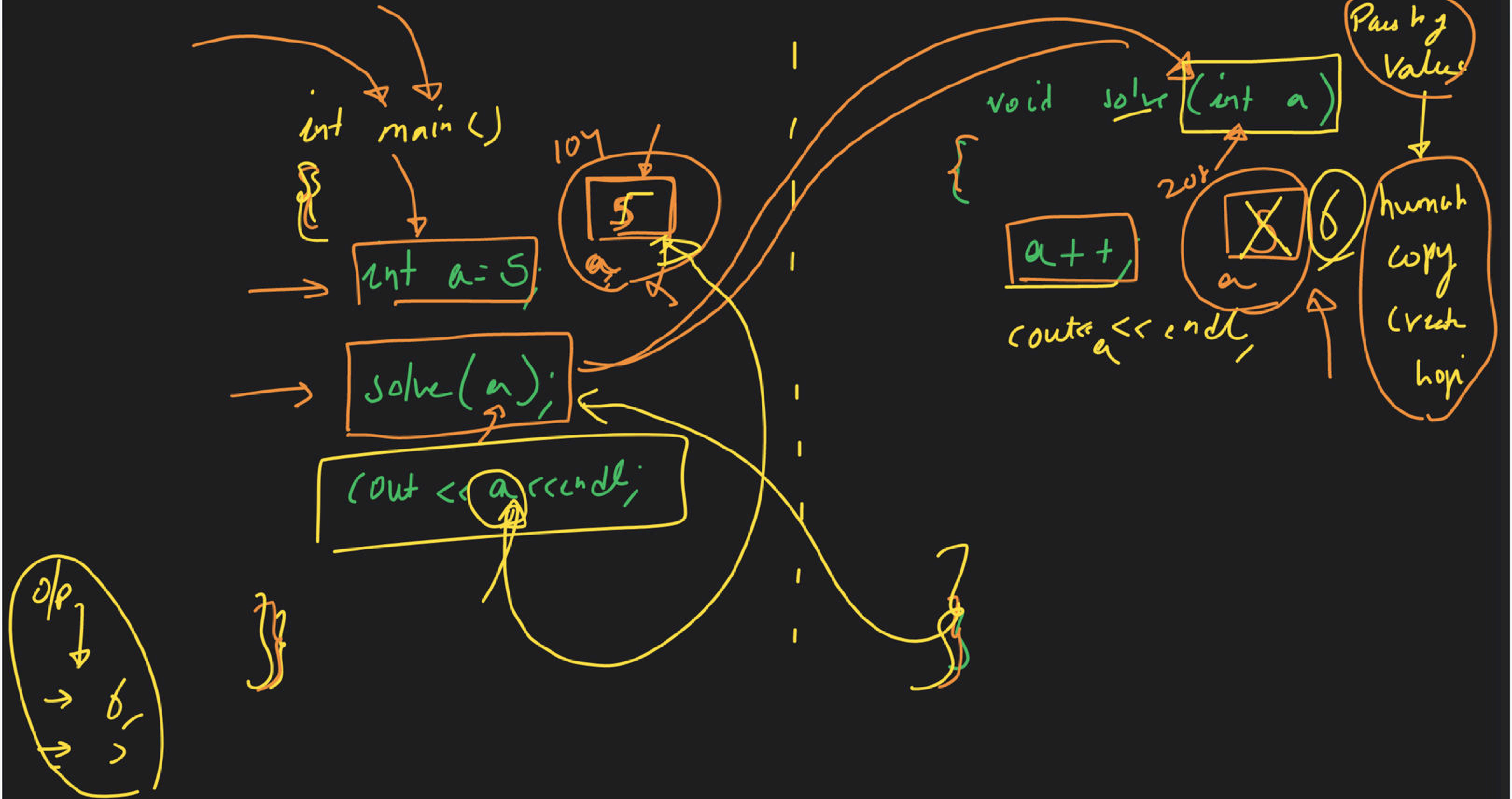
Array Level-2 [Join Here]

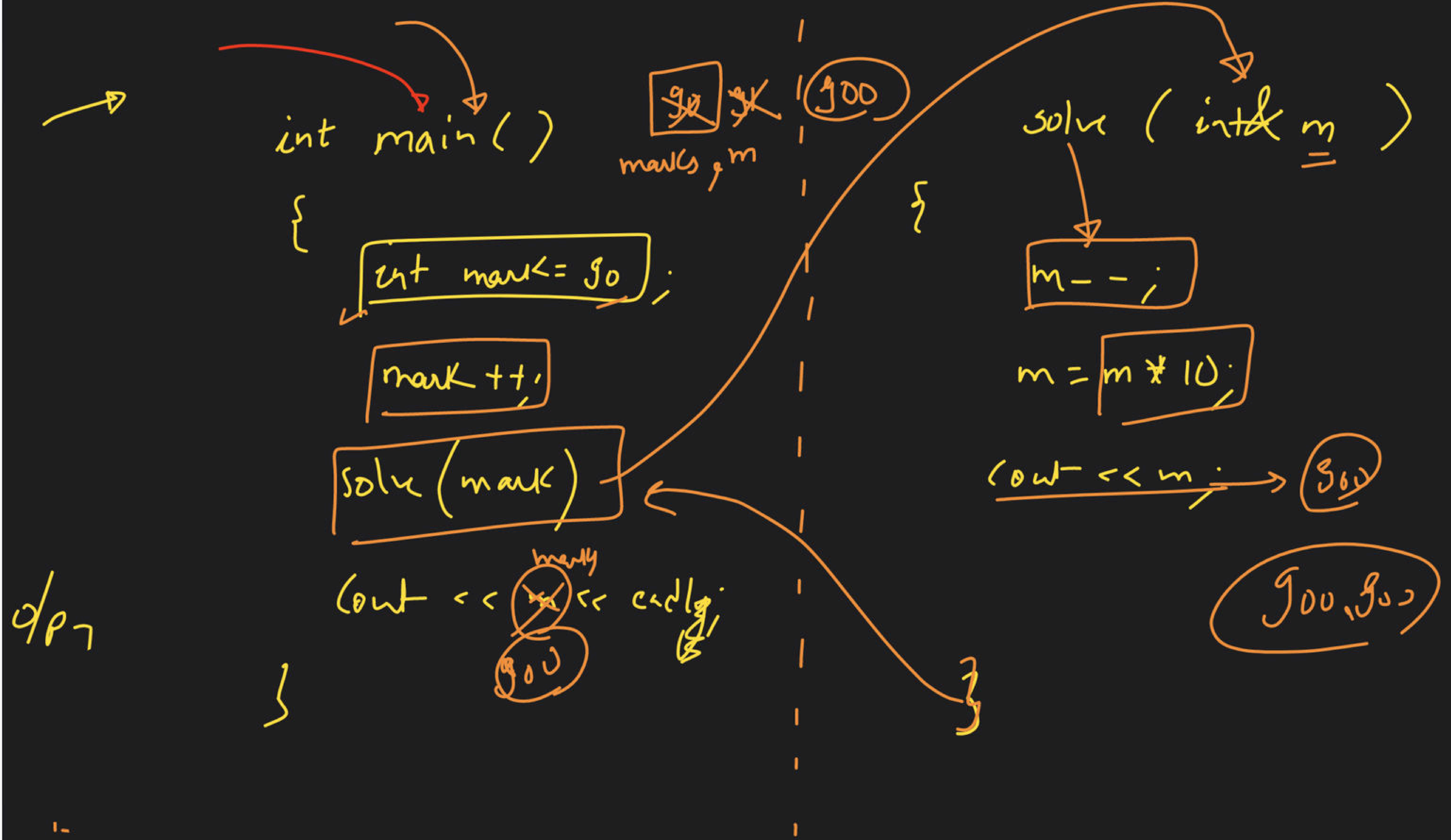
Special class

Love Babbar • Sept 8, 2023



```
int main()
{
    int a=5;
    solve(a);
    cout << a << endl;
}
```

```
void solve(int a)
{
    a++;
    cout << a << endl;
}
```



```
int main( )                         solve ( int& m )
{                                   {
    int marks = 90;                     m--;
    marks++;                            m = m * 10;
    solve(marks)                        cout << m;
    cout << marks << endl;          }
}
```

```
int main()
{
    int sundari = 100;
    sundari--;
    sundari -= 5;
    solve(sundari);
    cout << sundari;
}
```

