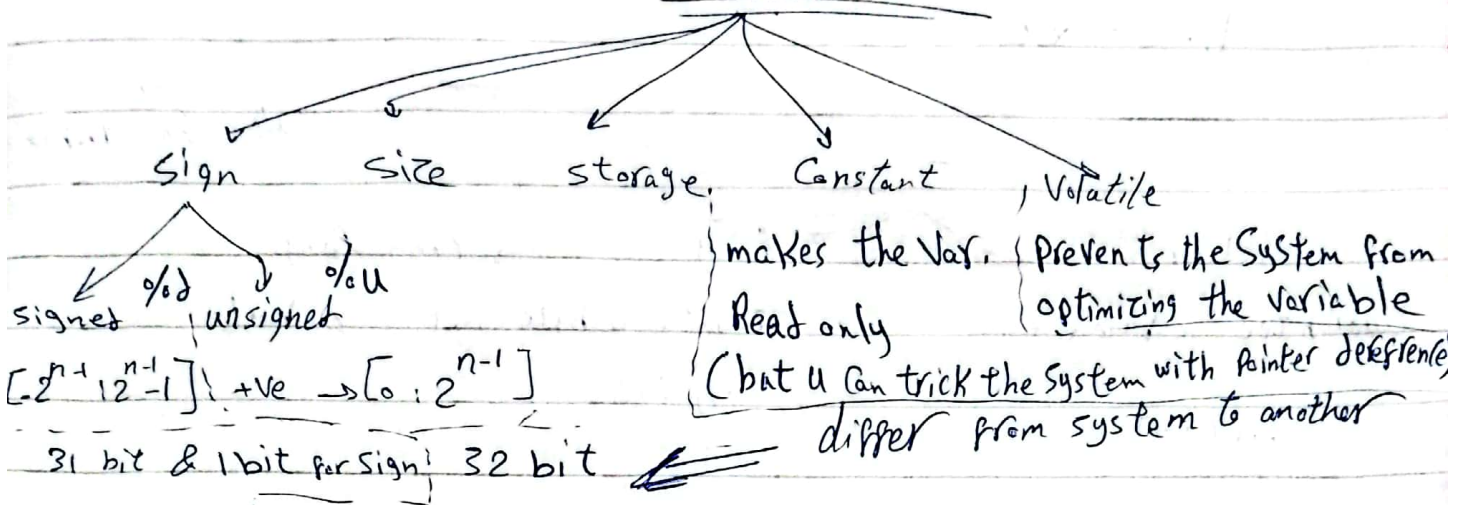


# Task\_4 Report by: Zyad Hesham AbdelWahab

## Data Modifiers



types of storing -ve int

Sign magnitude:

Convert ~~the~~ bit -

Cons: two values of zero.

Ex: 0101 → 1101  
changed the MSB

Some mathematical operations errors  
Like  $+1-1 \rightarrow -2$  not 0.

1's Complement:

Convert every bit from 0 to 1 and vice versa ( $0 \rightleftharpoons 1$ ) & the MSB represents the sign

Ex: 101 → 010  
Cons: two values of zero

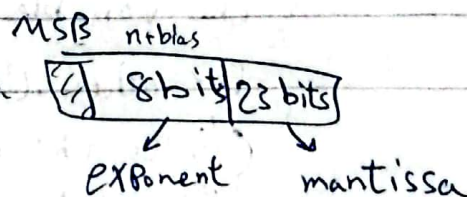
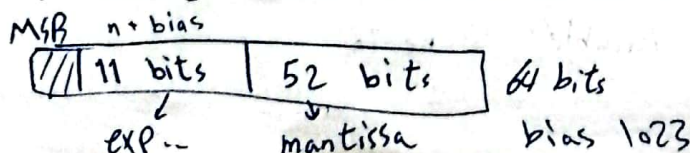
2's Complement:

Widely used in modern computers

Convert every bit ( $0 \rightleftharpoons 1$ ) and add 1, & the MSB represents the sign.

