

~~TRY~~

Recursion

1 - A
2 - B
3 - C
:
:
26 - Z

A curly brace groups the first 26 letters from A to Z.

number

1 2 3

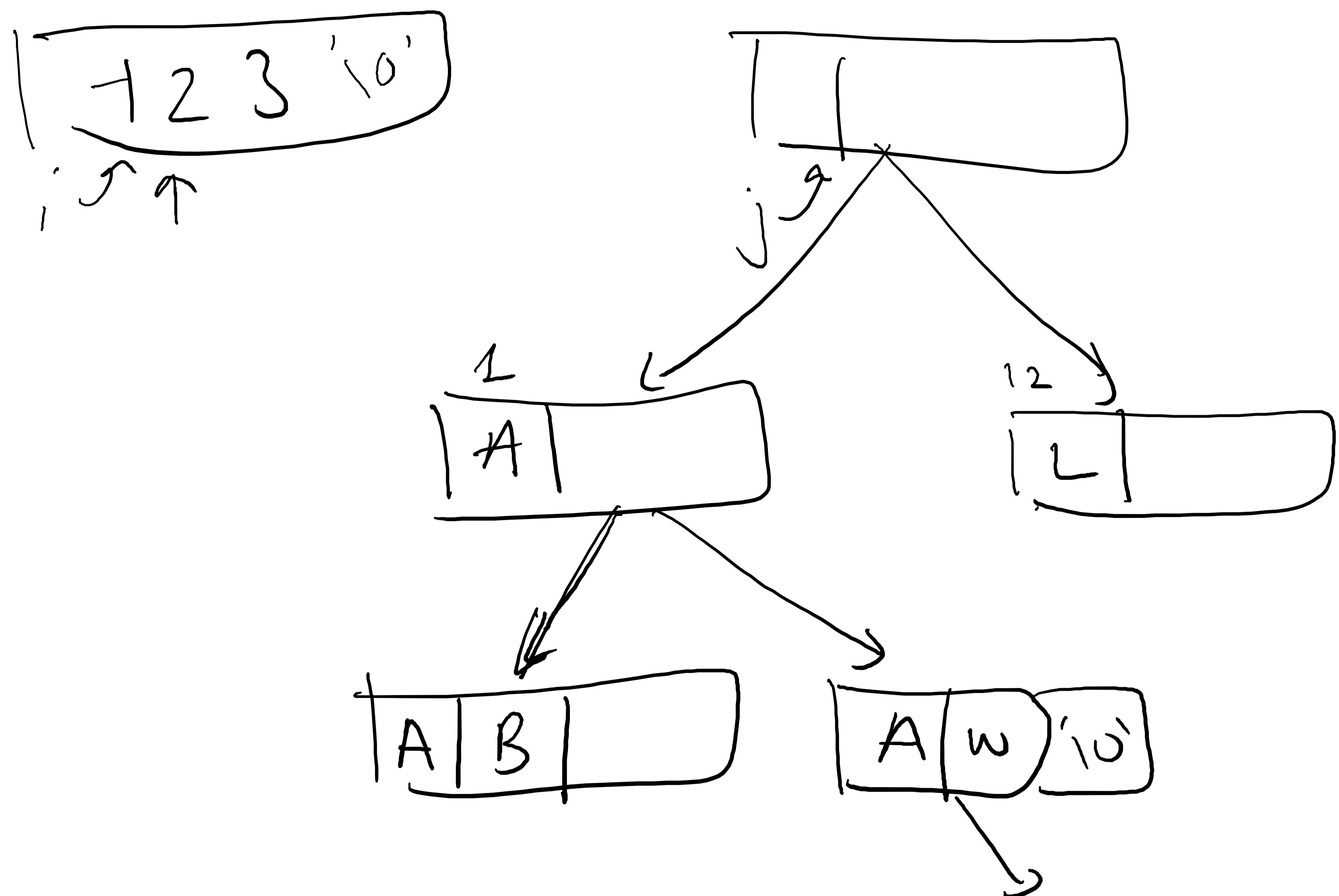
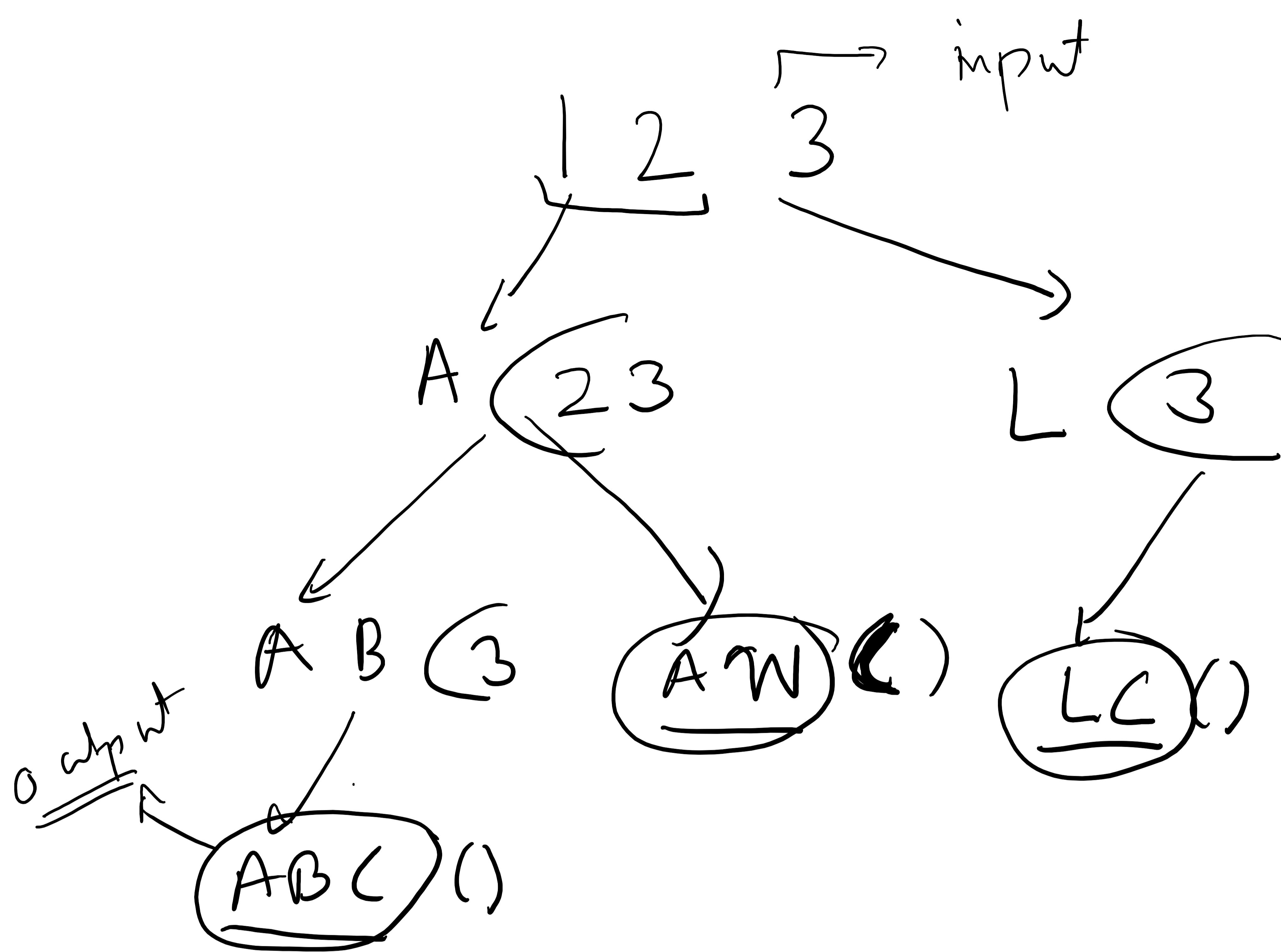
A B C]

