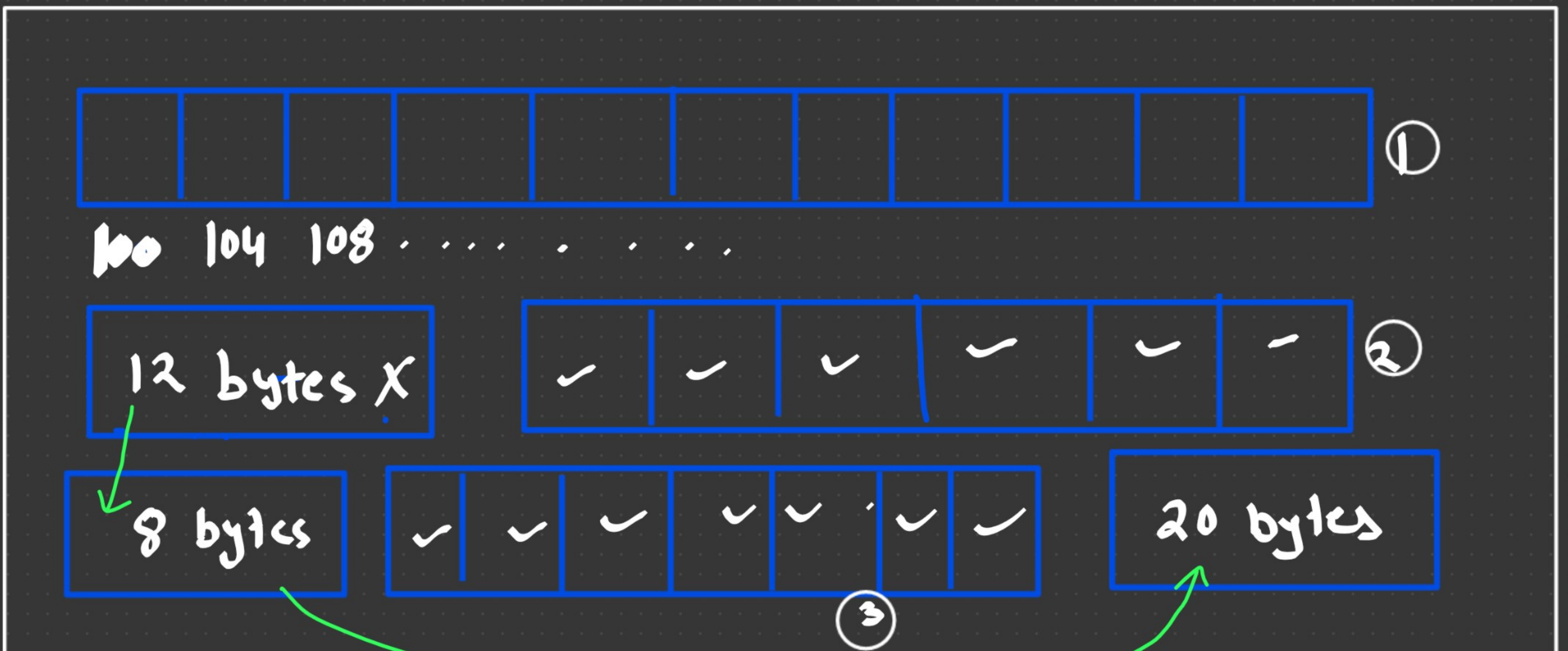