~~100~~ ~~99~~ ~~94~~ 93

sundari jaadu

sundari -= 5 ⟷ sundari = sundari - 5 = 99 - 5 = 94

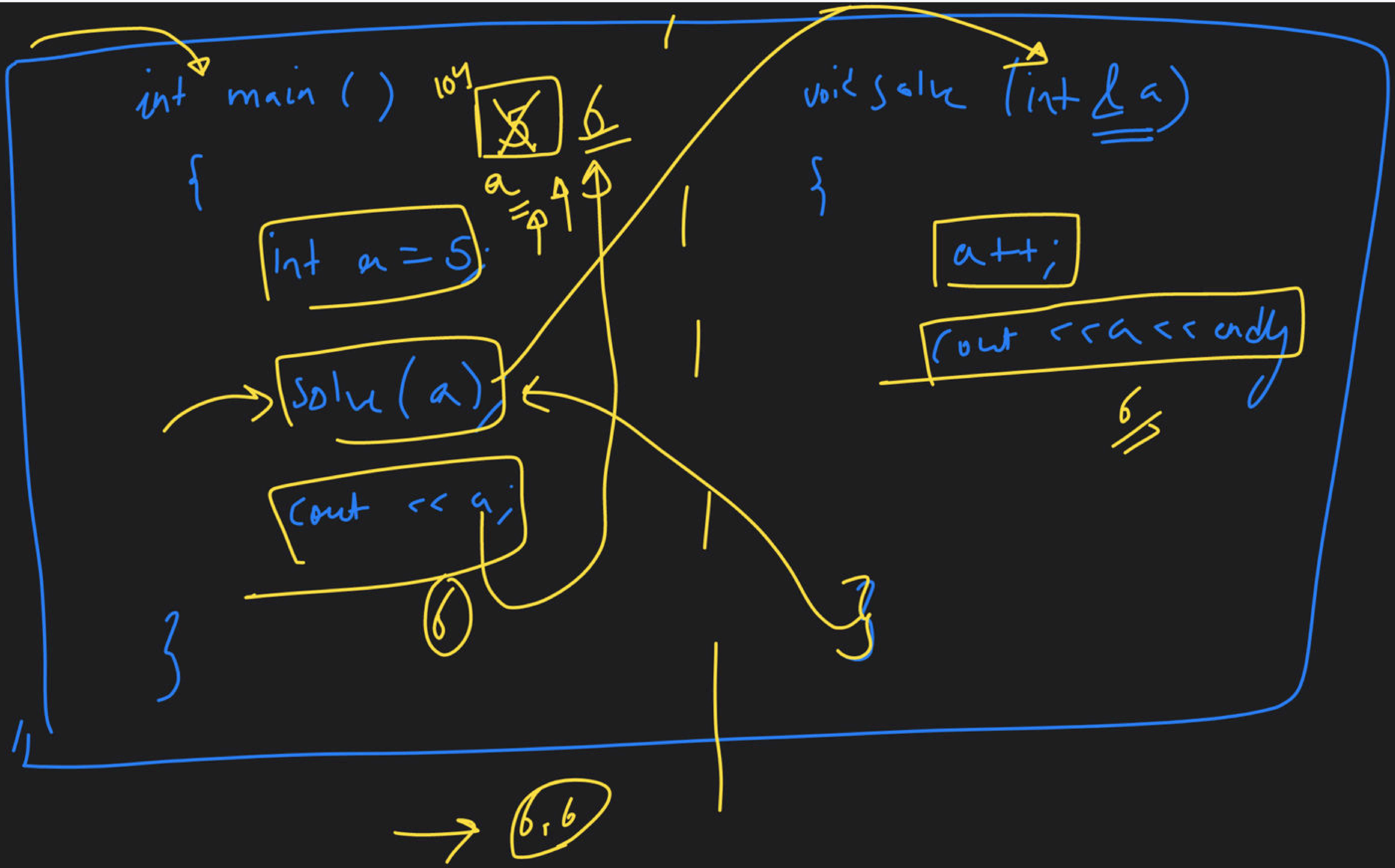
93

```
void solve(int& jaadu)
{
    jaadu--;
    cout << jaadu + 10;
    return;
}
```

