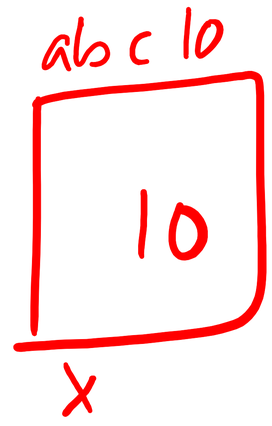
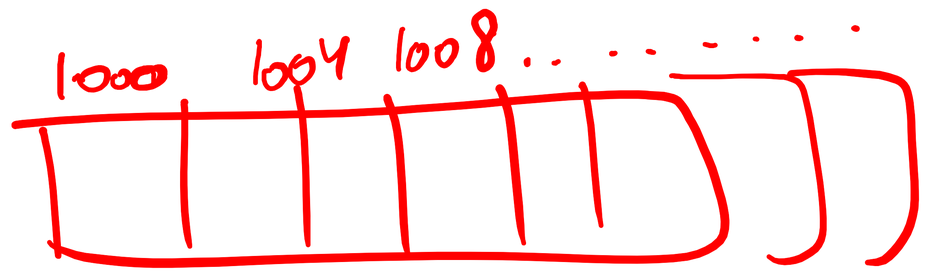
int x = 10



&& logical end

cout << &x ;

int a[10]