Ex: 101 → 011  
Used for integral only

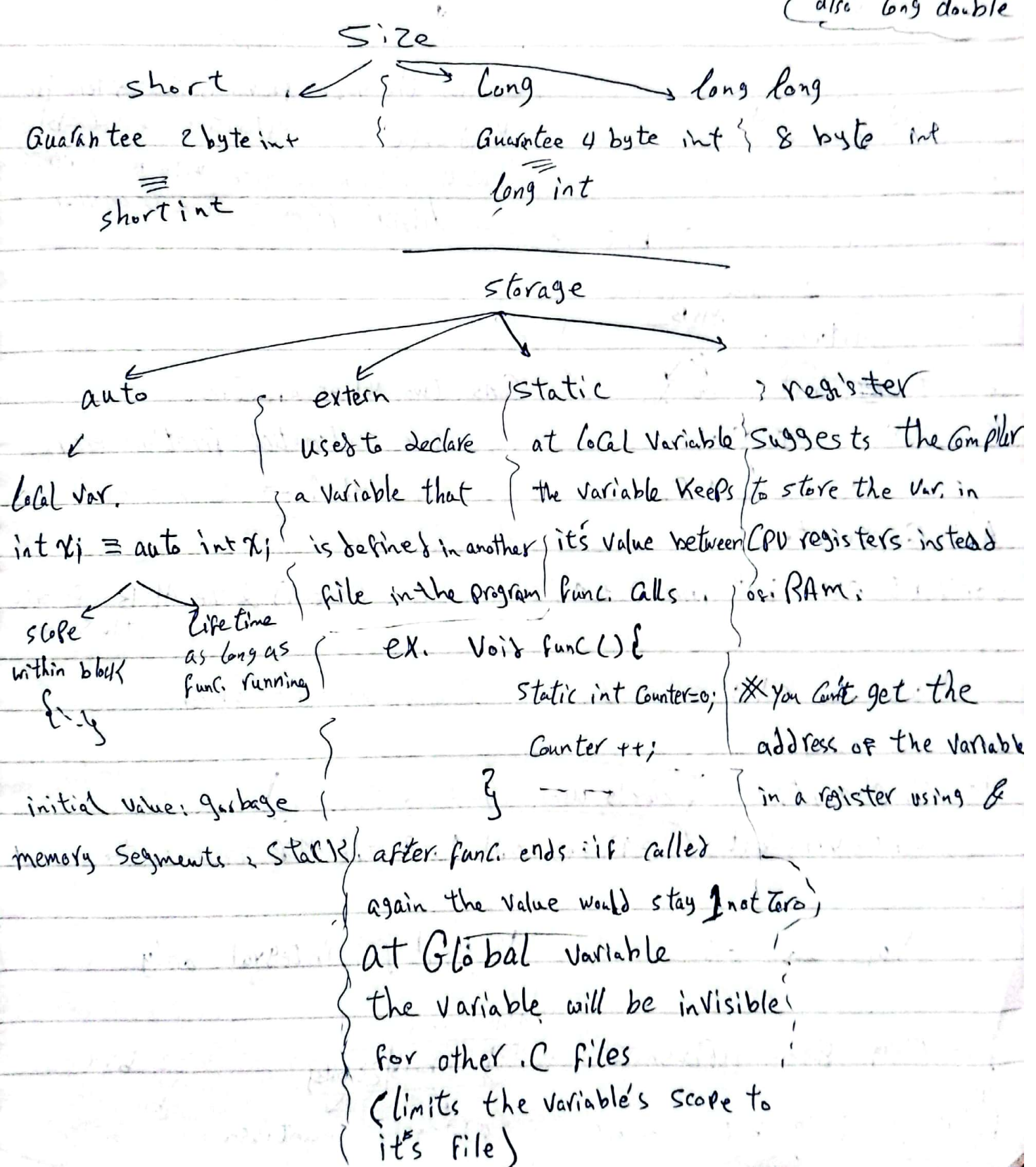
Floating Point representation:



32 bits  
bias 127

Normalization for floating point:  $(4.25)_{10} \rightarrow (100, 01)_2$   
 $1,0001 \times 2^2$  after normalization  
 $1, \text{---} \times 2^2$

there is also long double





# Pointers

Syntax.

Pointee type \* Pointer-Name

ex.

int \* ptr;      ptr = &int-Name

it stores the address of a variable

the size of the pointer differ due to system type

16-bit system : Pointer size 2 bytes

32-bit      "      :      "      4 bytes

64-bit      "      :      "      8 bytes

the Pointee type affects how many data addresses the pointer will hold

ex

```
int x = INT_MAX;
```

```
int *ptr = &x;
```

```
printf("%d", x);
```

```
printf("%d", *ptr);
```

```
printf("%p", ptr);
```

Output.

2107483647

"

0x0fee01 for example

if we used char\* ptr instead of int\* ptr it will read only the first byte of the int.