93+10 → $\boxed{103}$

O/P

103 93

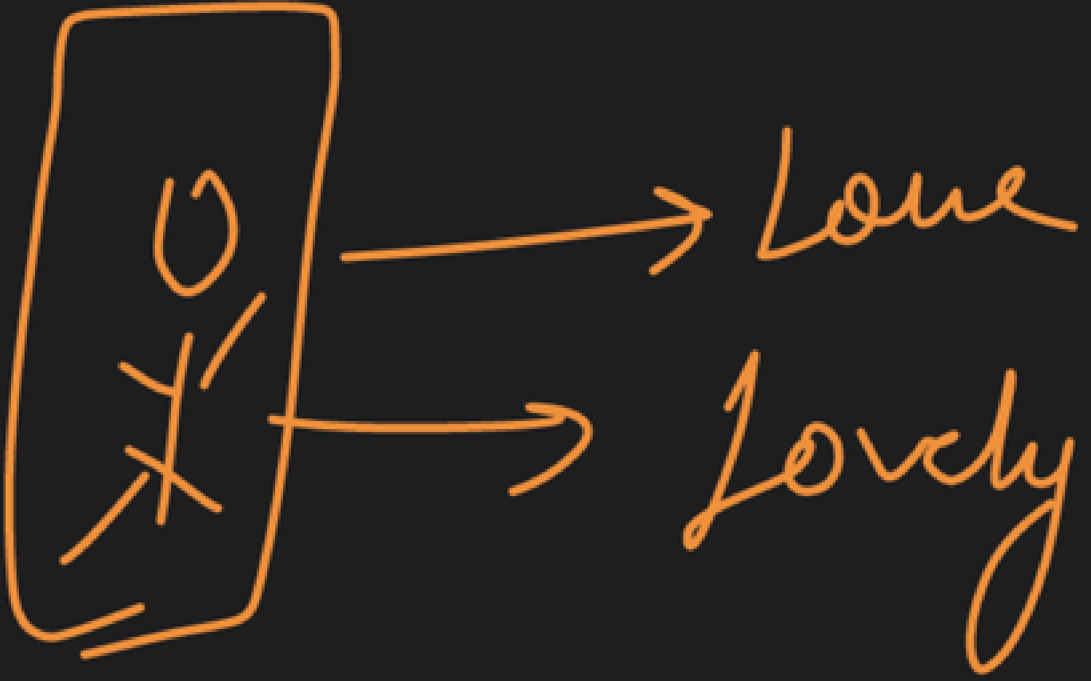
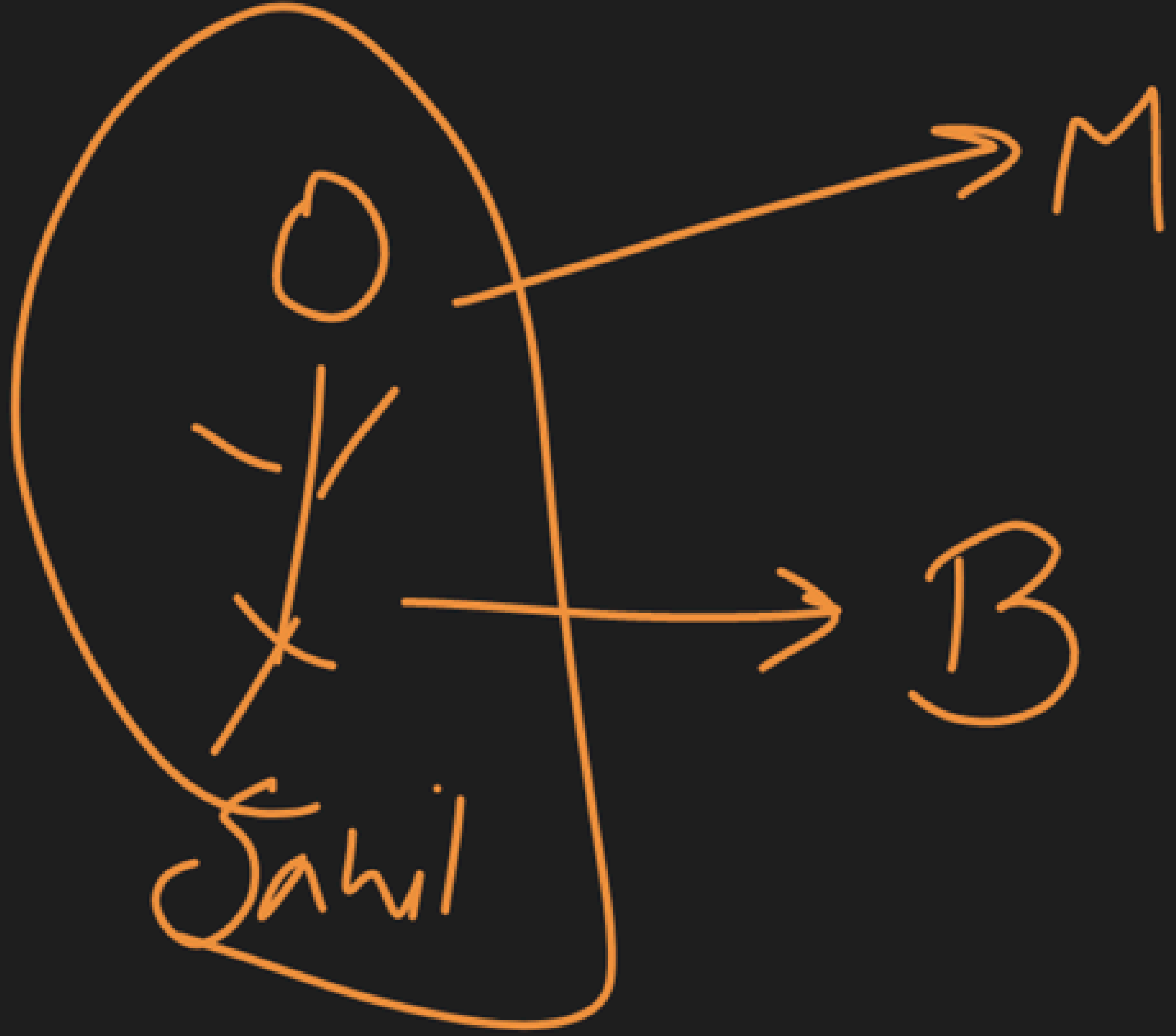
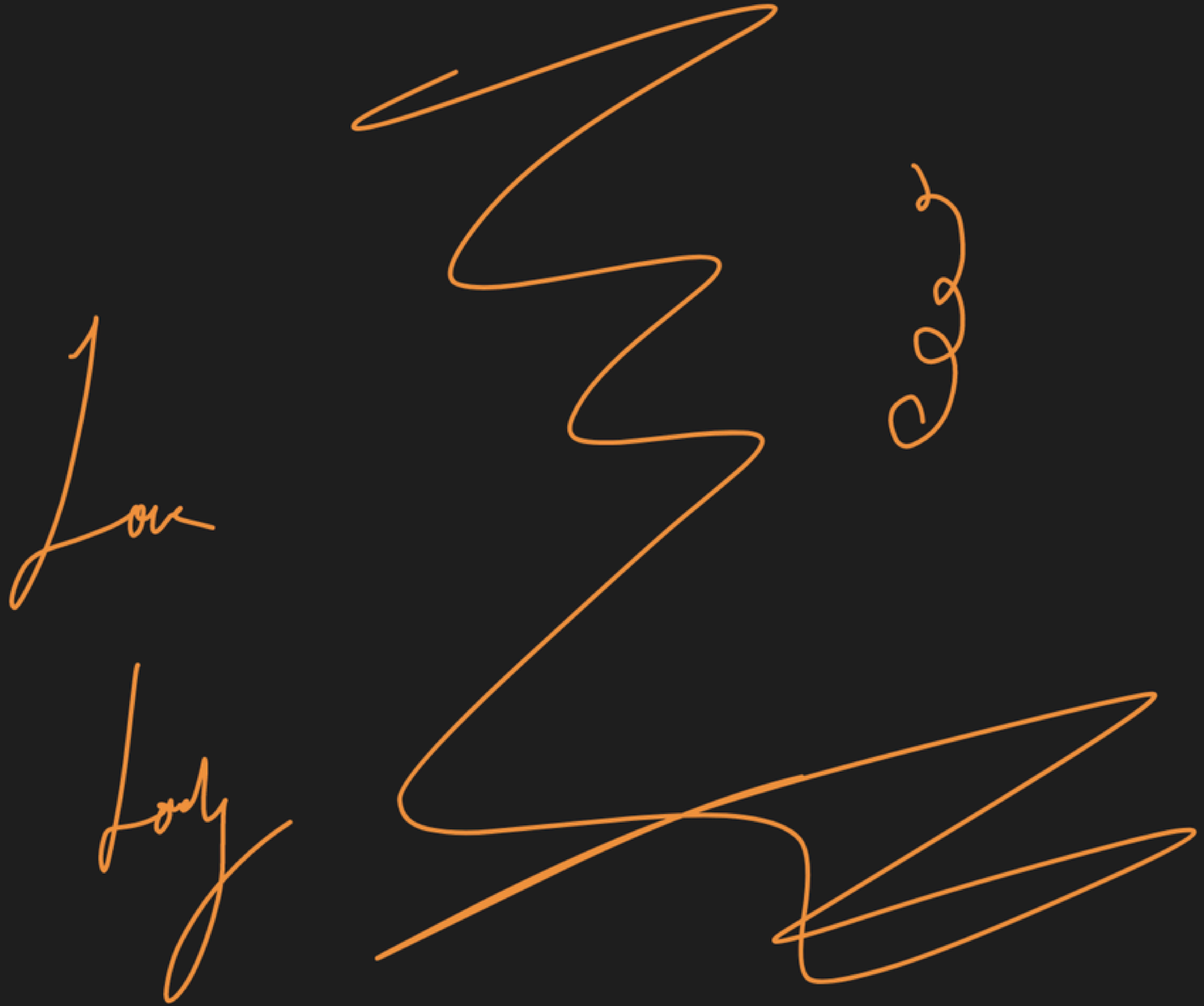


```
int main()
{
    int a = 5;

    solve(a);

    cout << a;
}
```

```
void solve(int &a)
{
    a++;
    cout << a << endl;
}
```

