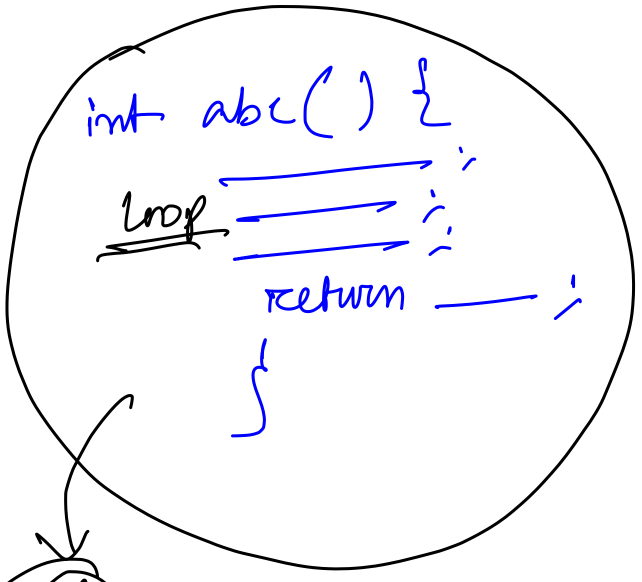
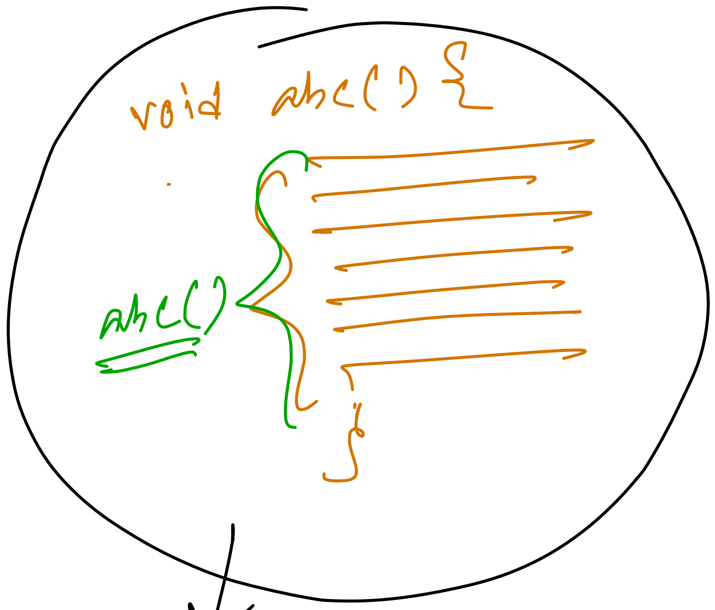
Recursion

Two way of writting Codes.

$$\begin{tikzcd} & \text{Iterative Method} \\ \text{Two way of writting Codes.} \arrow[ur] \arrow[dr] & \\ & \text{Recursive Method} \end{tikzcd}$$





Loop → Iterative function

A function that calls itself is called a recursive function.

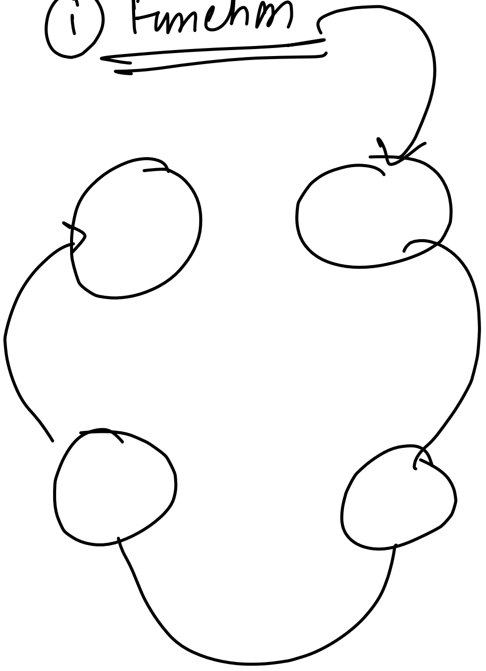
Terminating condition of a recursive function