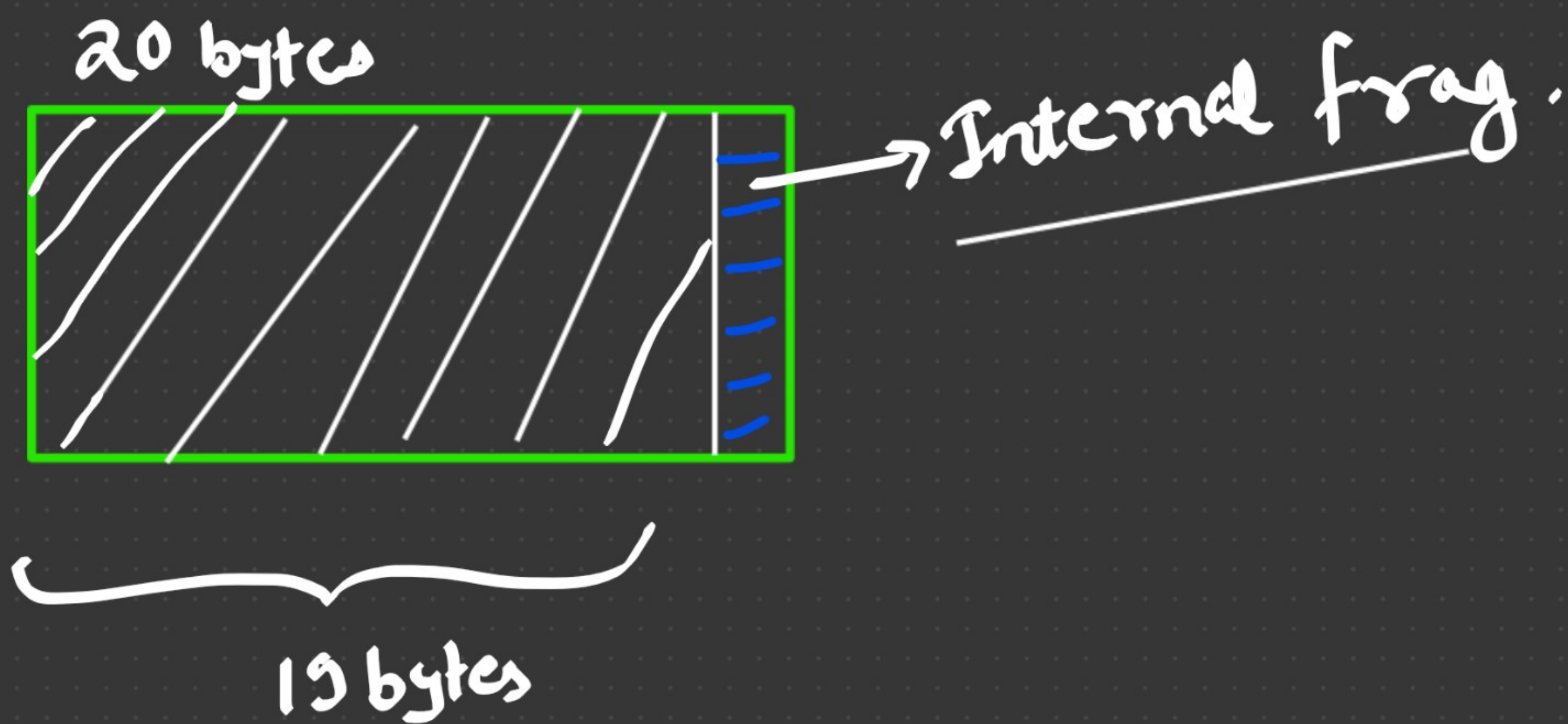
Linked List

RAM

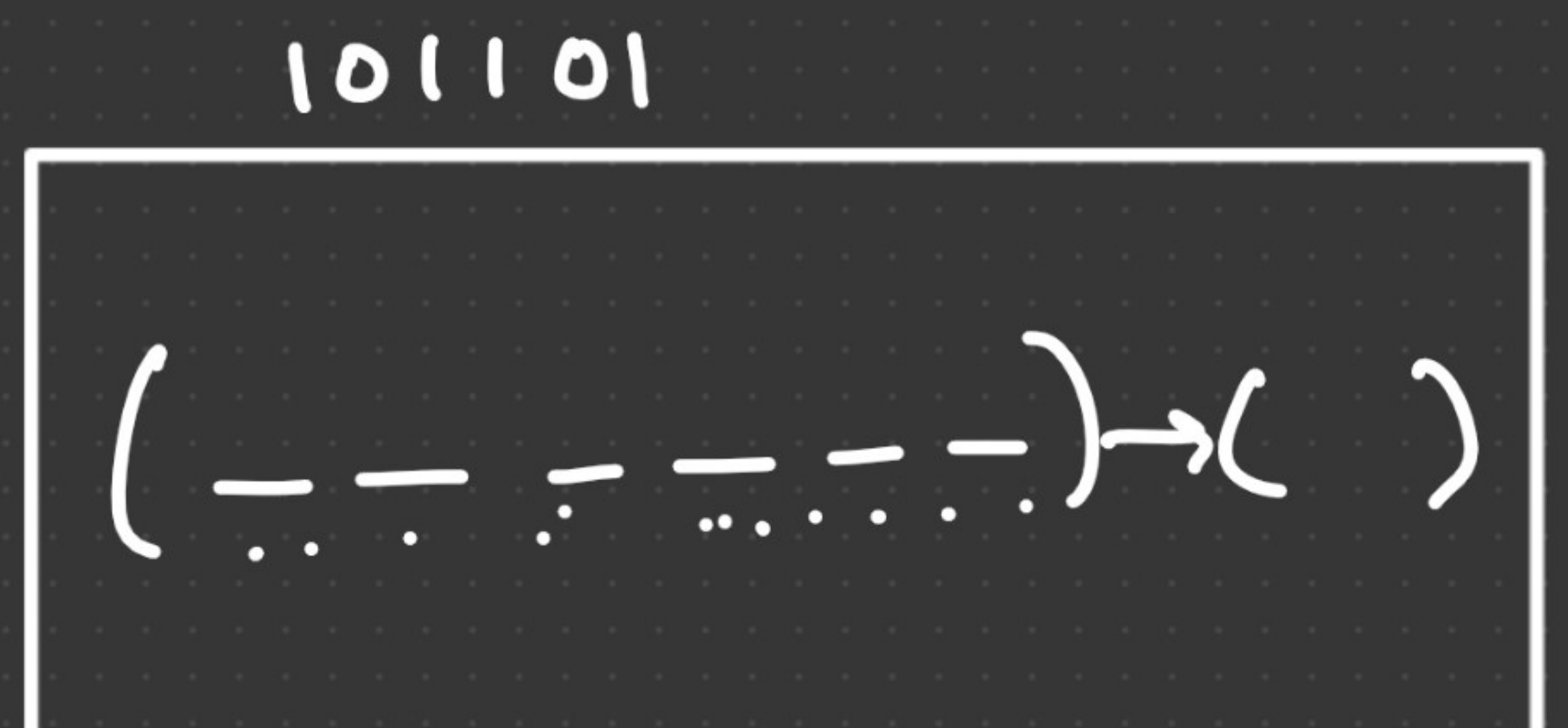