Love

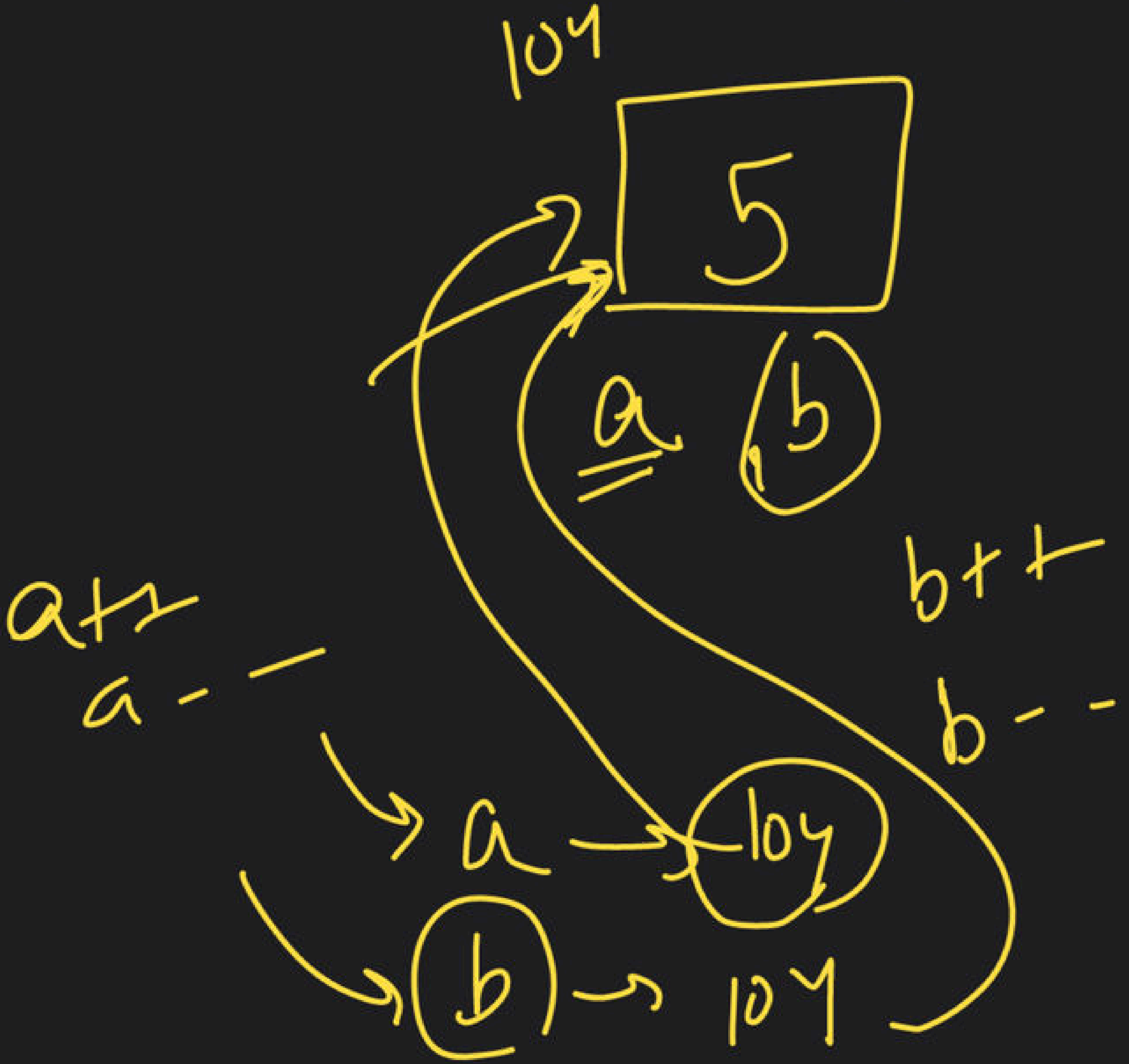
Lovely


Love
Lovely

M
B
Sahil

Love
Lovely

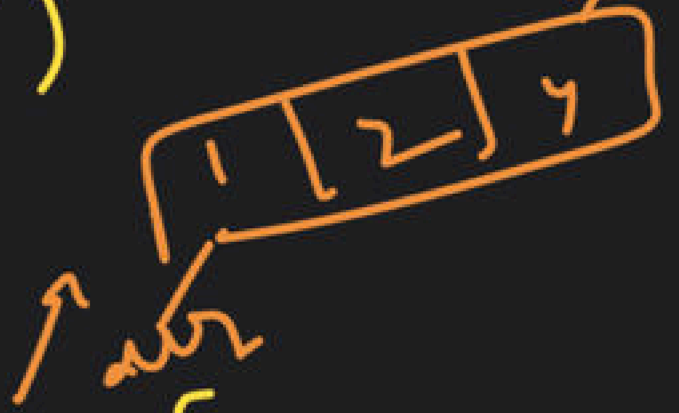
```
int a = 5
int & b = a
```

ref. variable

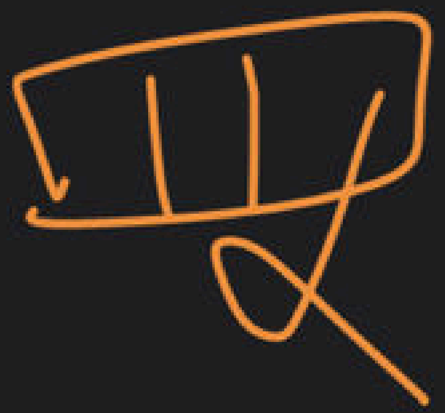
→ same mem location but different names

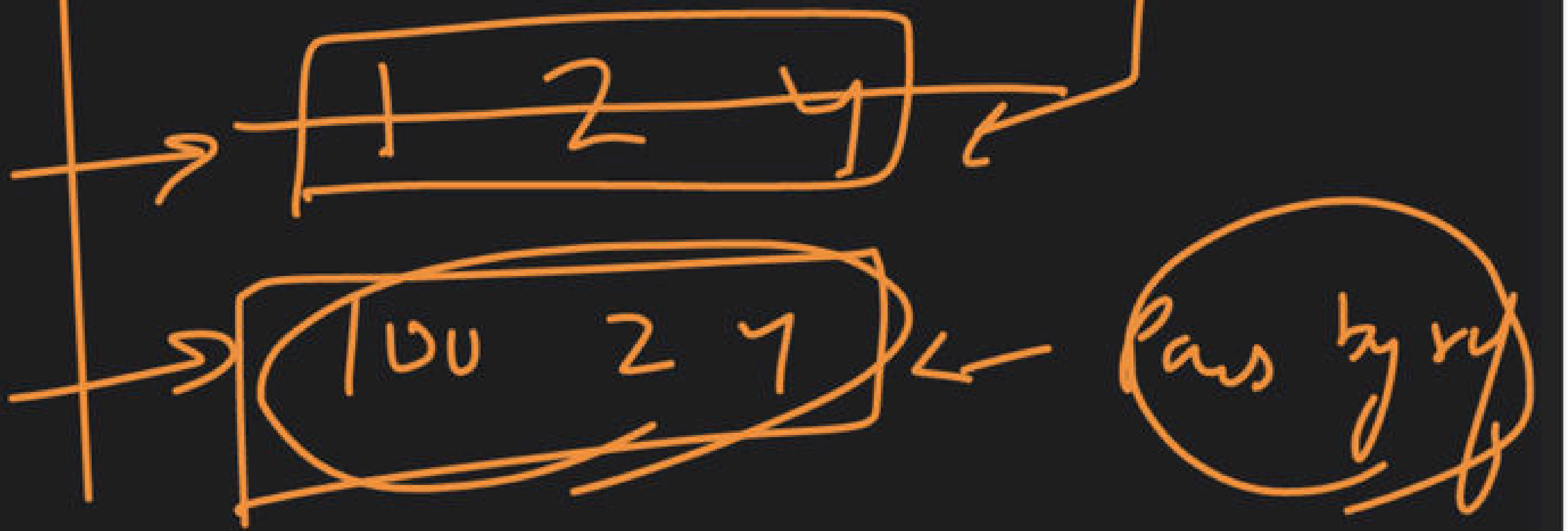


```
int main()
{
  int arr[]= {1, 2, 4}
  int n=3
  solve(arr, n)

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



```
solve(int arr[] int size)
{
  arr[0]=100;

}
```



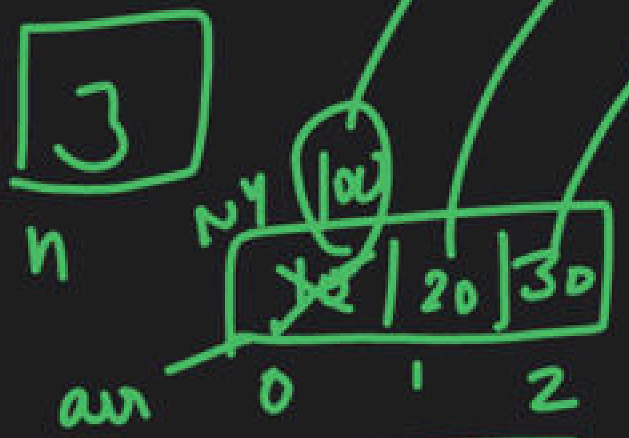


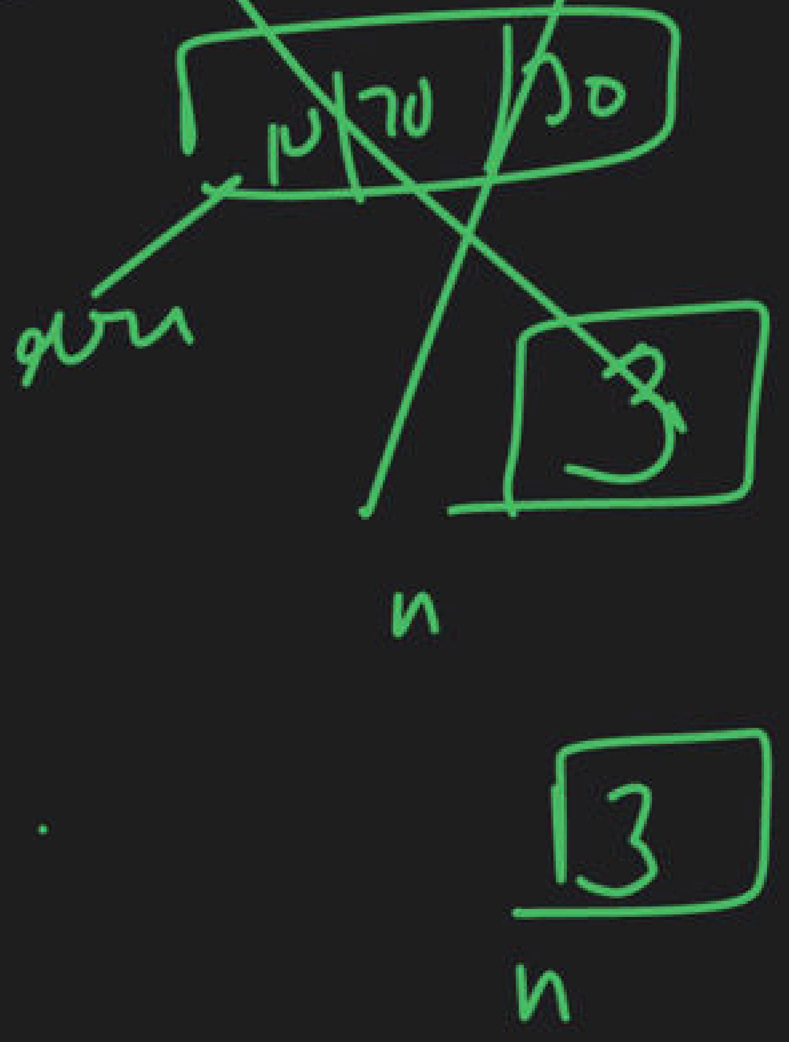
```
int main ( )
{
    int n = 3;
    int arr [ ] = { 10, 20, 30 };

    solve ( arr , n );

