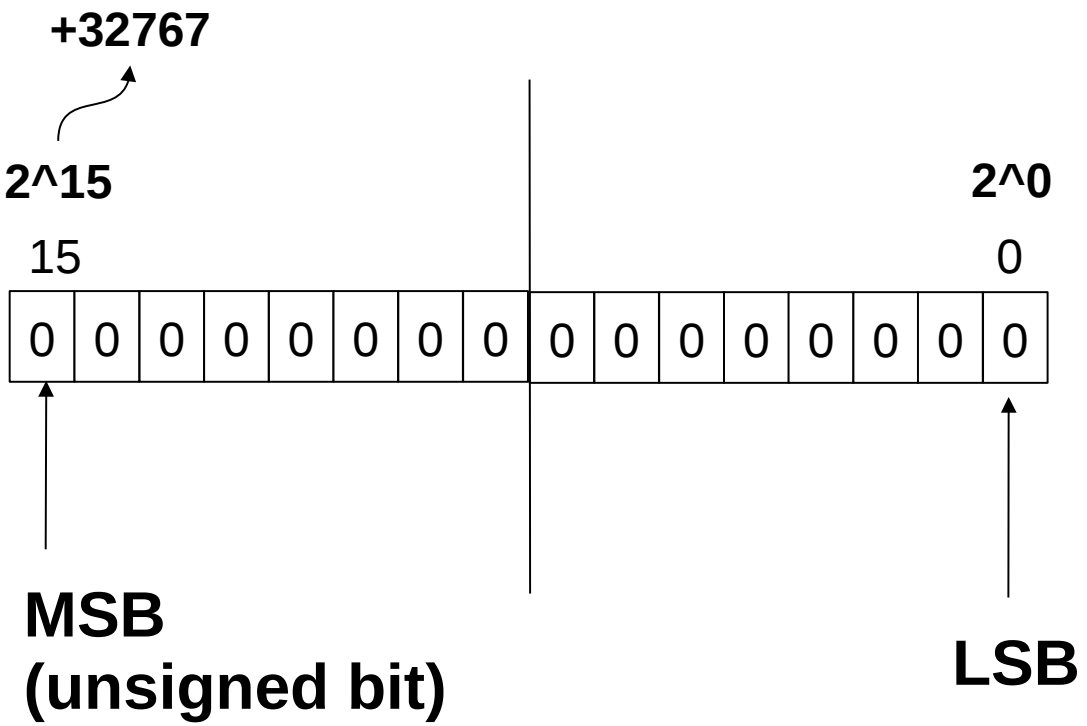
Singed short int min value --> -32768



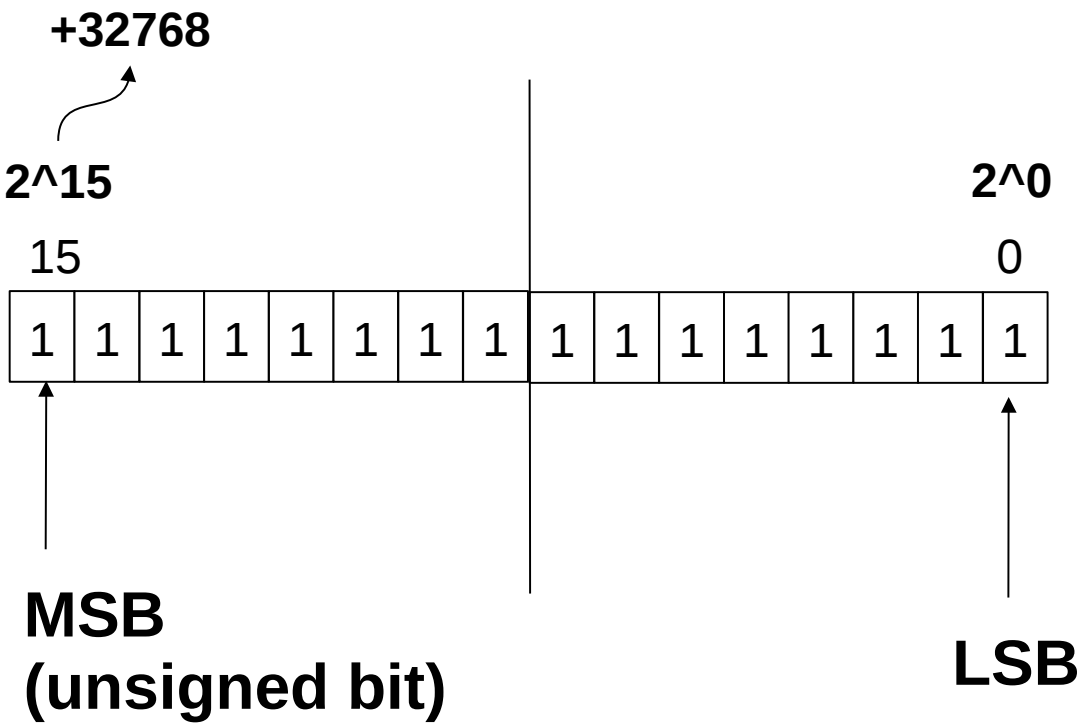
Singed short int max value --> +32767



unsinged short int min value --> 0



unsinged short int max value -->65535



float

Keyword : float

Size : 4 bytes (32 bits)

Format specifier : %f, %g, %e

double

Keyword : double

Size : 8 bytes (64 bits)

Format specifier : %lf

Keyword : long double

Size : 12 bytes (96 bits)

Format specifier : %Lf