L C]

A W]

12,3
1,23

A curly brace groups the numbers 1, 2, 3 and the letters A, B, C. Another curly brace groups L, C and A, W.



`Car(Car& X) {`

`model_no = X.model_no`

Deep Copy

`name = new char(len);
strcpy(name, X.name);`

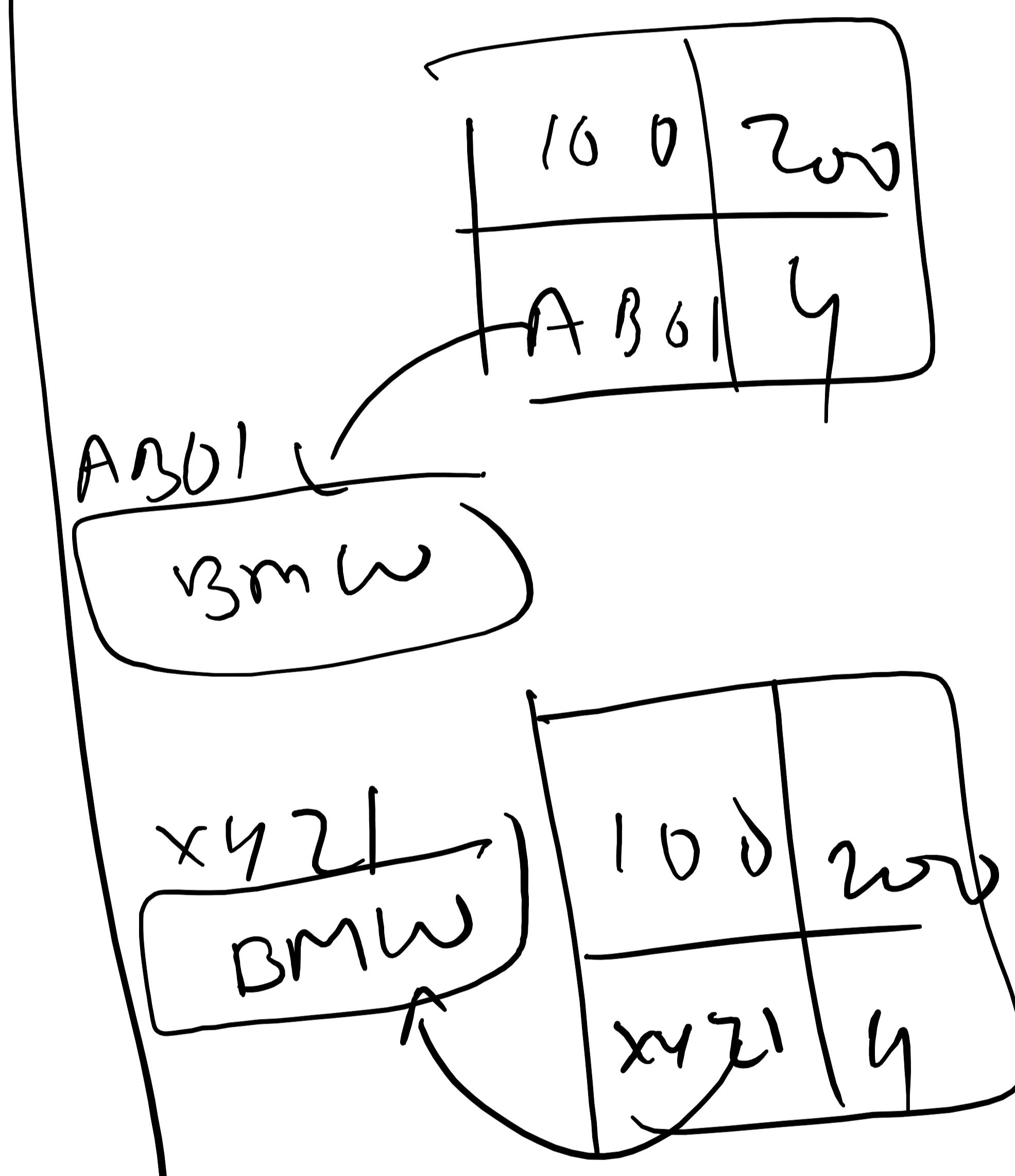
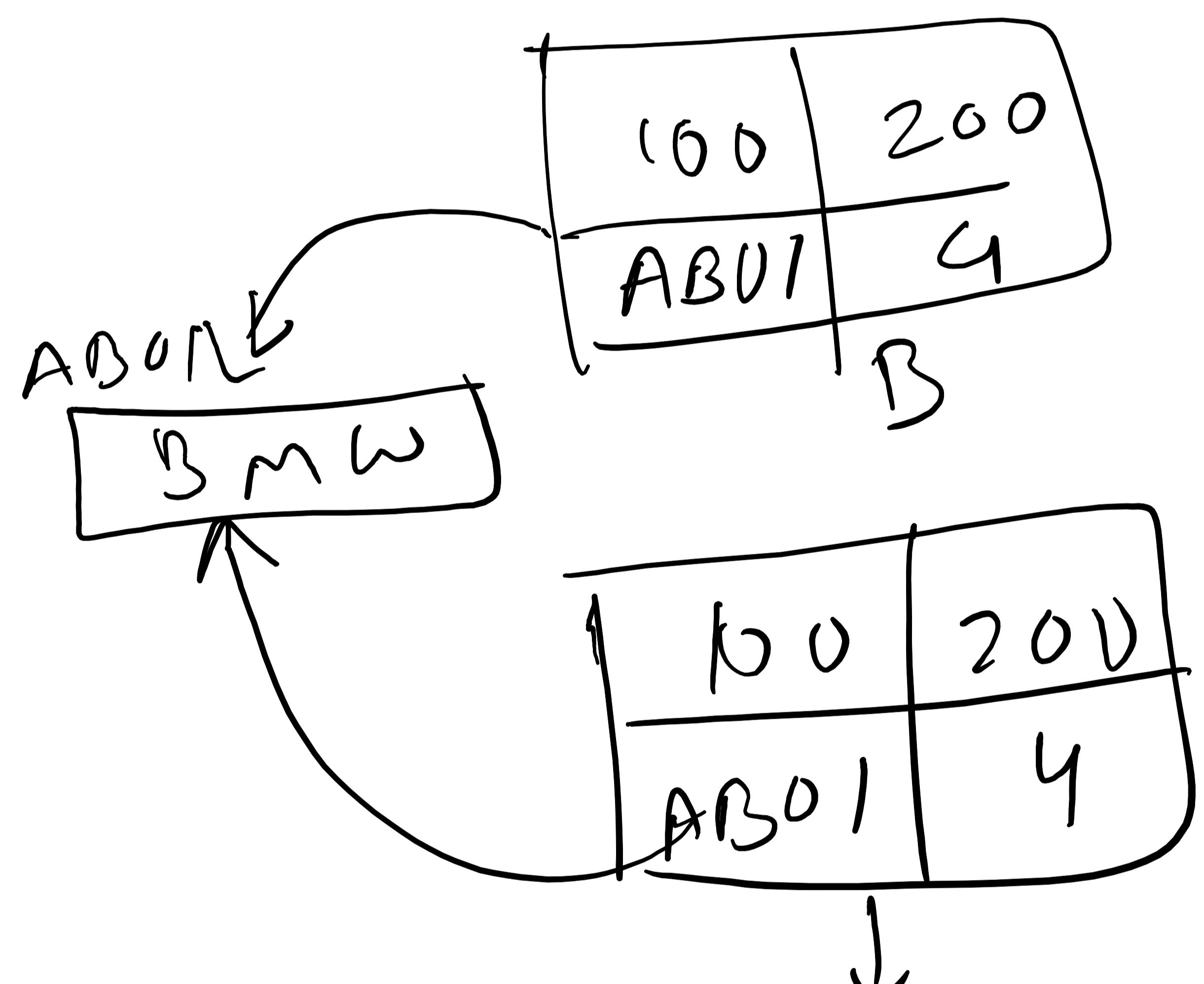
Shallow copy