a & b

y Bit wise Bang

cout << a ; | 1000

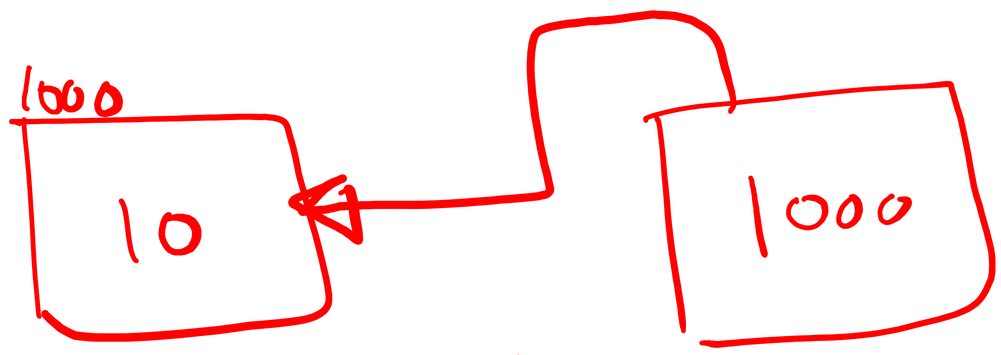
cout << &a ; | 1000

cout << &a[0] ; | 1000

cout << &a[1] ; | 1004

POINTER

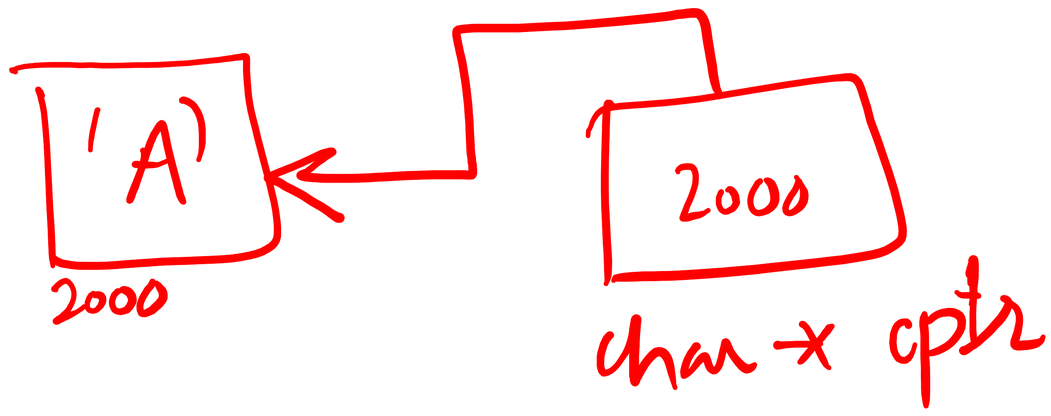
int x = 10

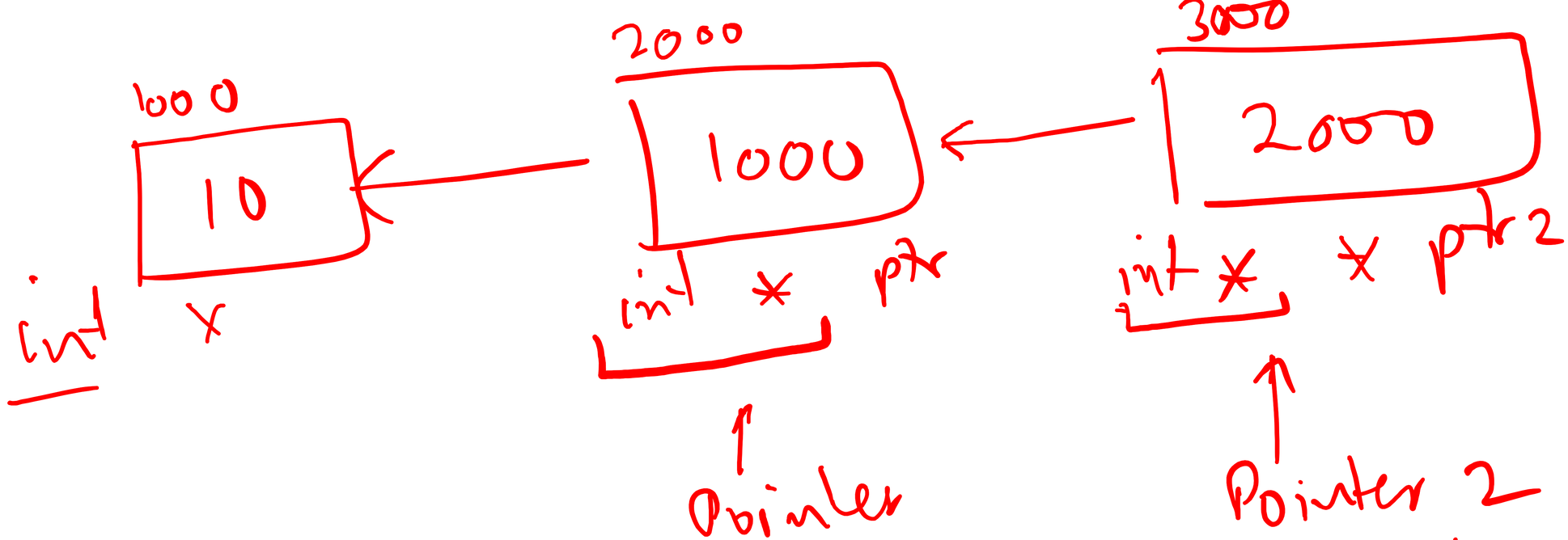


int * ptr ;

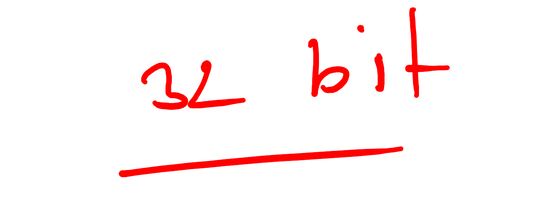
ptr = &x ; (Assign) int

char * cptr = &ch ; (Initialize)