    // print array
    for (i = 0; i < n; i++)
    {
        cout << arr[i] << " ";
    }
}
```

```
void solve ( int arr[ ], int n )
{
    arr [ 0 ] = 100;
}
```

copy nhi banegi

copy banegi





→ find Unique Element

↳ i/p → $\cancel{2} \wedge \cancel{10} \wedge \boxed{11} \wedge \cancel{13} \wedge \cancel{10} \wedge \cancel{2} \wedge \cancel{15} \wedge \cancel{13} \wedge \cancel{15}$ → ans

each element → occurs twice except one → find out

U. E → 11

How → how to solve ⇒ XOR → same val → 0

→ diff val → 1

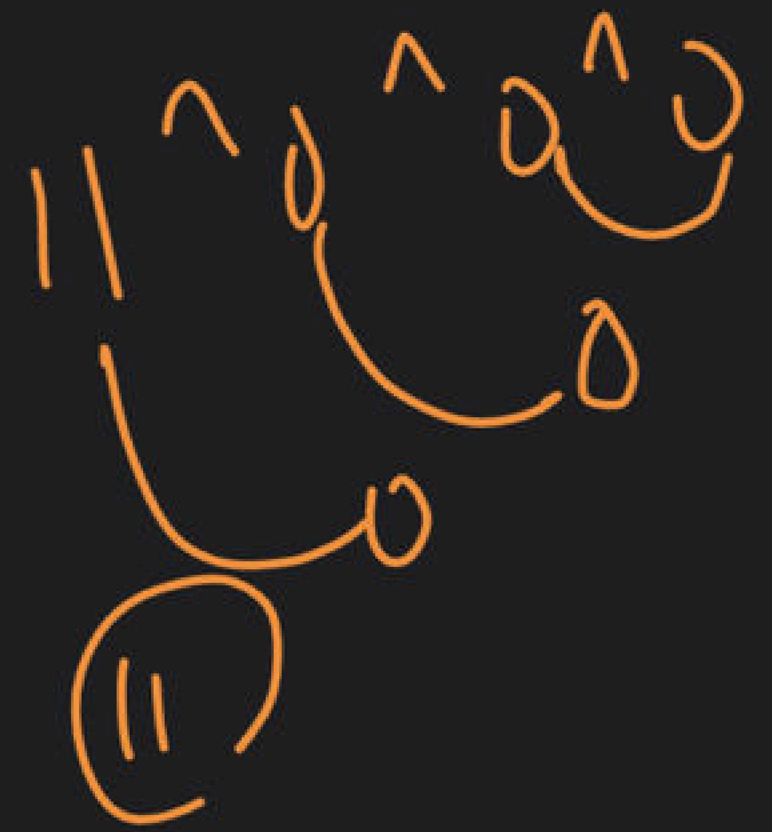
XOR

a	b	o/p
0	$\boxed{0}$	$\to 0$
0	$\boxed{1}$	$\to 1$
1	0	$\to 1$
1	1	$\to 0$

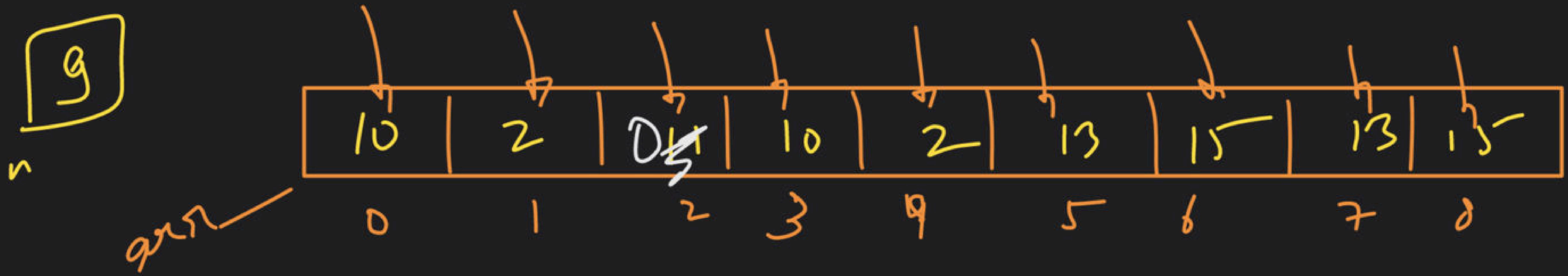
$11 \wedge 0 \wedge 0 \wedge 0$

$0 \wedge a \to a$

$0 \wedge Z \to Z$



$$\begin{tikzcd}
\text{Code} \arrow[r] \arrow[dr, bend right] \arrow[ddr, bend right] & \underline{\underline{\text{Optimal}}} \times \quad \text{But sol}^n \\
& \boxed{\text{Ek } \underline{\underline{\text{solution}}}} \\
& \text{Brute force} \arrow[r] & \boxed{\text{Basic sol}^n}
\end{tikzcd}$$



$$ans = \underline{0} \wedge \cancel{10} \wedge \cancel{2} \wedge \underline{11} \wedge \cancel{10} \wedge \cancel{2} \wedge \cancel{13} \wedge \cancel{15} \wedge \cancel{13} \wedge \cancel{15}$$

$$= 0 \wedge 11$$

$$ans = 11$$

→ i/p → array → [ ] → $\boxed{\{10, 20, 30\}}$

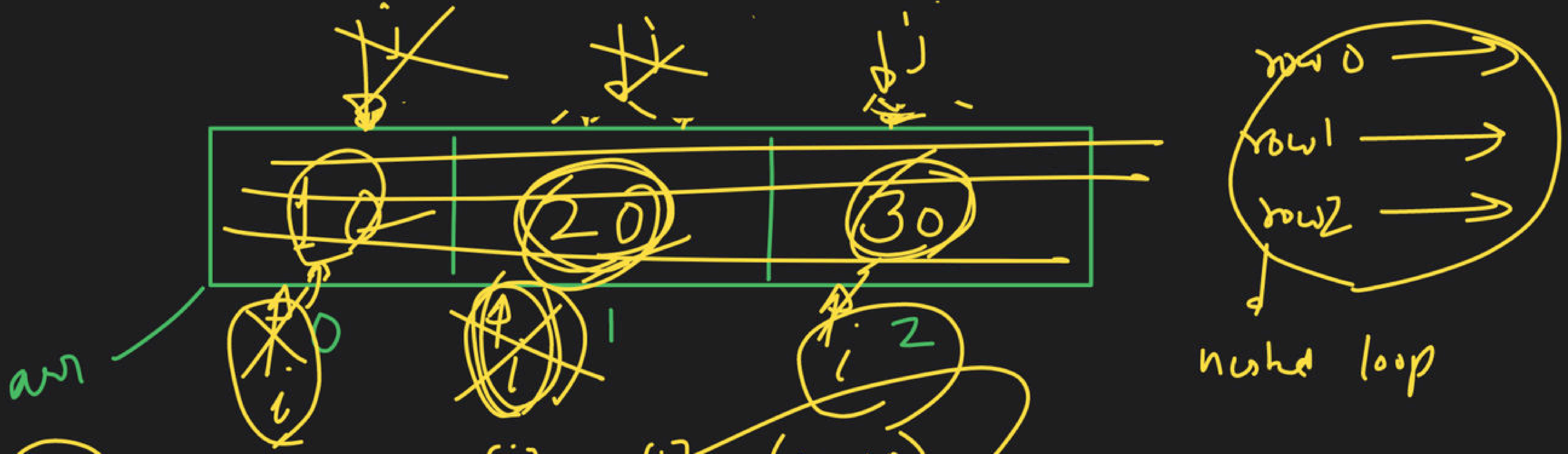
Print all pairs

2 min

(10, 10)	(20, 10)	(30, 10)
(10, 20)	(20, 20)	(30, 20)
(10, 30)	(20, 30)	(30, 30)

2 loops

H/W — Pair Sum / Two Sum



i = 0 , j = 0 → arr[i] , arr[j] → (10, 10)

j = 1 → (10, 20)

j = 2 → (10, 30)

i = 1 , j = 0 → (20, 10)

j = 1 → (20, 20)

j = 2 → (20, 30)

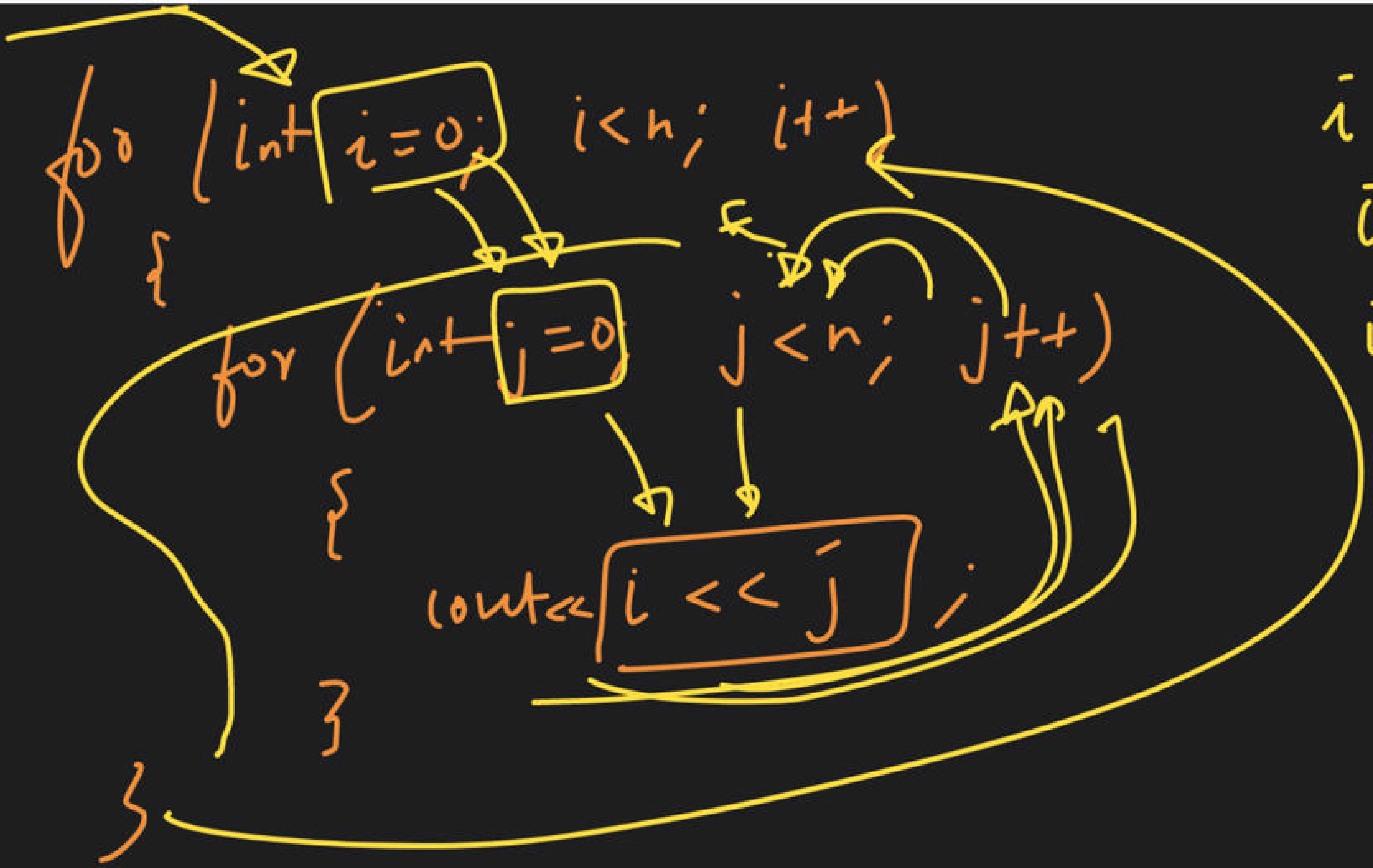
i = 2 , j = 0 (30, 10)

j = 1 (30, 20)

j = 2 (30, 30)