~~`name = X.name;`~~

`Car C(B);`

Shallow copy

Deep Copy



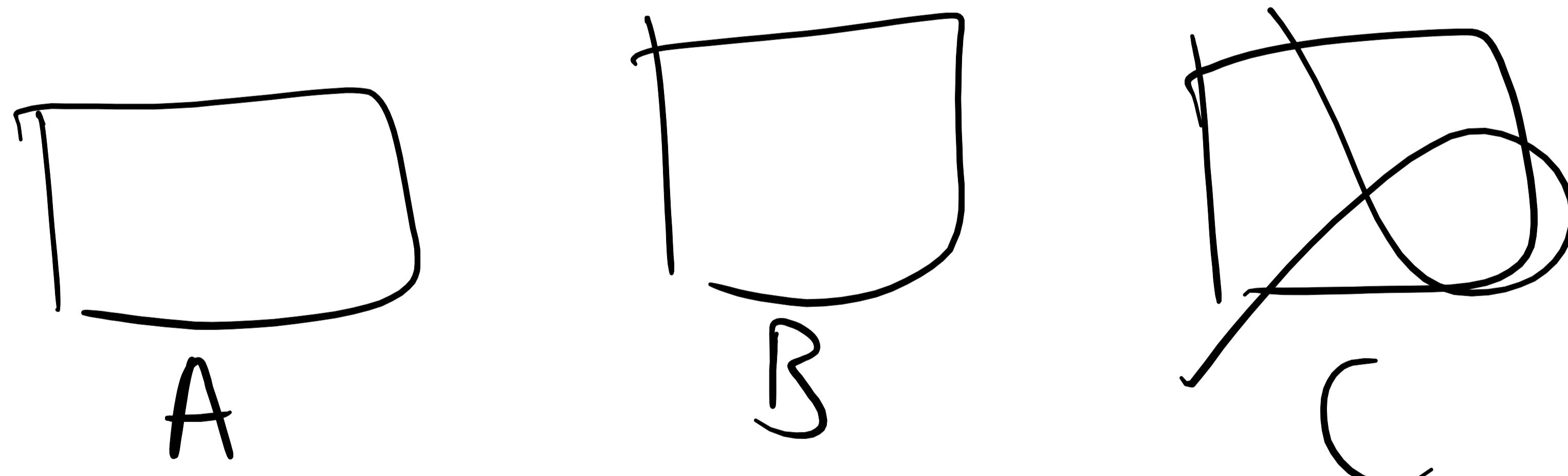
Operator Overloading

RT operator = (CarB) {

3
~~model1 = B.model;~~
~~sets = B.sets;~~
~~name = new char(len)~~
~~strcpy(name, B.name);~~

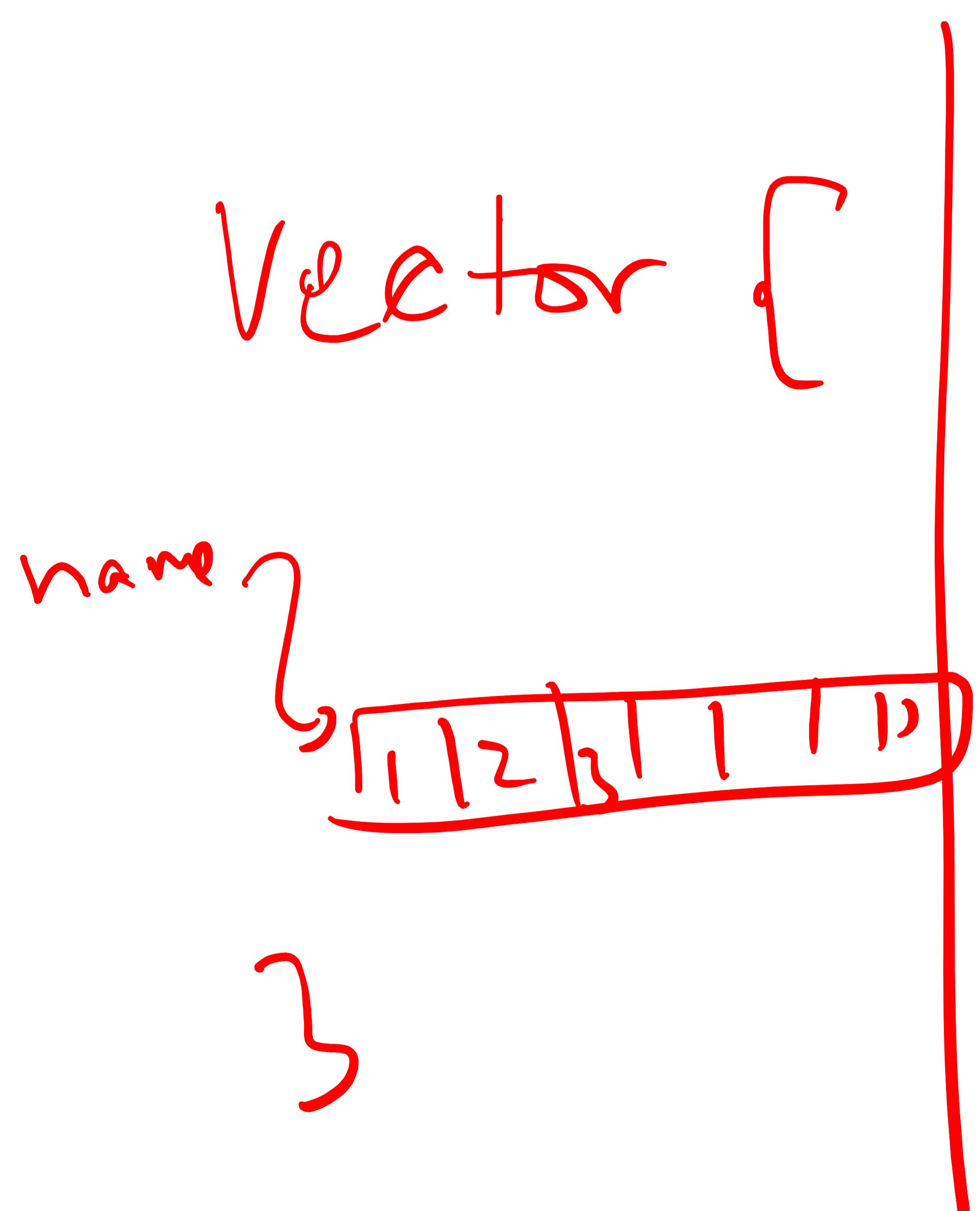
A = B,
A. = (B),
A.equal(B);