32 bit | 64 bit



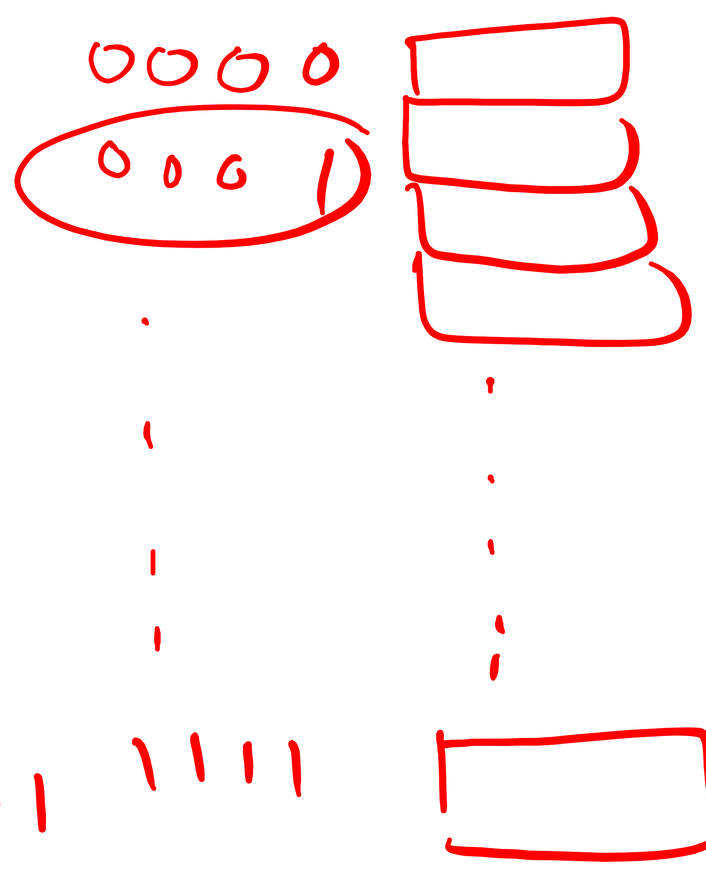
32 bits

Pointer 4 bytes

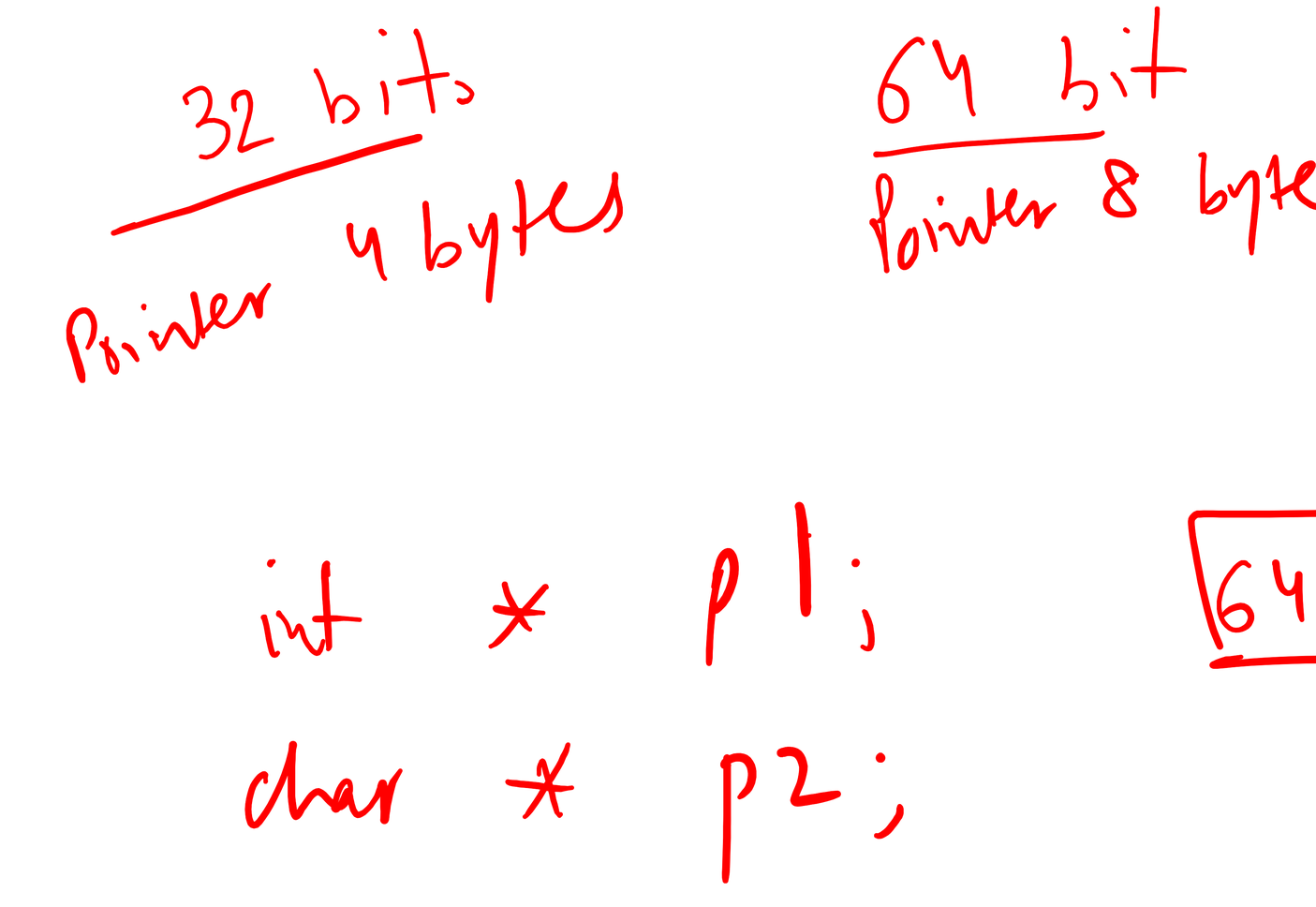
64 bit

Pointer 8 bytes

4 bit




```
int * p1;
char * p2;
```




```
int a = 5000;
char b = 'A';
int *ptr1 = &a;
char*ptr2 = &b;

cout<< * ptr1 <<endl;
cout<< * ptr2 <<endl;
```