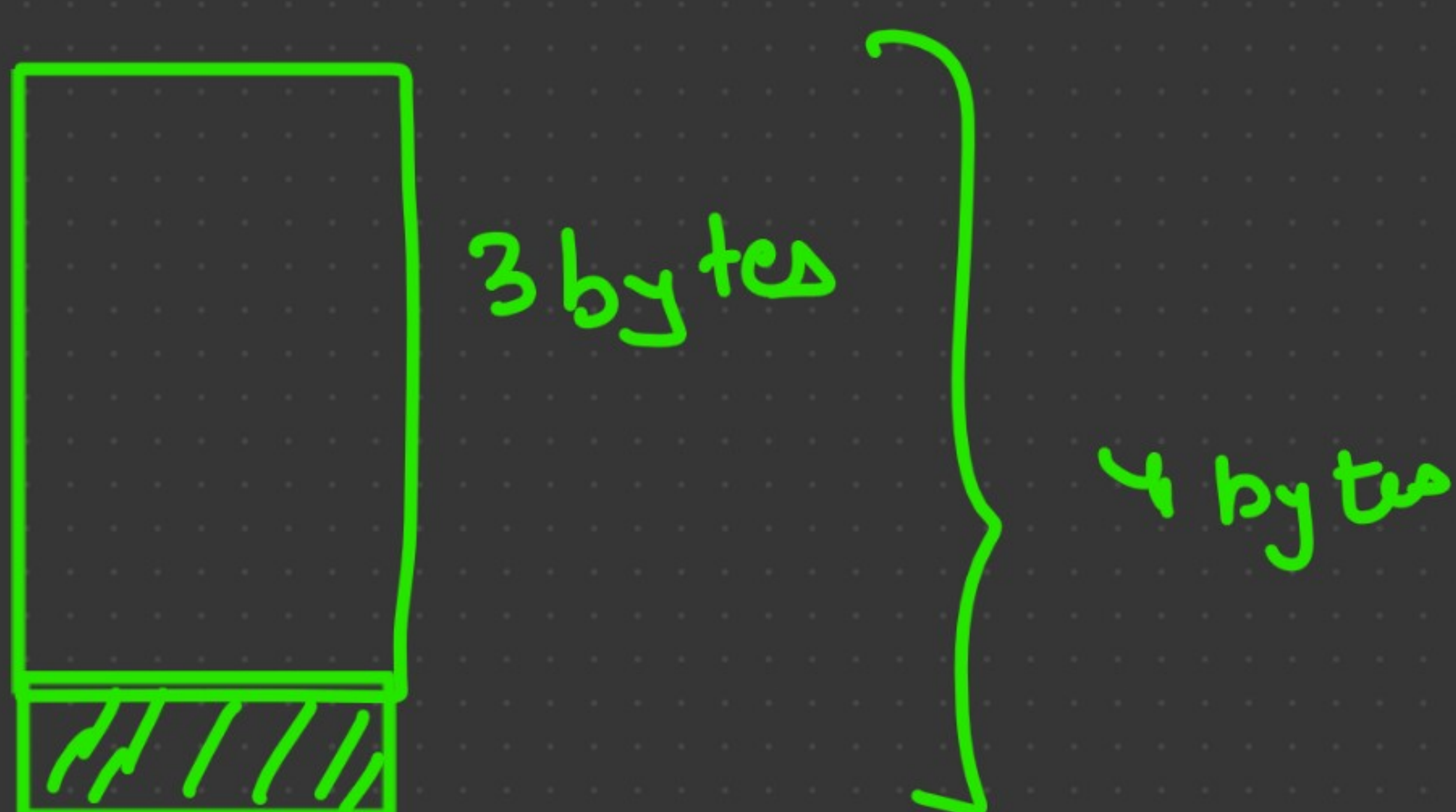