2
Count



Hello

int x = c + b;



vector b;

v.push-back(1);

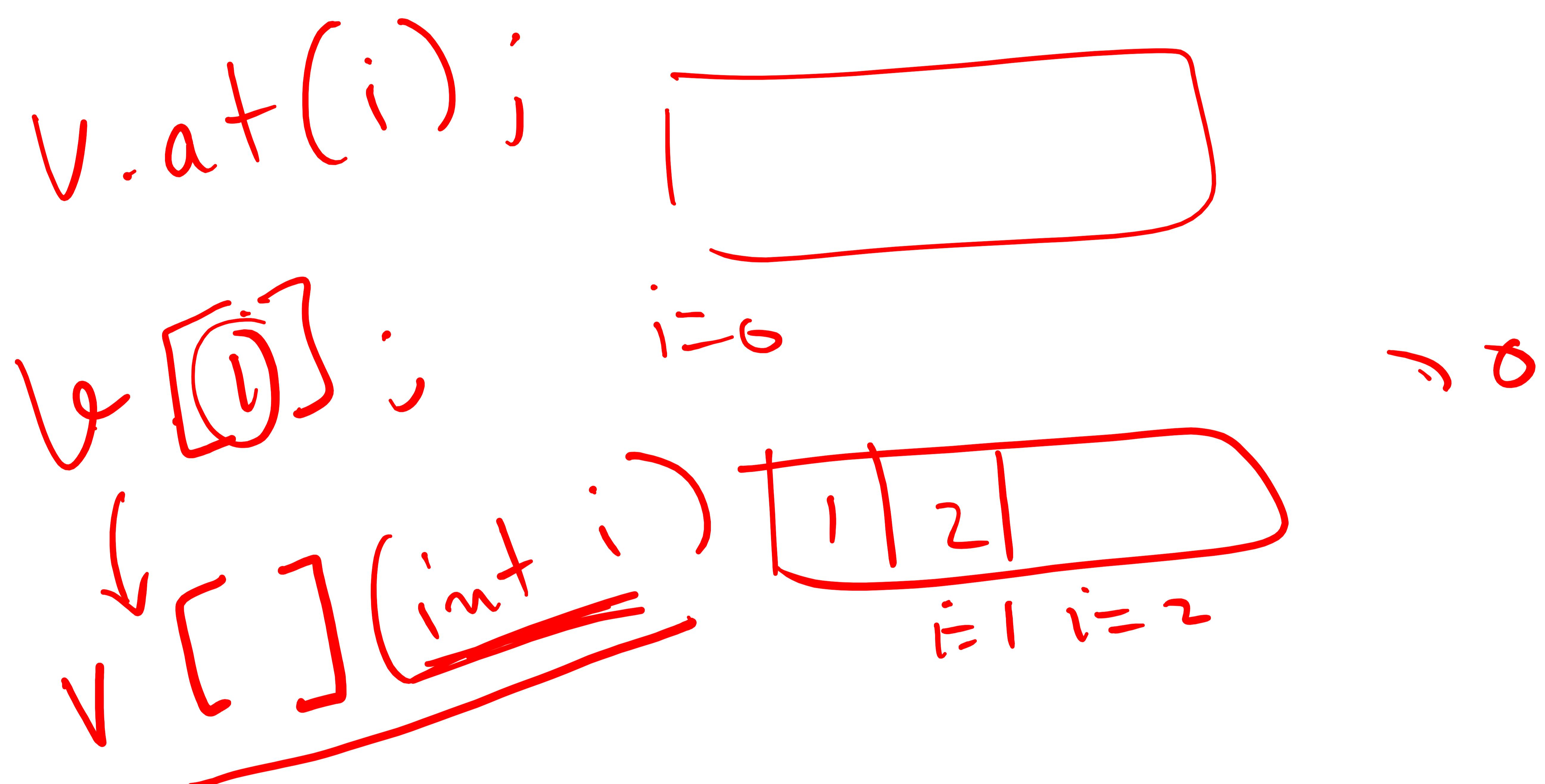
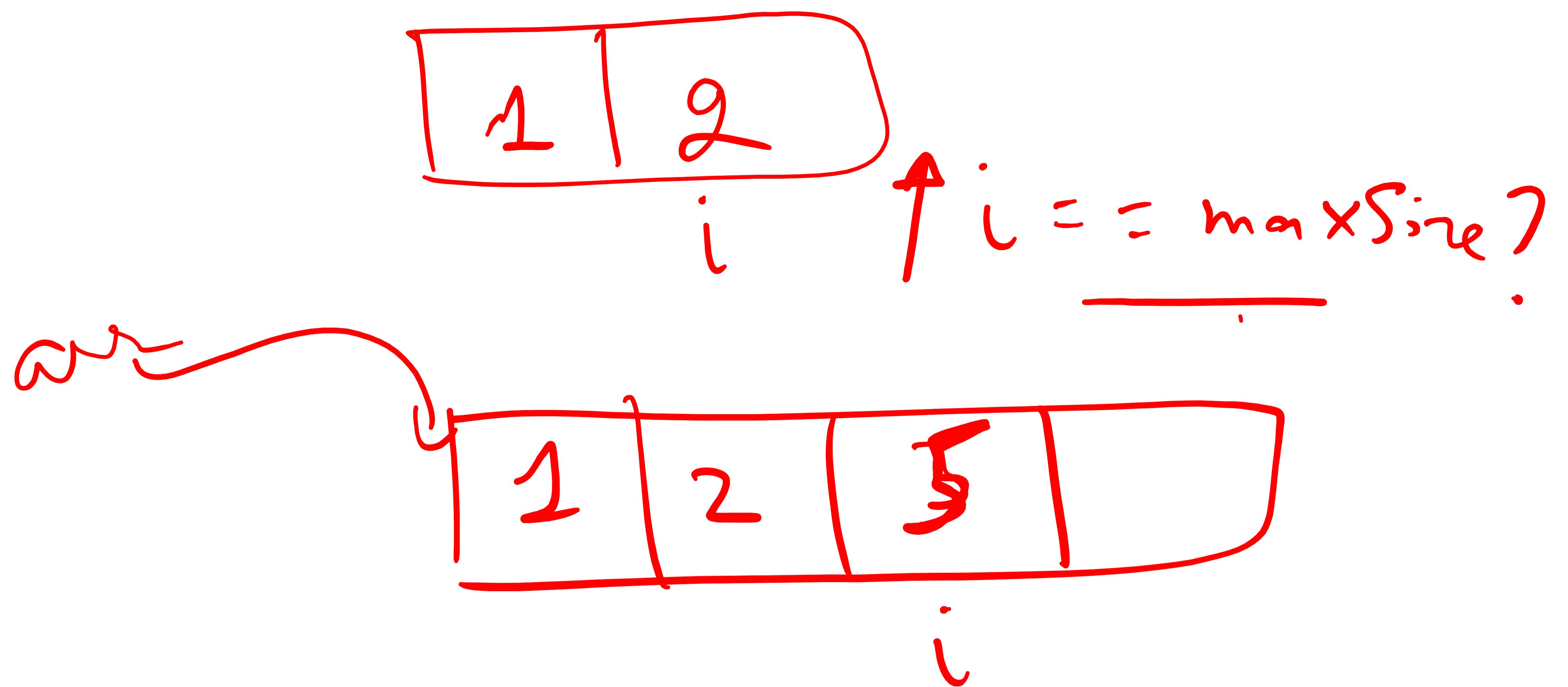
v.pop-back(k);

v1 + v2 ;

1 1 2 1 3 1 4 1 3 1 6

v.print();

