

```

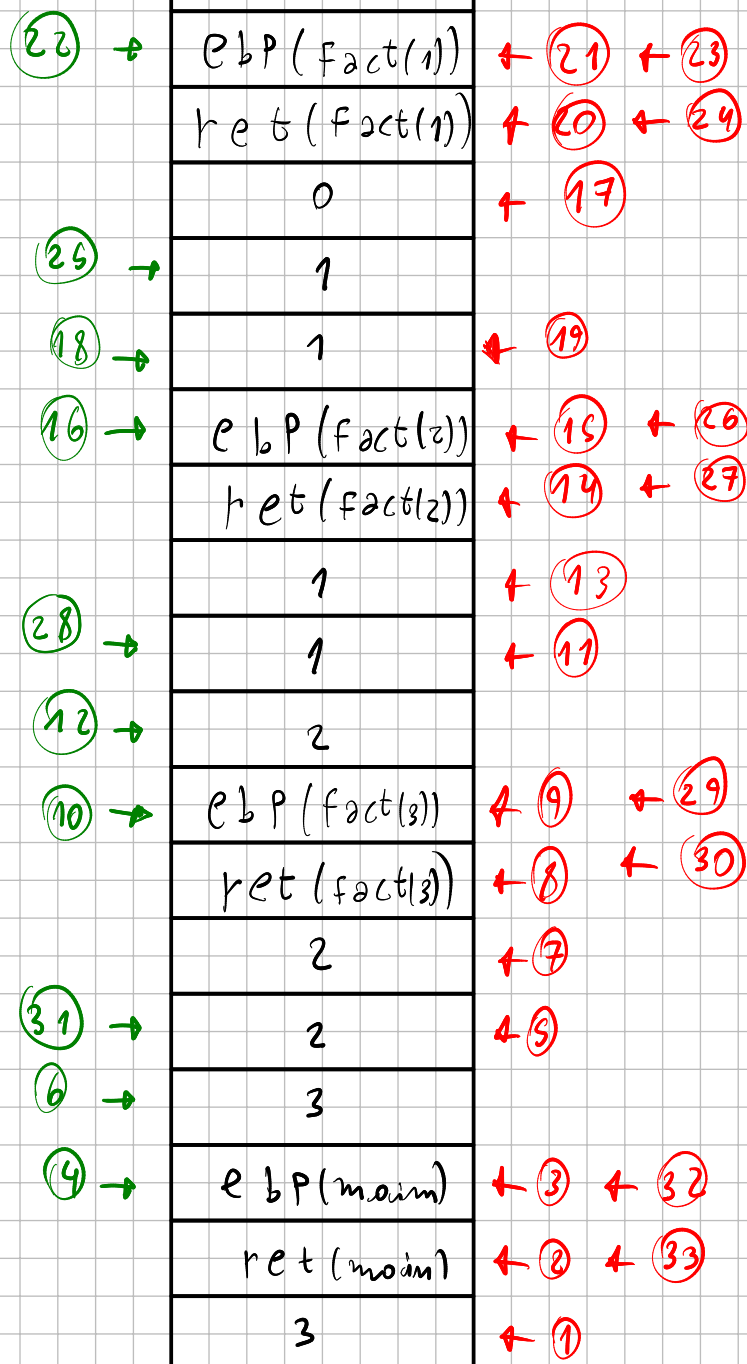
int factorial(int n) {
    int aux, fact;

    if(n==0)
        return 1;

    aux = n;
    fact = factorial(n-1);

    return fact * aux;
}

```



← 4 bytes →

- ① push 3
- ② call Factorial(3)
- ③ push ebp
- ④ mov ebp, esp
- ⑤ sub esp, 8
- ⑥ mov [ebp-4], 3
- ⑦ push 2
- ⑧ call factorial(2)
- ⑨ push ebp
- ⑩ mov ebp, esp
- ⑪ sub esp, 8
- ⑫ mov [ebp-4], 2
- ⑬ push 1
- ⑭ call factorial(1)
- ⑮ push ebp
- ⑯ mov ebp, esp
- ⑰ sub esp, 8
- ⑱ mov [ebp-8], 1
- ⑲ push 0
- ⑳ call factorial(0)
- ㉑ push ebp
- ㉒ { mov ebp, esp  
mov eax, 1
- ㉓ leave
- ㉔ ret

③① {  $\text{mov } [ebp-8], \text{eax}$   
 $\text{mov } \text{eax}, [ebp-8] * [ebp-4]$

③② leave

③③ ret

②⑤ {  $\text{mov } [ebp-8], \text{eax}$   
 $\text{mov } \text{eax}, [ebp-8] * [ebp-4]$

②⑥ leave

②⑦ ret

②⑧ {  $\text{mov } [ebp-8], \text{eax}$   
 $\text{mov } \text{eax}, [ebp-8] * [ebp-4]$

②⑨ leave

③⑦ ret