`int a[10];` \Rightarrow 40 bytes $\Rightarrow 20 + 12 + 8$

Fragmentation $\begin{cases} \text{Internal Frag. (X)} \\ \text{External frag. } \checkmark \end{cases}$



Memory Allocation :- $2^0 \text{ bytes}, 2^1, 2^2, 2^3, \dots, 2^n$
 $1, 2, 4, 8, 16, \dots$

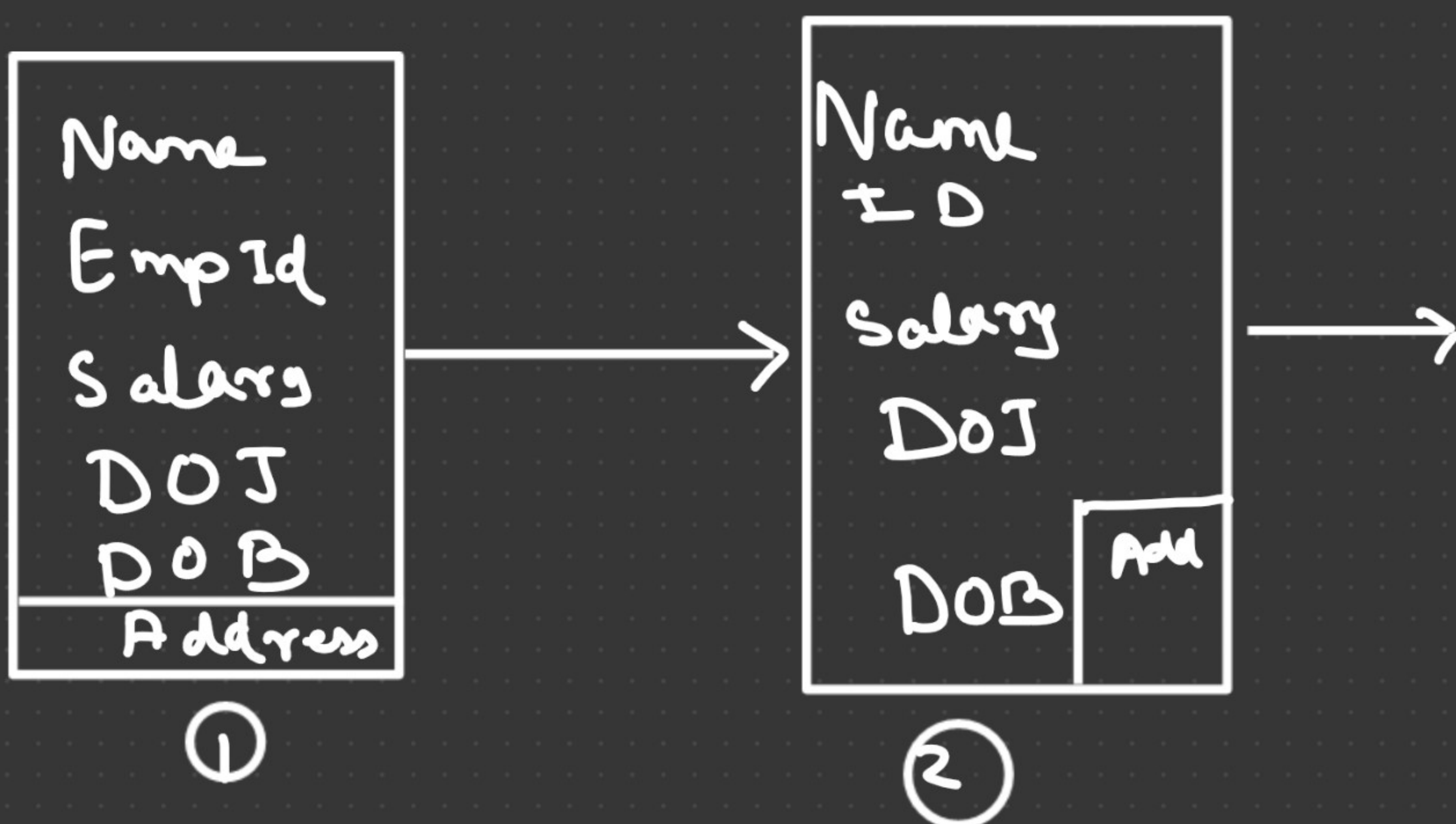
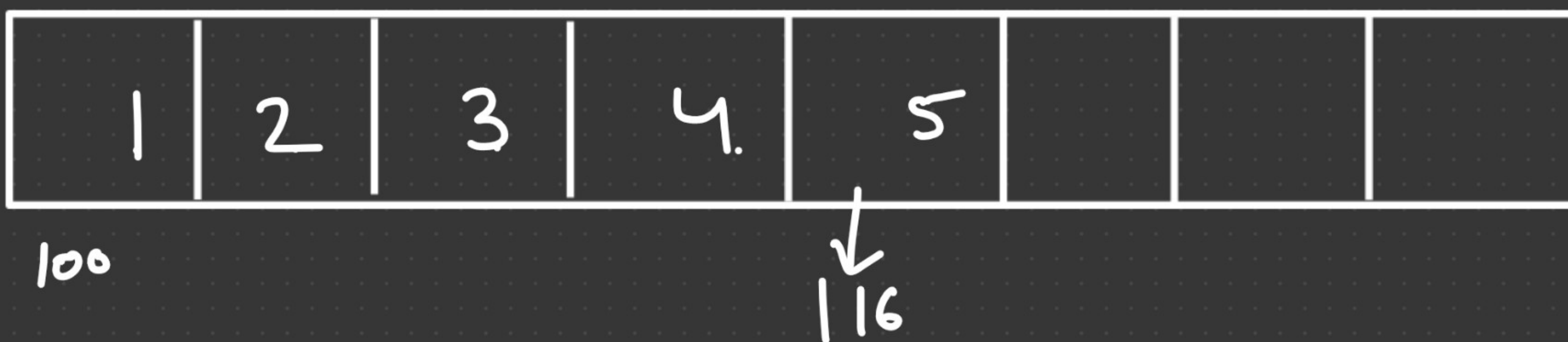
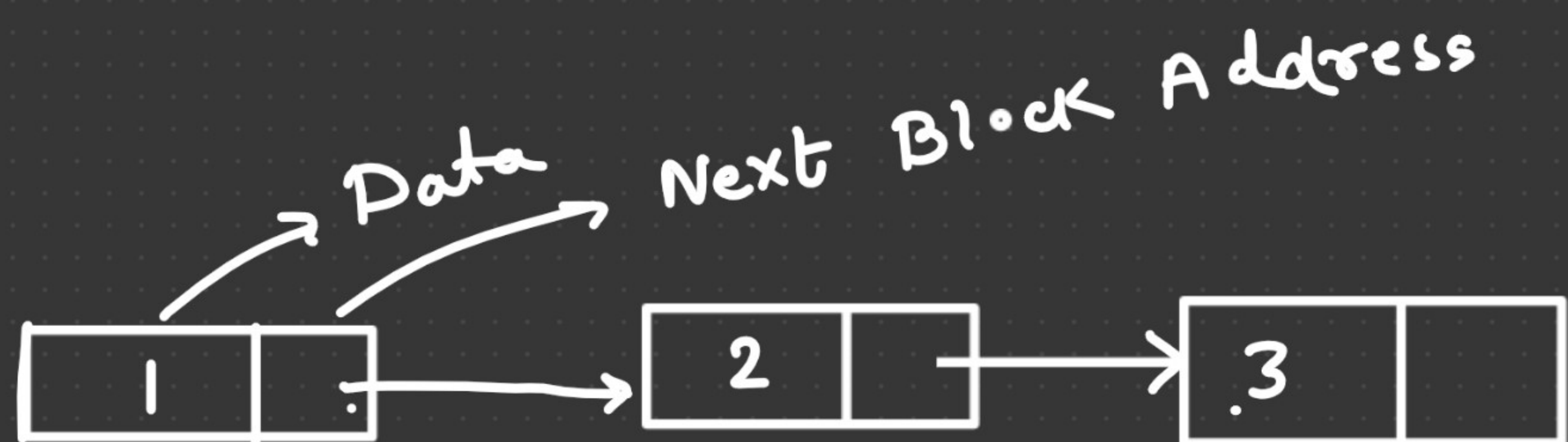


