RECURSION
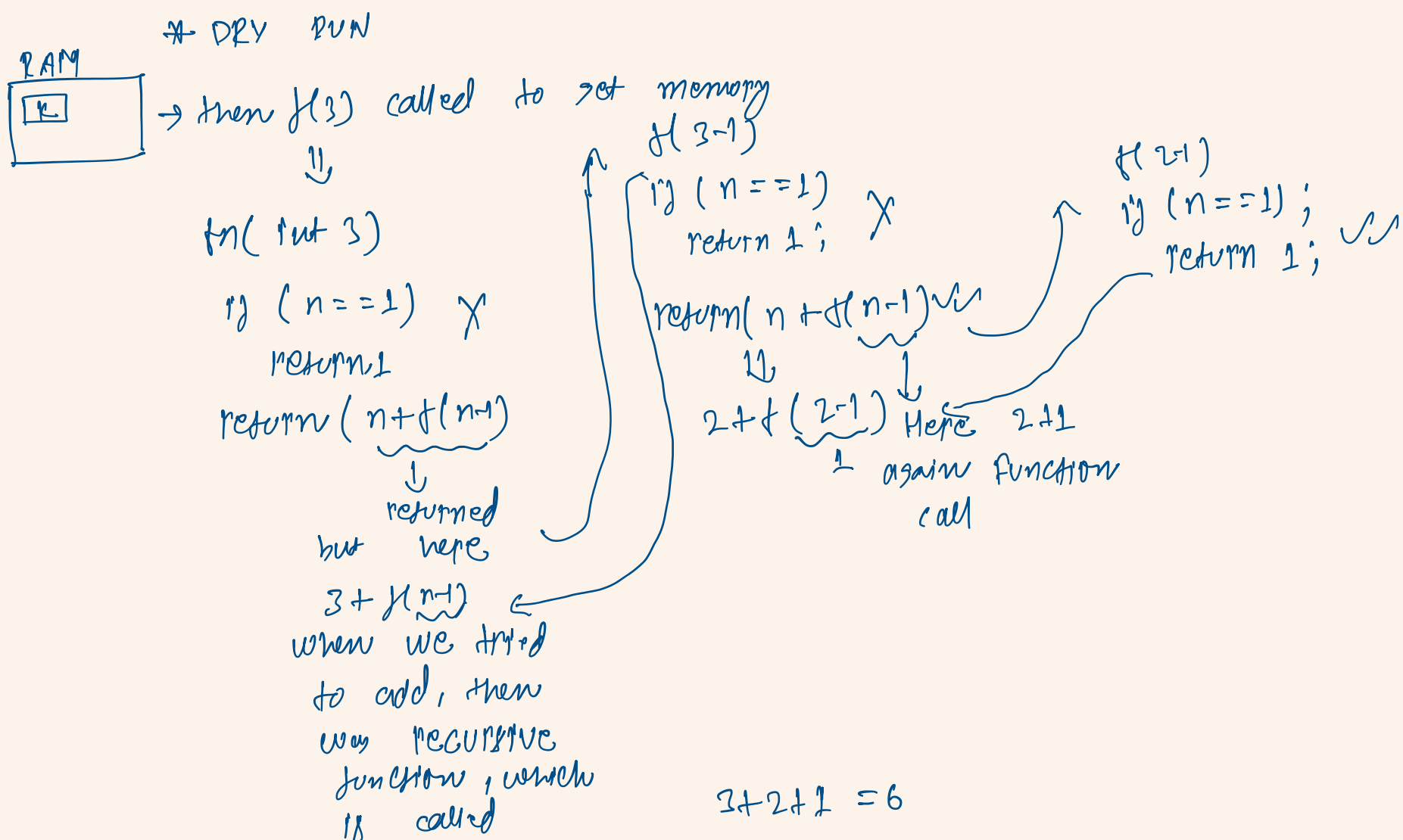
What is Recursion ?

⇒ Function calling itself is called Recursion.

⇒ A recursive method solves a problem by calling a copy of itself to work on a smaller problem.

⇒ It is important to ensure that recursion terminates.

⇒ Each time the function is calling itself with a simpler version of the original problem

⇒ Recursive code are generally shorter and easier to write than iterative code.

⇒ Solution of some problems are easier to formulate recursively.

RECURSION TREE

```
int main ()
{
    int k;
    k = f(3);
    printf ("%d", k);
    return 1;
}

int ft ( int n)
{
    if (n == 1)
        return 1;
    return (n + f(n-1)) → recursion (recursive call)
}
```

DRY RUN

RAM

| k |

→ then f(3) called to set memory

⇓

fn(int 3)
  if (n == 1) ✗
    return 1
  return (n + f(n-1)

⇓
returned but here

3 + f(n-1) ⟵
when we tried to add, then was recursive function, which is called

f(3-1)
  if (n == 1) ✗
    return 1;
  return(n + f(n-1)

⇓

2 + f(2-1) Here 2+1
  ↳ again function call

f(2-1)
  if (n == 1);
    return 1; ✓

3 + 2 + 1 = 6

HOW TO APPROACH RECURSION PROBLEM

① write the thing which you want and assume that the function will give the value.

② het a termination point, where function terminates.

③ Recall the function but in simpler form or reduced form.