Const / copy / Dest

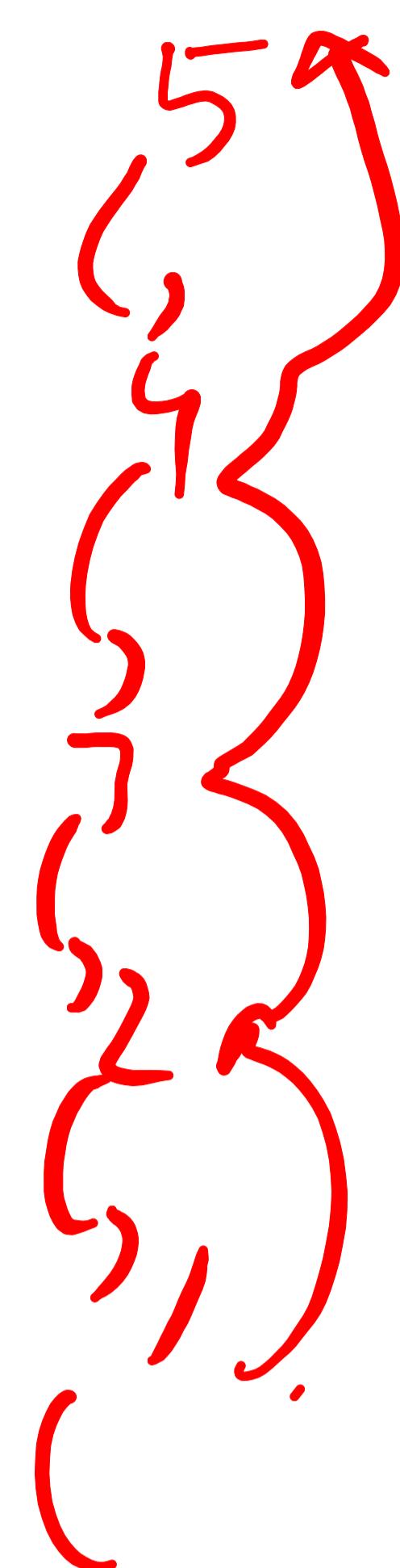


~~$N!$~~ = $\text{for } (i = 0 \underline{\underline{= N}})$

$\underbrace{K}_{\downarrow} \quad \underbrace{\text{ans} = \text{ans} * i}_{\downarrow}$

KN
 $O(n)$

Recursion



$$\underline{KN} = O(N)$$

Fibonacci

① Iteration

$$\begin{aligned} a &= 0 \\ b &= 1; \end{aligned}$$

for (n times)

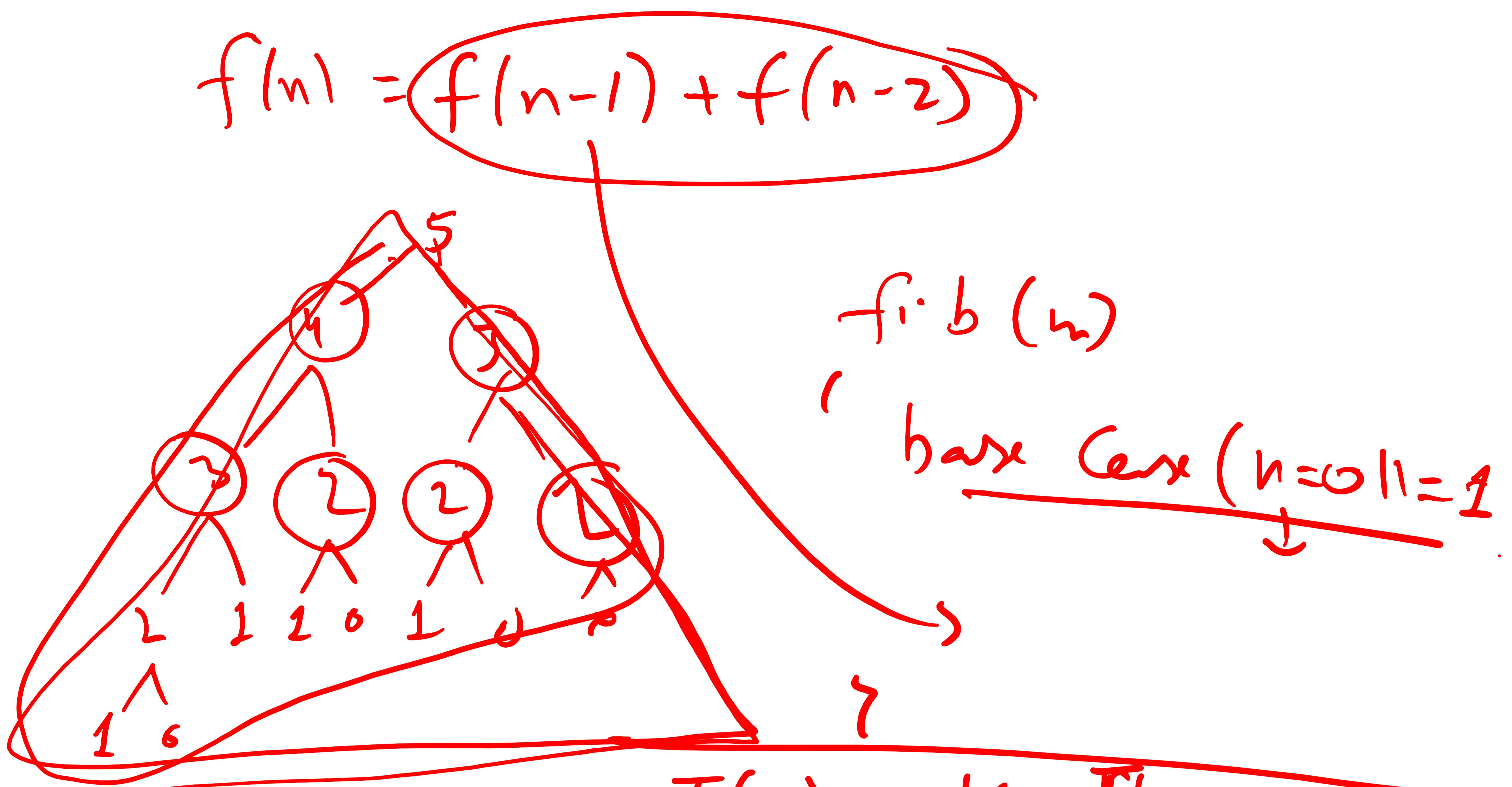
$$\begin{aligned} c &= a + b; \\ a &= b; \\ b &= c; \end{aligned} \quad \left. \begin{array}{l} \\ \\ \end{array} \right\} \rightarrow k$$