2 MP, 8 MP, 16 MP.

50 MP X

2 MP
48 MP

16 MP ✓
32 MP ✓
2 MP ✓



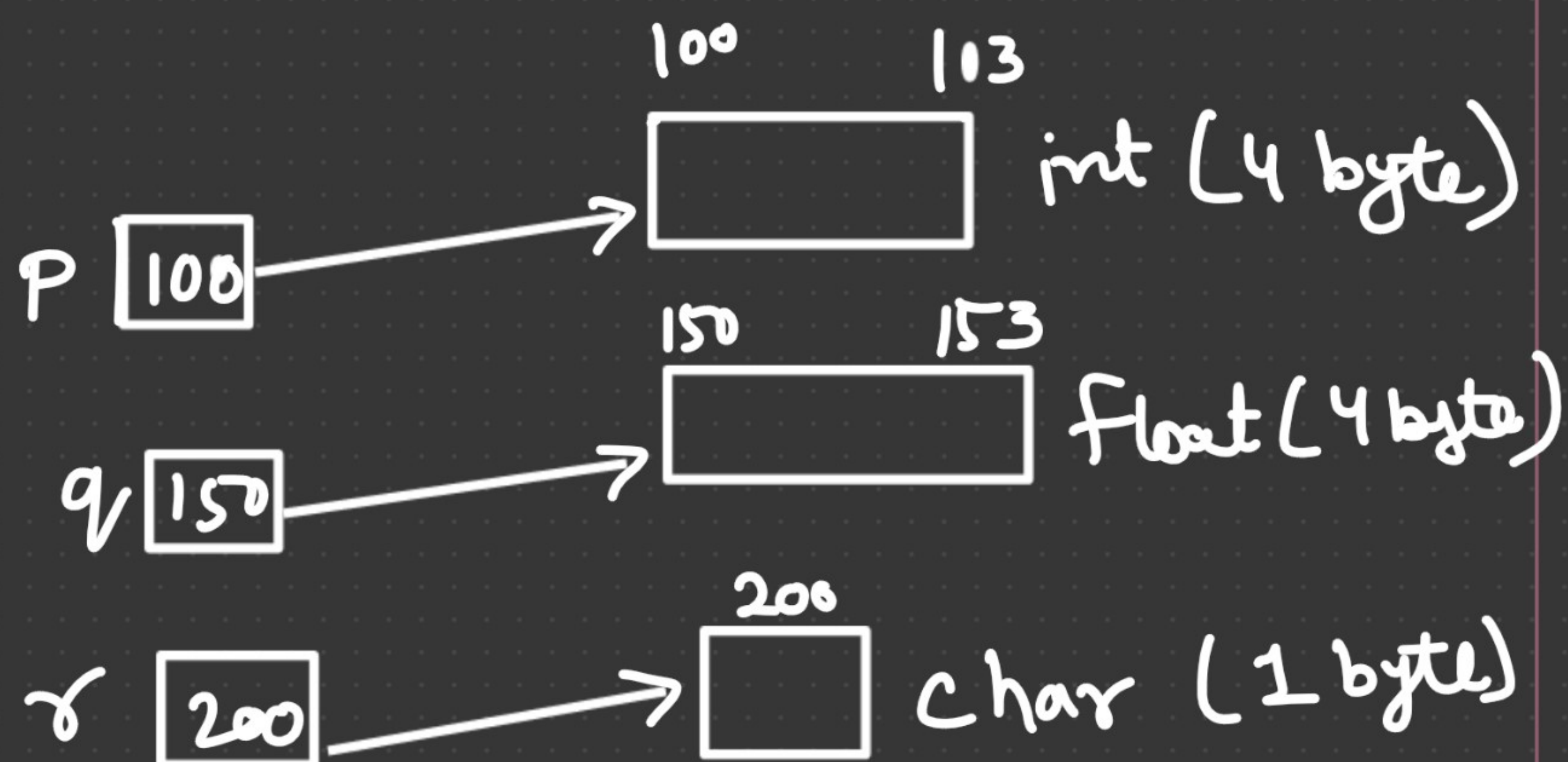
Userdefined DataType

Struct class



```

int *p;
float *q;
char *r;
  
```



```

class node
  {
  
```

```

    int value;
    node *next;
    node()
    {
        next = NULL;
    }
  };
  