int

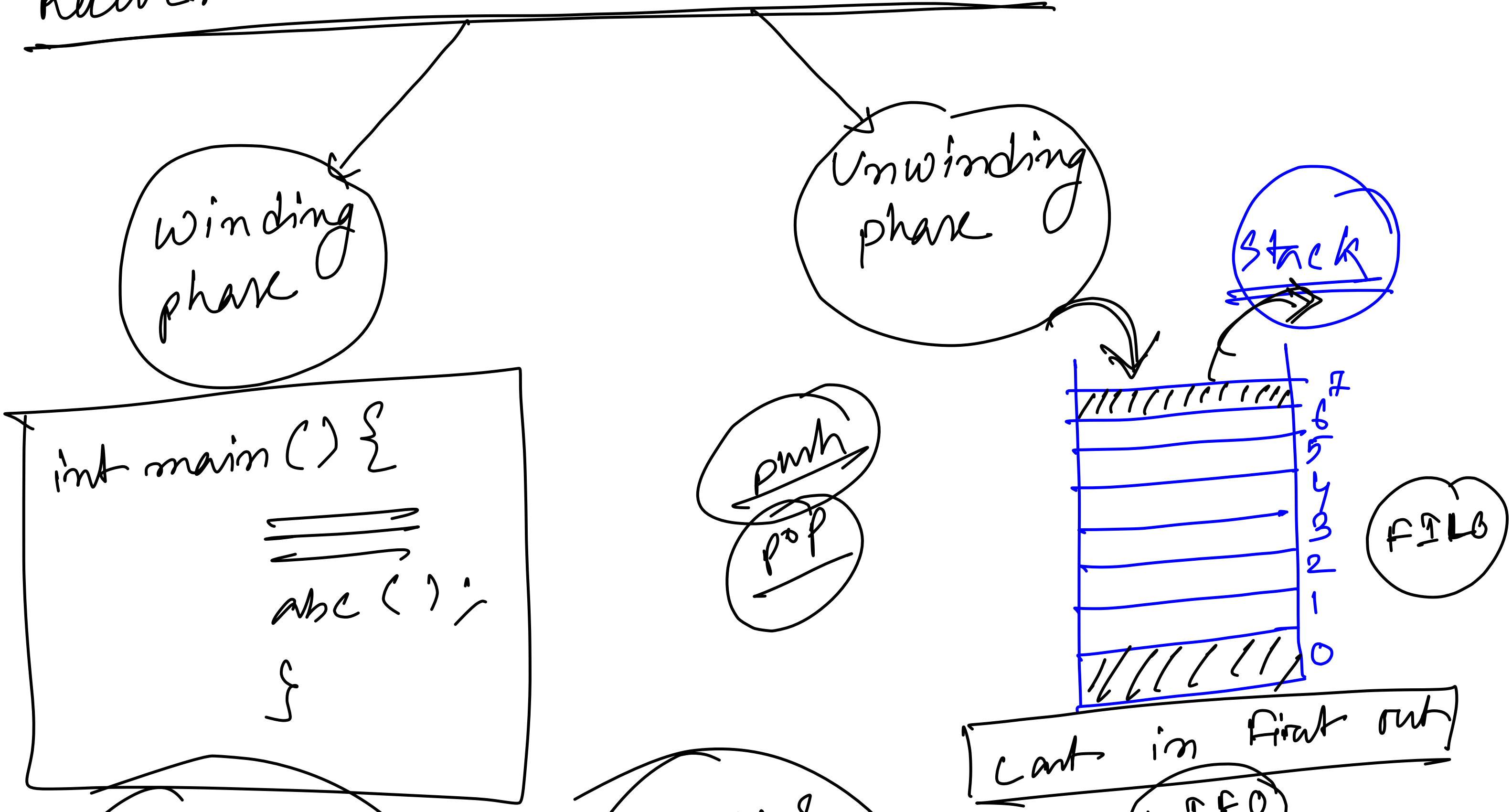
float

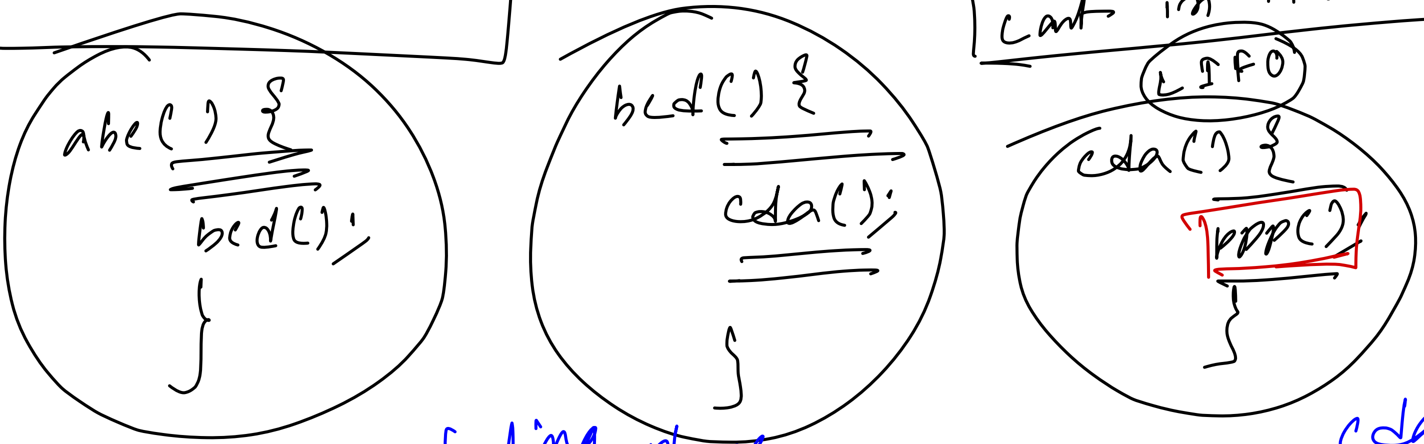
```
void abc(----) {
    if(________)
        return;
    ________
    ________
abc() ________
    ________
}
```

