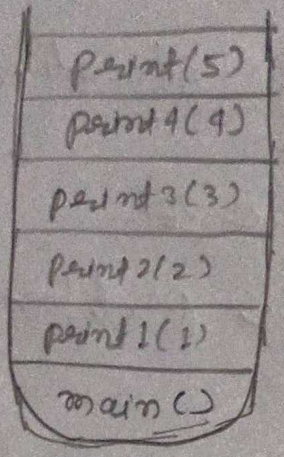
Recursion

How function calls work in languages.

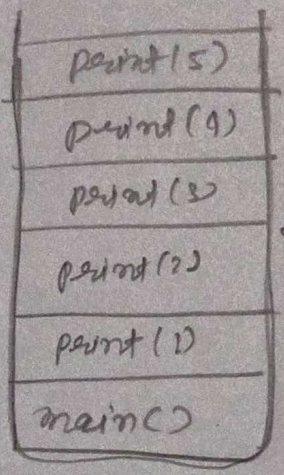


Imp. points

* While the func. is not finished executing it will remain in stack.
* When a function finishes executing it is removed from the stak and the control is restored to where the function was called.

Output
1
2
3
4
5

35:00



every func. call will take some memory.

```
static void print (int n) {
    if (n == 5) {
        return;
    }
    cout (n);
    printf (print(n+1));
}
```

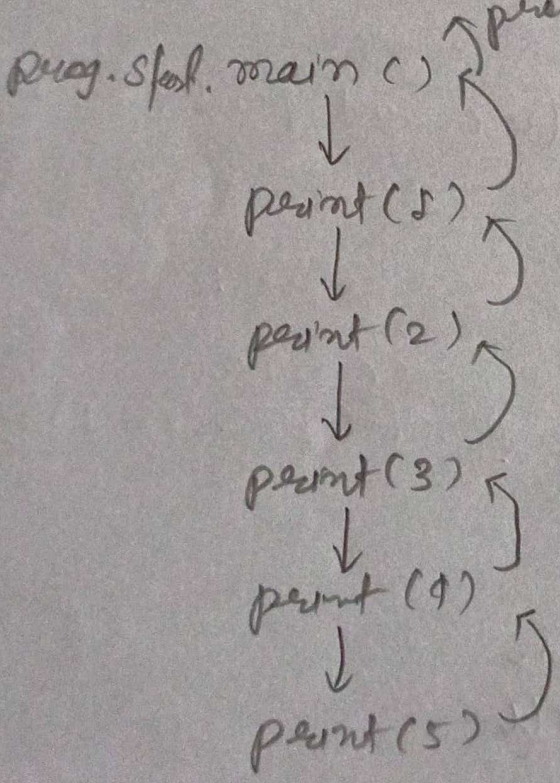
* Base condition in Recursion Condition where recursion will stop making new calls.

No base cond^n. → function will keep happening and stack will be filled again and again.

Why Reursion ?.

* It helps us in solving bigger / complex problem in a simple way.
* We can convert recursion solution into iteration & vice versa.
* space complexity is not constant because of recursive call.
* It helps us in breaking down bigger problem into smaller problem.

Visualising Recursion



52:21