```



- 1) Create node
- 2) Insert new node
- 3) Print.

```

class linkedlist
  {
  
```

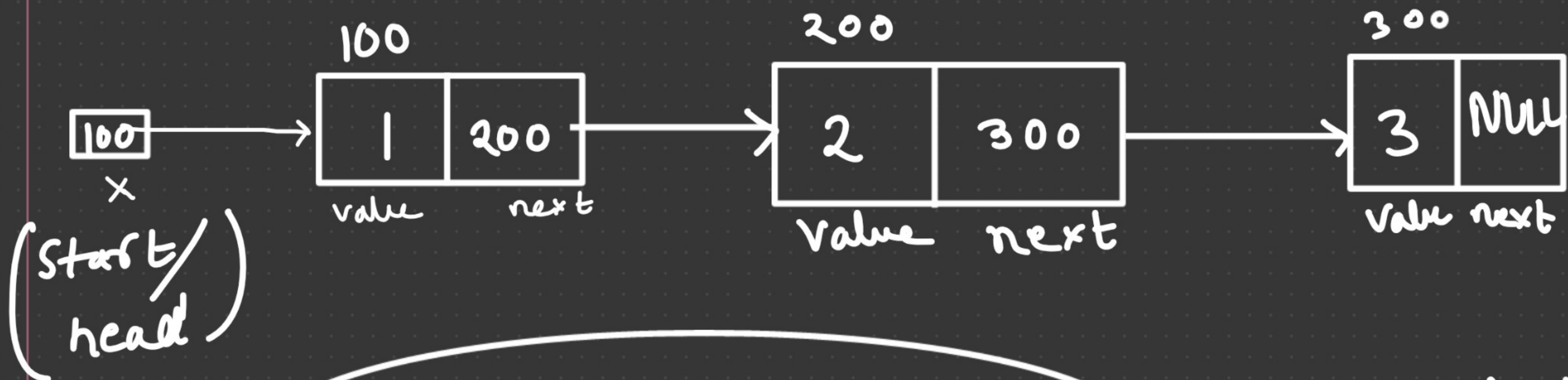
```

    public:
        create node();
        Insert (value);
        print ();
  
```

```

};
  