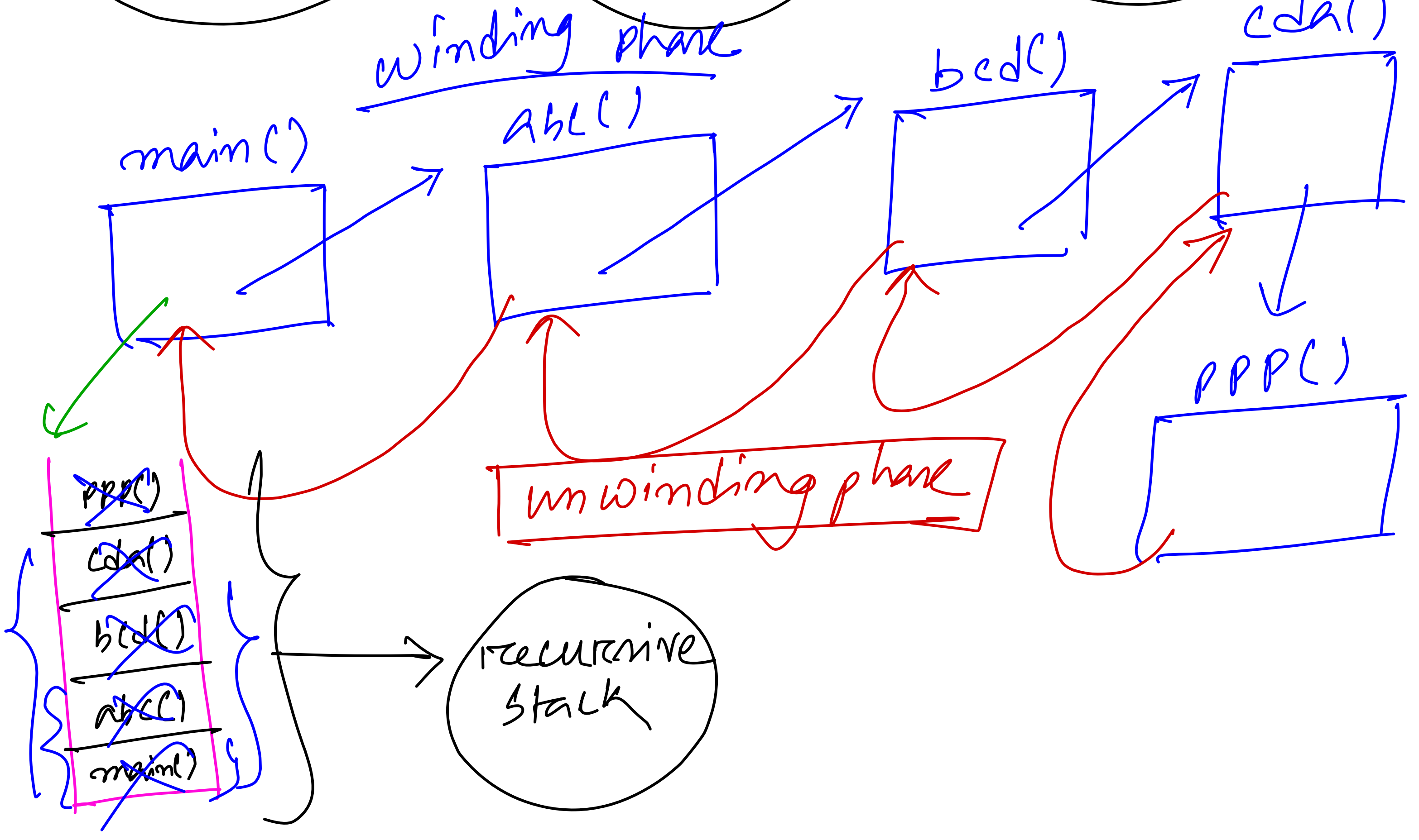
① function