Dereference *

- a * b
- Pointer
- Dereferencing

&Bucket = Address

*Address = Bucket

cout << x; → 10

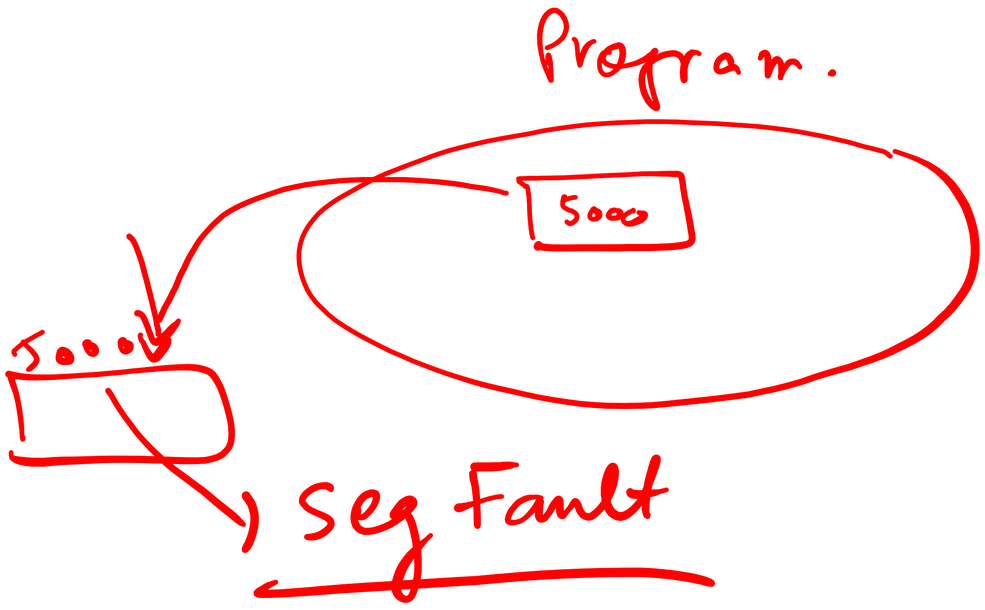
cout << &x; → 1001

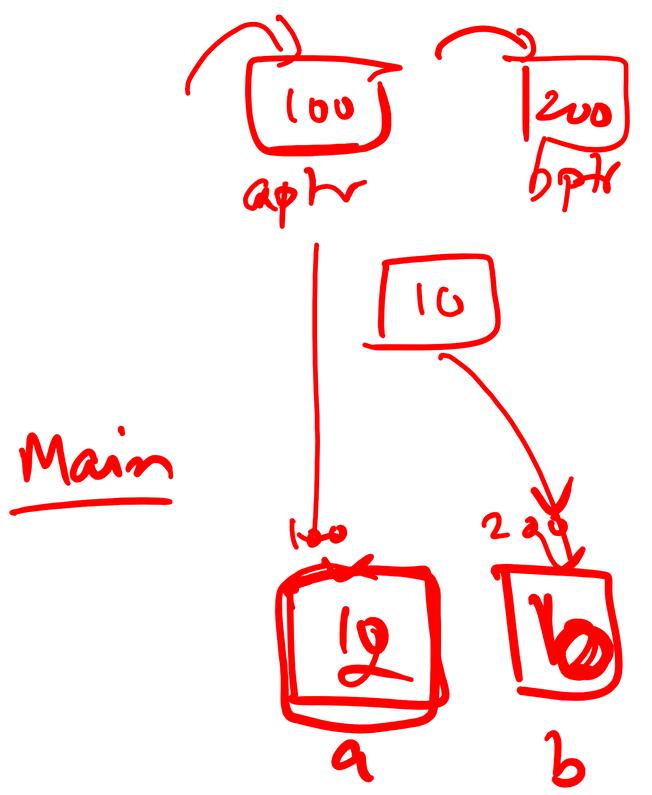
cout << *(&x); → 10

int *ptr = &x;

cout << ptr → 1001

cout << *ptr → 10





*(a+0)

*(a+1)

*(a+i) = a[i]

int x = 10;

10

5 × 4 = 20 bytes

a = 4 bytes.

Symbol Table

Key	Value
a	1000
x	400

x = 10 y = 20

int *ptr = &x ; ✓

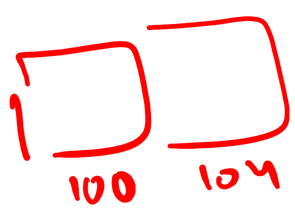
ptr = &y ; ✓

int a[10]; ✓

ptr = a ; ✓

a = &x ; (x)

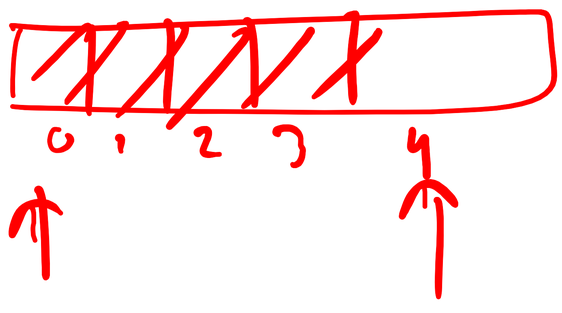
a = &y (x)



ptr2 = ptr + 1 ;

ptr2 - ptr1 = ?

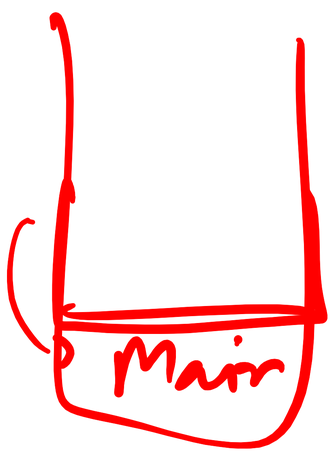
100 - 200 = (-25)