```
for ( i = 0;   i < n/2;   i++)
{
                n              >= 0    j--
    for ( j = ~~0~~ i ;   j < n/2;   j++)
    {
        cout << i << " " << j;
        cout << arr(i) << " " << arr[j];
    }
}
```

88%



```
for (int i=0; i<n; i++)
{
    for (int j=0; j<n; j++)
    {
        cout<< i << j ;
    }
}
```

$i=0;\ j=0; \rightarrow (0,0)$

$i=0\ \ j=1 \rightarrow (0,1)$

$i=0\ \ j=2 \rightarrow (0,2)$

$j=3$

$i=1\ \ j=0 \rightarrow (1,0)$

$j=1 \rightarrow (1,1)$

$j=2 \rightarrow (1,2)$

$i=2\ \ j=0 \rightarrow (2,0)$

$j=1 \rightarrow (2,1)$

$j=2 \rightarrow (2,2)$

$i=3 \rightarrow f \rightarrow \alpha$

i/p → array → [1, 2, 3, 4]

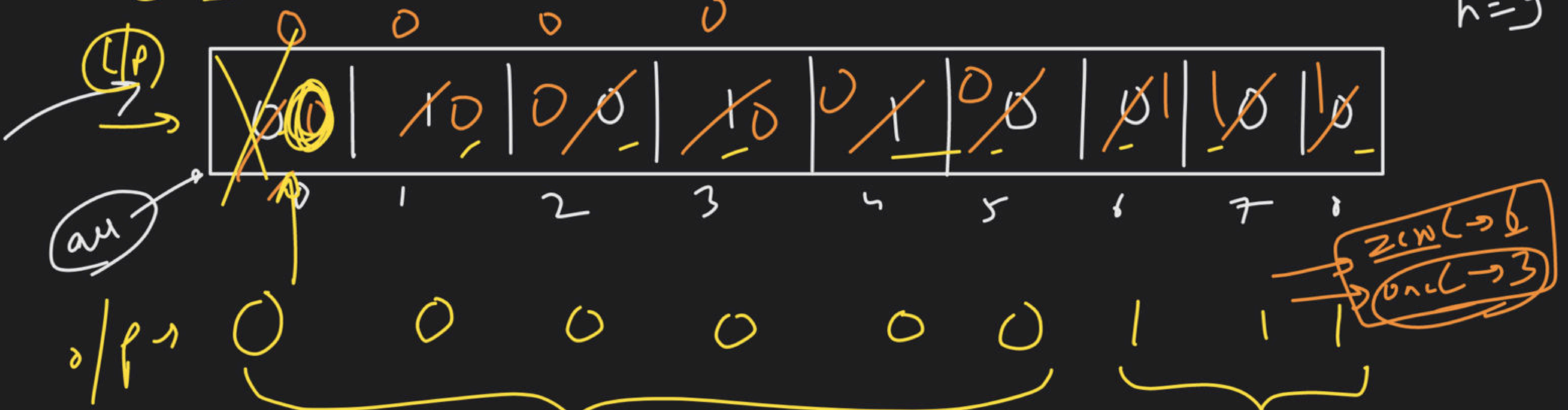
→ print all triplets