$O(N)$

②

Recursion





$$T(n) = K + \underline{f(n-1)} + \underline{T(n-2)}$$

$K(1+2+4+\dots 2^{n-1})$

$$1 \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 = \frac{2^n - 1}{2 - 1} = k2^n$$

$$\approx O(2^n)$$

$$a \cancel{x^2} + bx^1 + cx^3 + d$$

$$\cancel{N} + N-1 \quad N-2 + \dots$$

$$\Rightarrow \approx O(N^2)$$

$$= O(N)$$

$$K \left(((k-1) + (k-2) + (k-3) + \dots) \right)$$

$$= K^2 + (n - k)$$

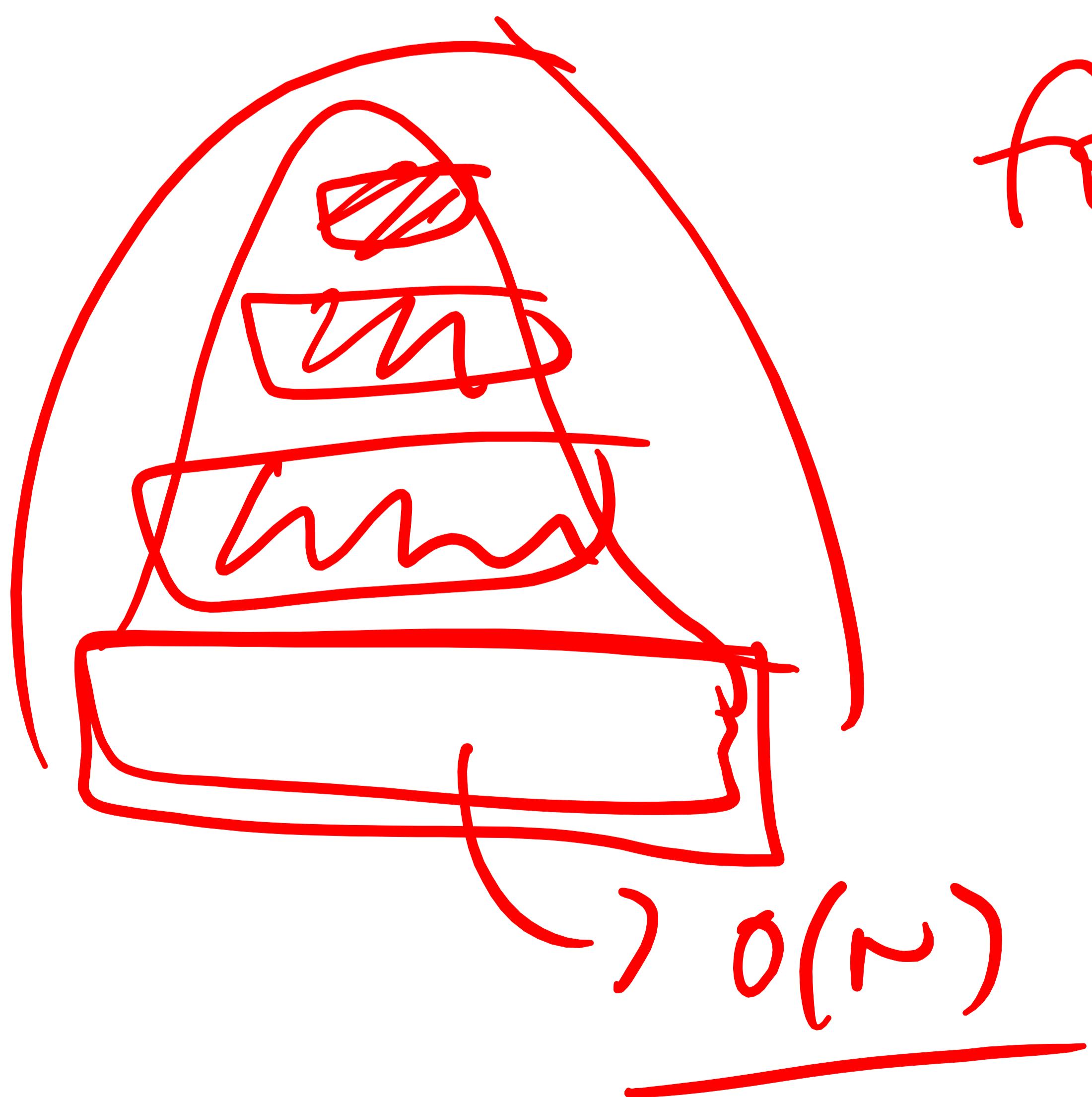
2 loops second half

$K \rightarrow O(1)$ space.

$\rightarrow O(N)$ Space
 int * arr;

```

for ( i=0 ; i<n ; i++ )
{
  delete[] arr;
  arr = new int[i];
}
  
```



$$\underline{\underline{O(n)}} = O(N^2)$$