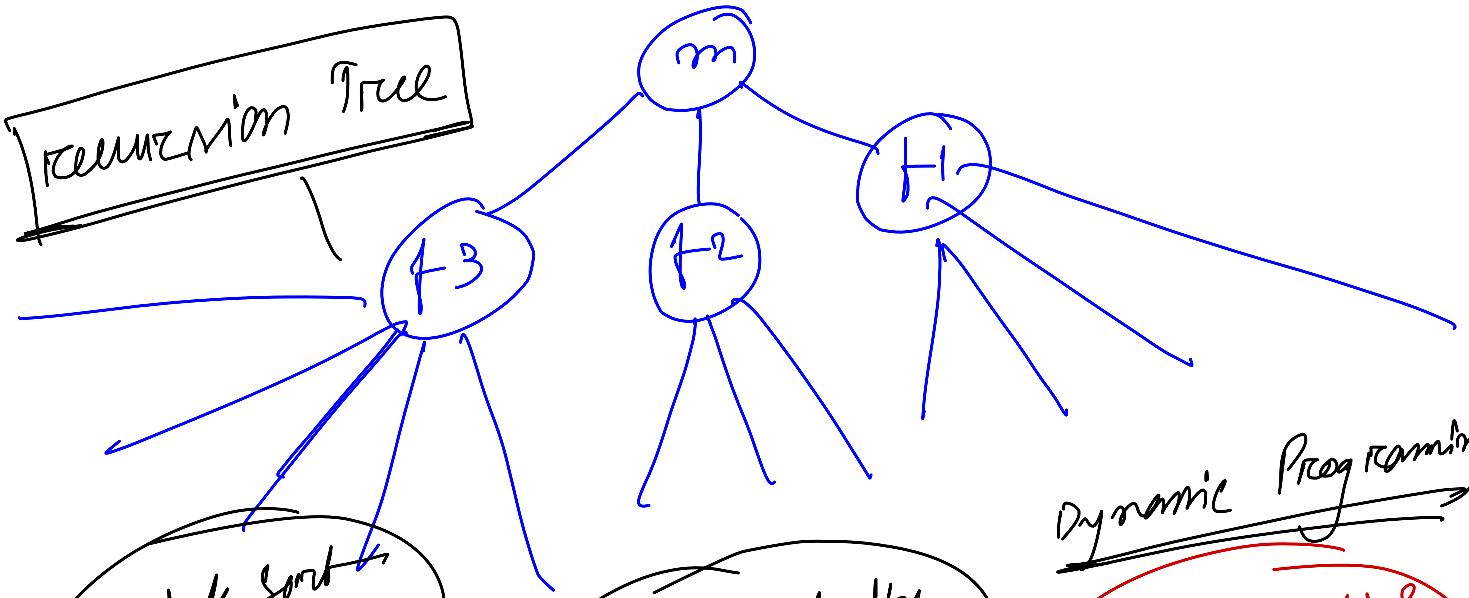
Recursive functions have two phases.