```
for (i=0; i<n; i++)
{
    for (j=0; j<n; j++)
    {
        for (k=0; k<n; k++)
        {
            cout << arr(i) << arr[j] << arr[k];
        }
    }
}
```

H/W → Three Sum / Triplet Sum

→ Sort 0's & 1's

n=9

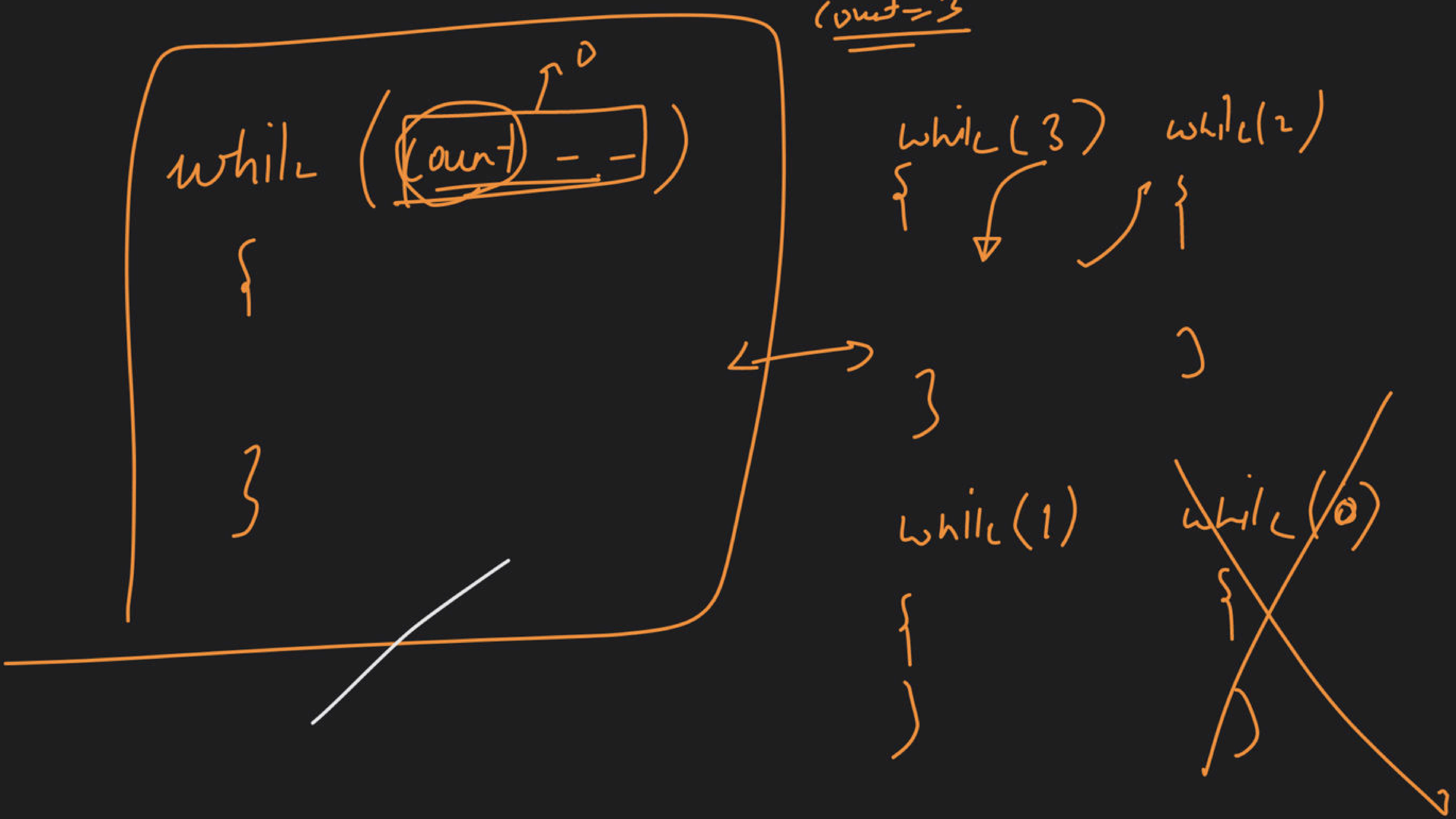


① Counting ⟶ Kyuki dikhata hu

② 2 pointer approach ⟶ H/w

③ sort ( ) ⟶ increment

Logic → Count → 0 & 1
→ place 0
→ place 1



count=3

```
while (count--)
{

}
```

```
while(3)
{

}

while(2)
{

}

while(1)
{

}
```

~~while(0)~~
~~{~~
~~}~~

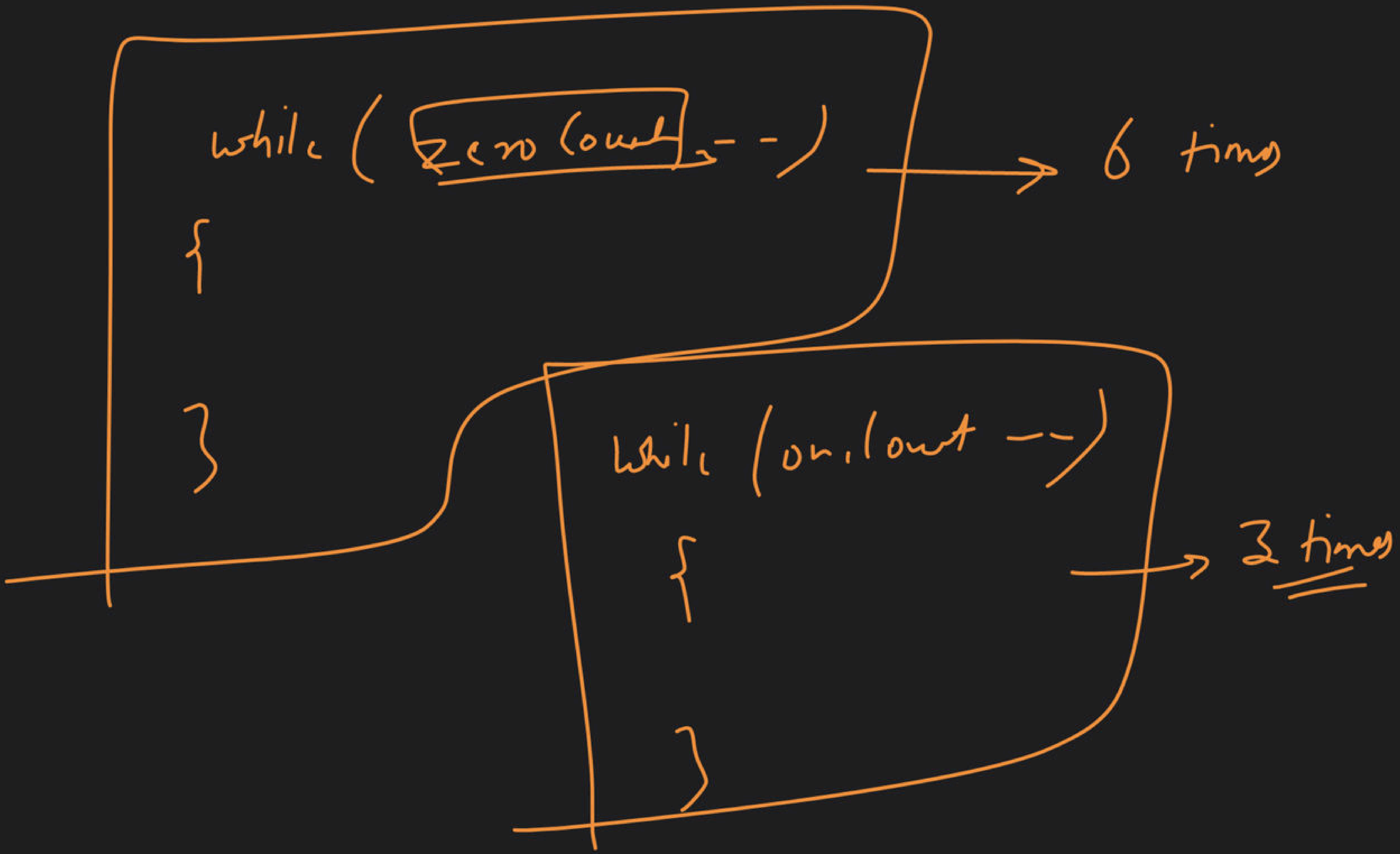
```
while (marks - -)
{

}
```

total kitni baar

or

total kitni iteration

marks



```
while ( zeroCount-- )
{

}
```

```
while (oneCount --)
{

}
```

Step 1

counting

```
int zeroCount = 0;
int oneCount = 0;

for (i=0; i<n; i++)
{
    if (arr[i] = = 0)
    {
        zeroCount ++;
    }
    if (arr(i) — = 1)
    {
        oneCount ++;
    }
}
```

$\boxed{\text{zeroCount} = 1}$

oneCount = 3

Step B

