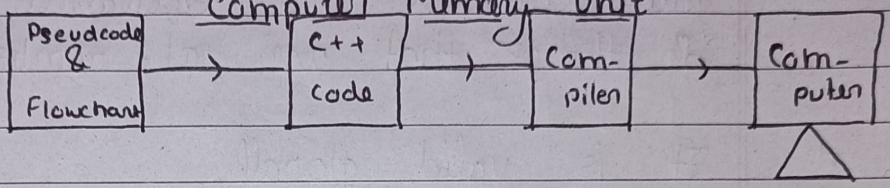
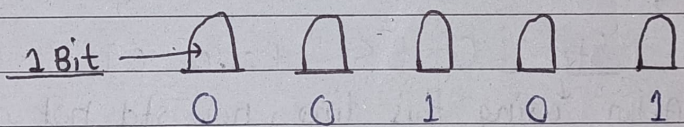
# Day - 3

## Computer Memory Unit




⇒ Compilen is also do error handling and optimization.

⇒ How things store in transistor?




S = 101

⇒ Smallest Unit = 1 Bit

⇒ 1 Bit = Binary
8 bit = 1 bytes
1024 bytes = 1 kB
1024 kB = 1 MB
1024 MB = 1 GB
$2^{10}$ GB = 1 TB
$2^{10}$ TB = 1 PB

⇒ We can't store 'A' directly in the transistor.

⇒ So, we ~~convert~~ A allocate 'A' a no. that is 65.

So, A = 65 = 1000001

⇒ B = 66 , C = 67.

⇉ Now ASCII Table comes in the picture that contains the equivalent number of things that are not number.

⇉ It is valid globally.

* First code in C++

```
#include <iostream>  ⟶ Header File
using namespace std;
int main()
{
    std::cout << " Hello CA ";
(After using this line, now std not requires)
    cout << 2+3;
```

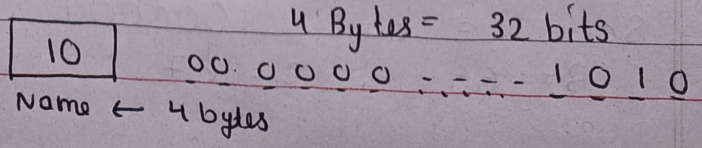
⇉ << ⟶ Insertion Operator

* Variables & Data type

⇉ Char : a, b, c
⇉ Number: 1, 2, 3, 4
⇉ Word: How are you
⇉ Yes or No (Gesture)

Data types

⇉ int name = 10;
Data type ↙ ↘ variable



2. Char (Name)
↳ Variable
(Alpha)

⇒ Name should be readable & easy to understand.

Ex: char c = 'a';
'b';
'1';
'%';

'a' → 97 → 1 byte = 8 bit
c

⇒ 97 = 01100001

3. Float: 1.2, 2.6, 7.823
↙
4 bytes = 32 bit

4. double d = 4.93256789012 } These are used when we have large numbers.

5. long int → 8 bytes

6. Boolean b = 0, 1, true, false;
↳ 1 byte

⇒ For negative numbers, we give half no. of bits. ~~for~~

Ex: $2^{32} \rightarrow 2^{16} \mid 2^{16}$
↙ For -ve no. ↘ For +ve no.

| | |
|---|---|
| 100 → 0 | 000 → 0 |
| 101 → -1 | 001 → 1 |
| 110 → -2 | 010 → 2 |
| 111 → -3 | 011 → 3 |

So here, 0 representing by 100 & 000 both

So for this, we make a pattern —

100 → -4
101 → -3
110 → -2
111 → -1

So for converting ~~to~~ negative no. into binary —

$-2$

0 1 0

2's com. 1 0 1

2's com. 1 0 1

+ 1

1 1 0 → $(-2)_{10}$

int 32 bit

$2^{31}$ $2^{31}$

$-1, -2, \ldots -2^{31}$ $0, 1, 2, \ldots 2^{31}-1$

So zero will go to the +ve side.