Recursion Tree



quick Sort
merge Sort

Time Complexity
Space ''

Dynamic Programming

```
int main() {
    abc(10);
    return 0;
}
```

Iterative way

Condition

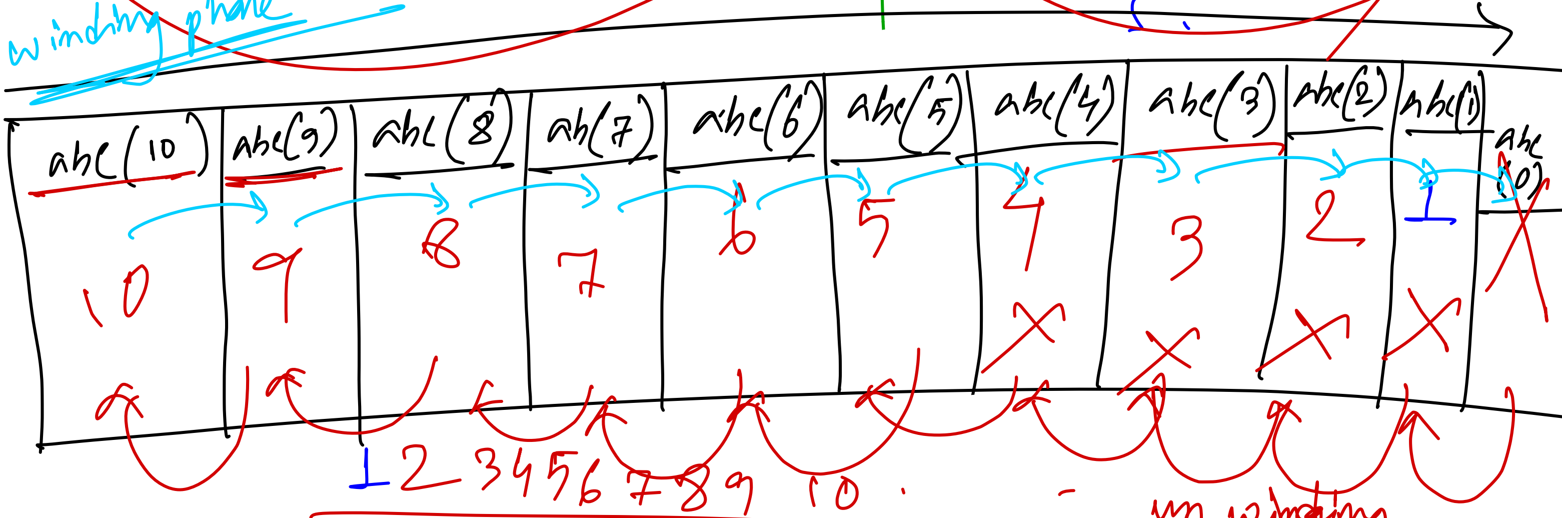
```
void abc(int n) {
    for(int i=1; i<=n; i++) {
        printf("%d\t", i);
    }
}
```

Recursive way

```
void abc(int n) {
    if(n==0)
        return;
    abc(n-1);
    printf("%d\t", n);
}
```

Inc/Dec

winding phase



1 2 3 4 5 6 7 8 9 10 .

un winding phase

Sum of N natural Numbers using Recursion

5

```
int sum (int n, int s) {   ~~int s = 0;~~
    if (n==0) return s;
    s += n;
    sum (n-1, s)
}.
```

```
int main() {
    s = 0
    n
    pf( sum(n, s));
}
```

s = 5+4+3 +2 +1 → 15

```
int s;
int abc (n) {
    if (n==0) return s;
    s+= n;
    abc (n-1);
}.
```

```
main( ) {
    abc (5)
}
```

$$\text{Sum}(5) = 5 + \underbrace{4 + 3 + 2 + 1}_{\text{sum}(4)}$$

$$\text{Sum}(n) = \begin{cases} n + \text{Sum}(n-1) & \rightarrow n > 0 \\ 0 ; & \longrightarrow n == 0 \end{cases}$$

```
int sum( int n) {
    if(n= =0)   return 0;
    return  n + sum(n-1);
}.
```