```


`node *x = new node();` → 1) Create node



`node *y = new node();` → new Node created

`x → value = 1;`

`y → value = 2;`

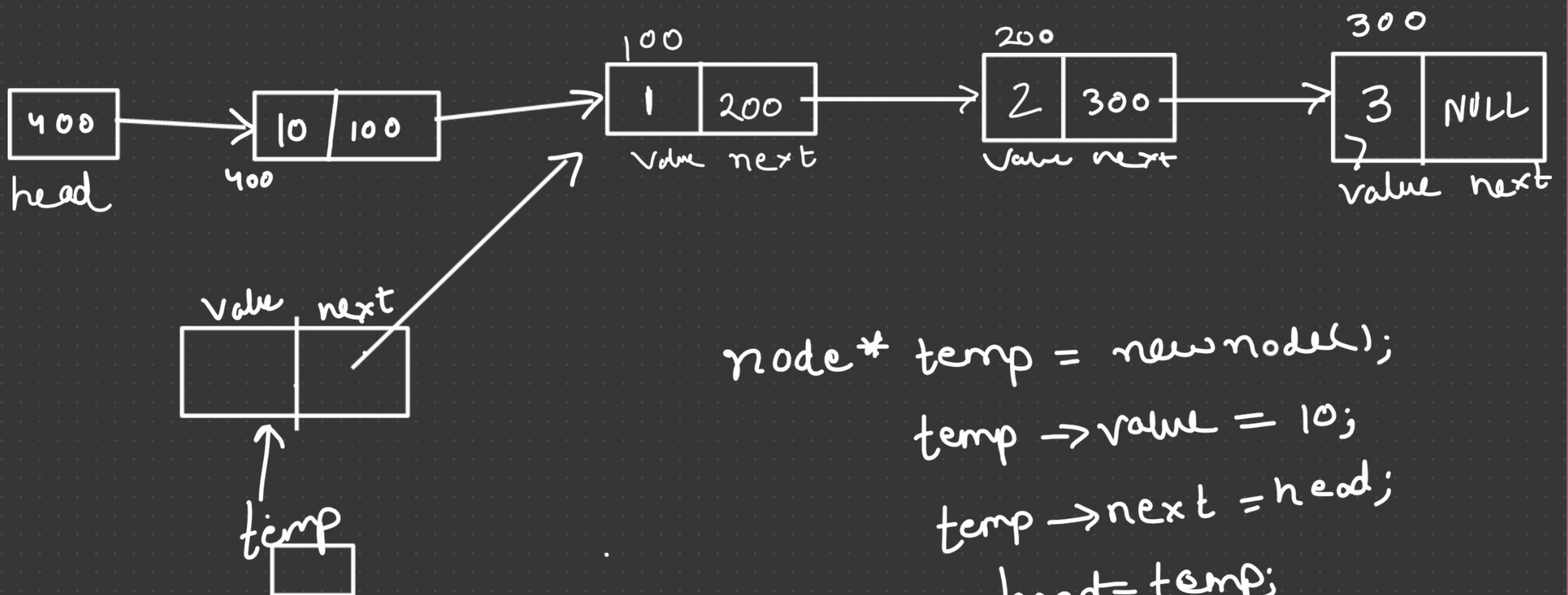
`x → next = y;`

repeat

`y → value = 3;`

`x → next → next = y;`

→ "don't"



`node *temp = new node();`

`temp → value = 10;`

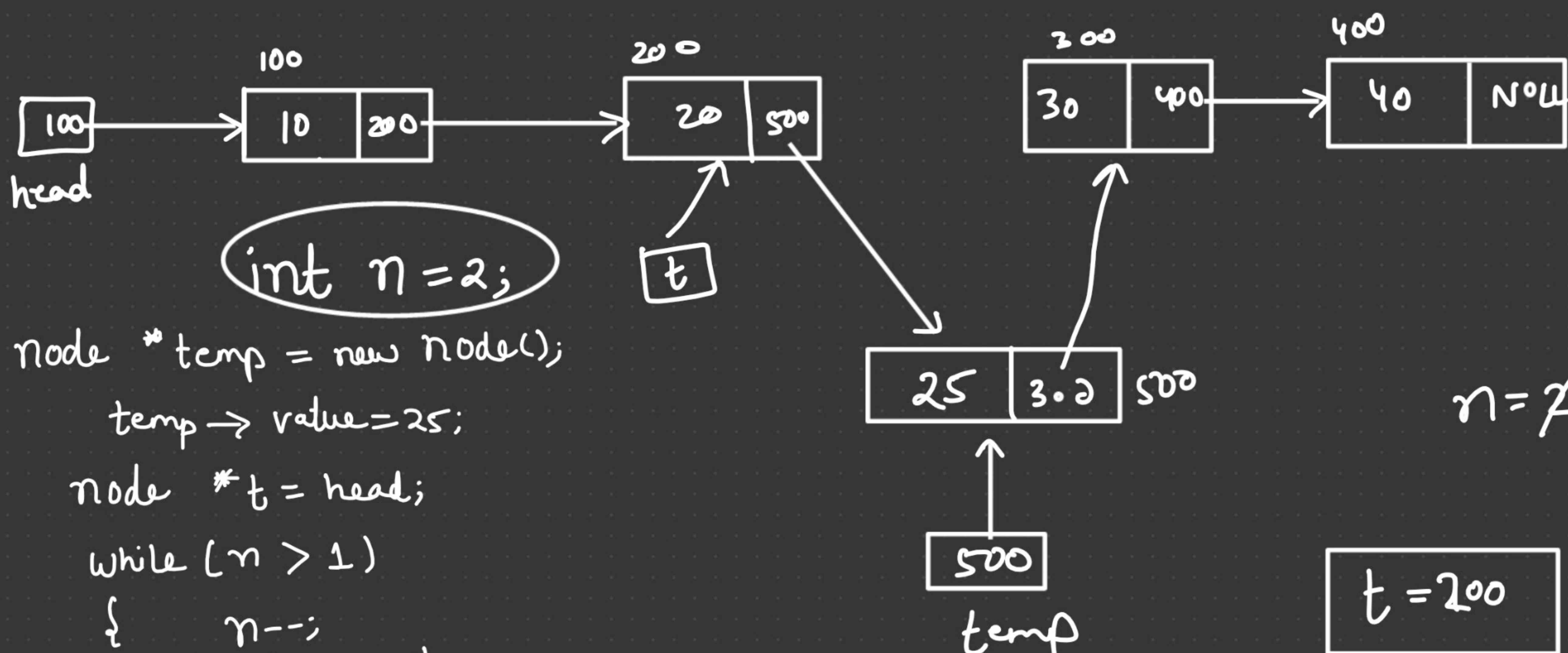
`temp → next = head;`

`head = temp;`



`t = 100`

`(t → next) = 200`



```

node *temp = new Node();
temp->value = 25;
node *t = head;
while (n > 1)
{
    n--;
    t = t->next;
}
temp->next = t->next;
t->next = temp;

```

Doubly Linked List