```
int index = 0

while (zeroCount --)
{

    arr[index] = 0;

    index++;

}
```

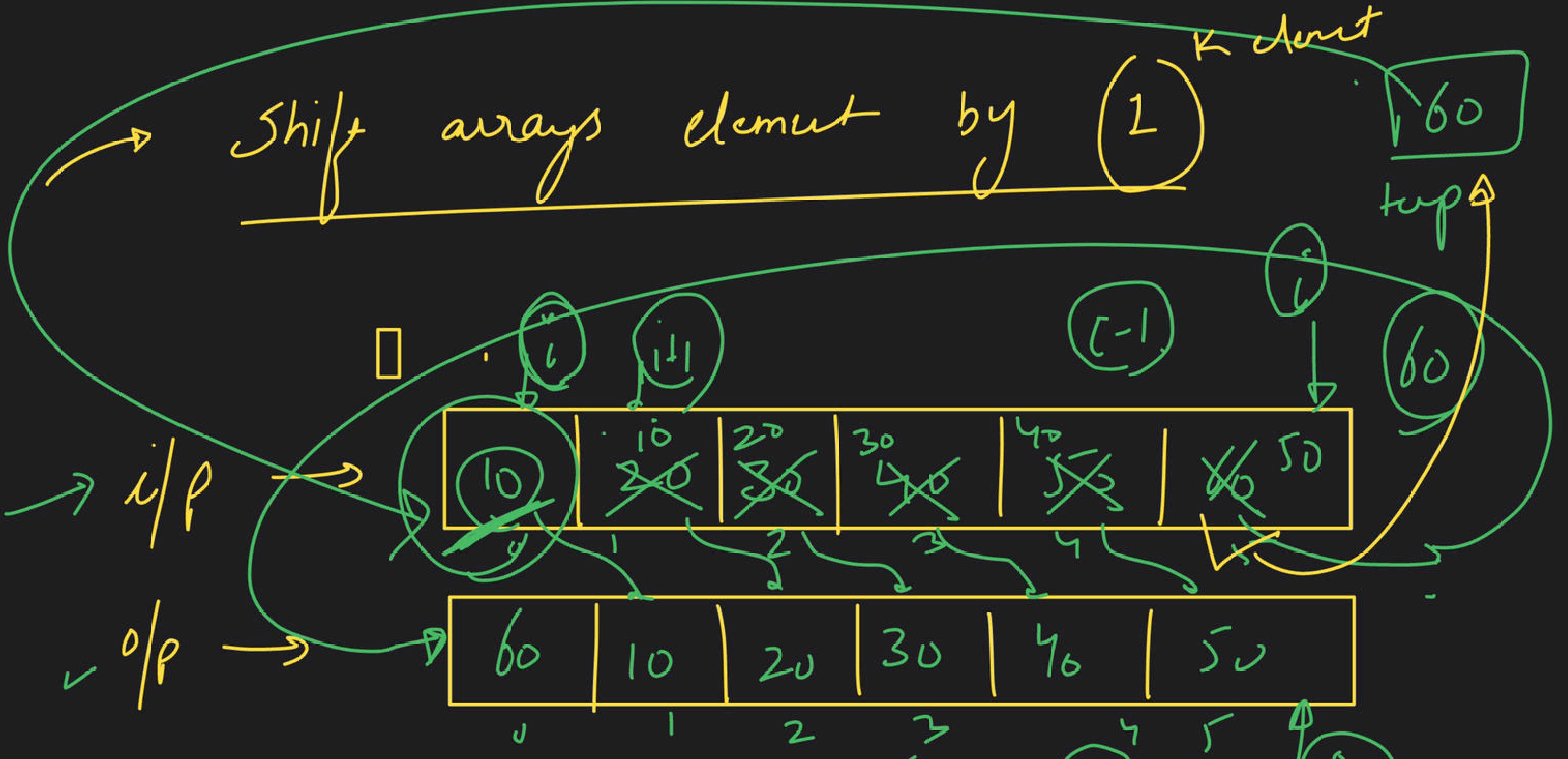
6 times

Step 3

```
while( onecount --)
{

    arr[index] = 1;
    index ++;

}
```

Shift arrays element by 1

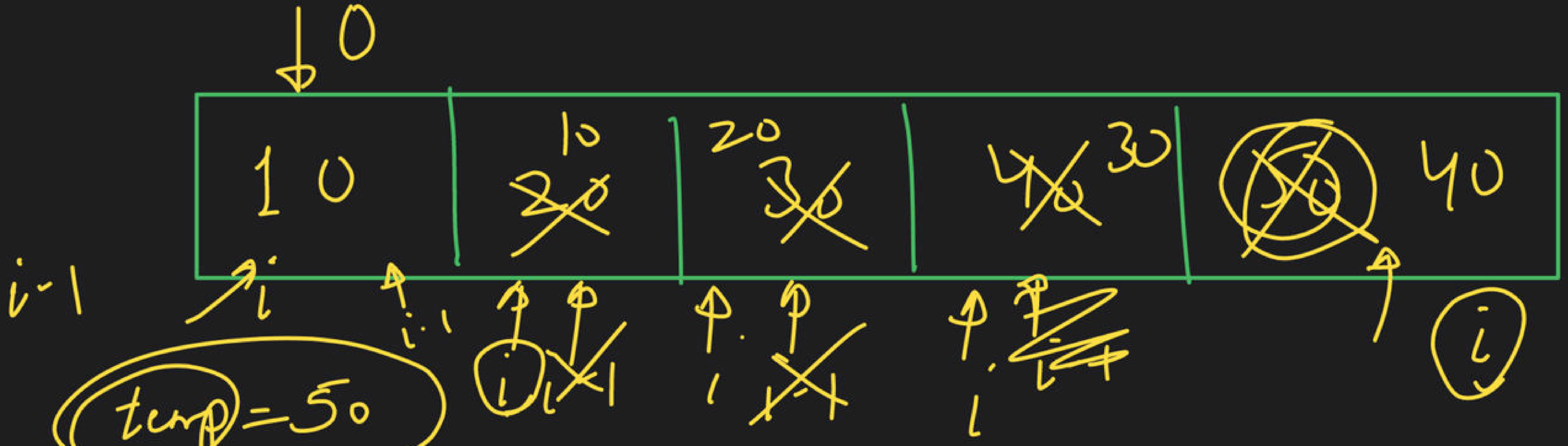
k element



~~arr[i+1]~~ = arr · (i-1)

i

arr[i] = arr(i-1)



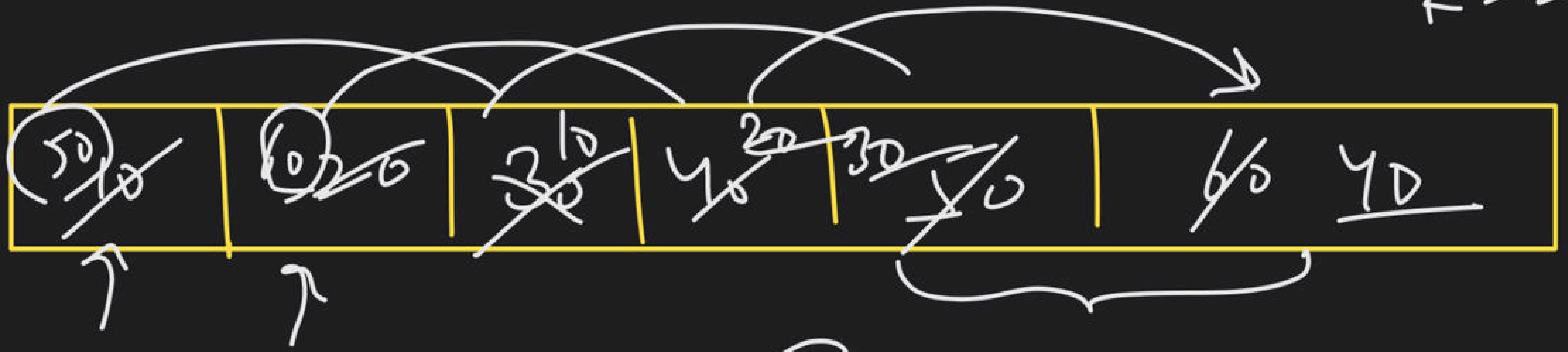
(A) temp = 50

(B) i = n-1 → ~~i=0~~ i=1

→ arr[i] = arr[i-]

(C) arr[0] = temp

K = 2



① temp[] = {50, 60}

② shift

③ copy temp