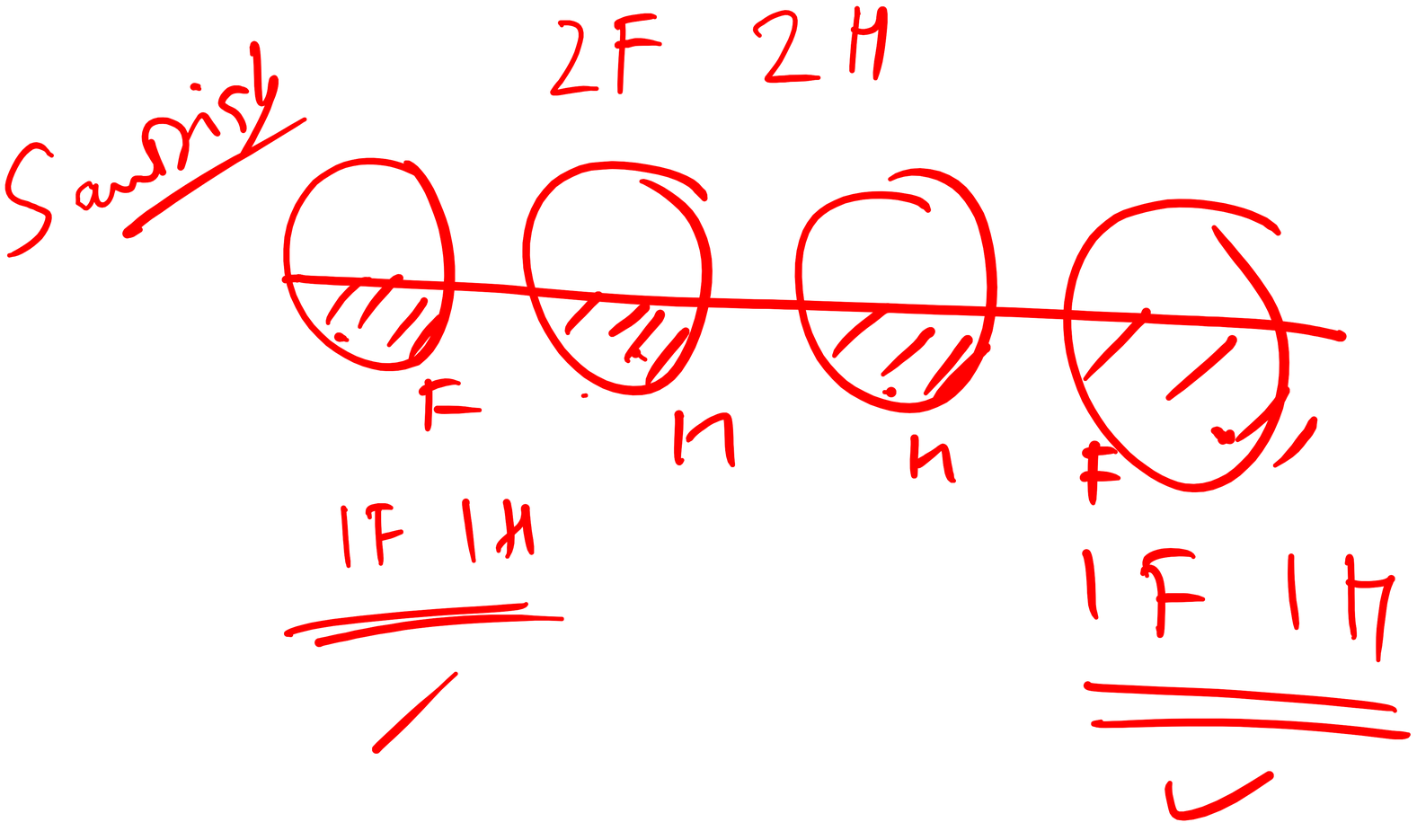


5, 6, 7, 8, 9, 10
11, 12, 13, 14

```
int sum(int a[], int n)
{
    for(i=0; i<n; i++)
        sum += a[i];
```

ROM → a = a + 1





CF-1

$\underbrace{-1+2}_{1}\ \underbrace{-3+4}_{1}\ \underbrace{-5+6}_{1}$ n even $= n/2$

$\underbrace{-1+2}_{1}\ \underbrace{-3+4}_{1}\ \underbrace{-5}_{-5} = -3 = -\left(\frac{n+1}{2}\right)$