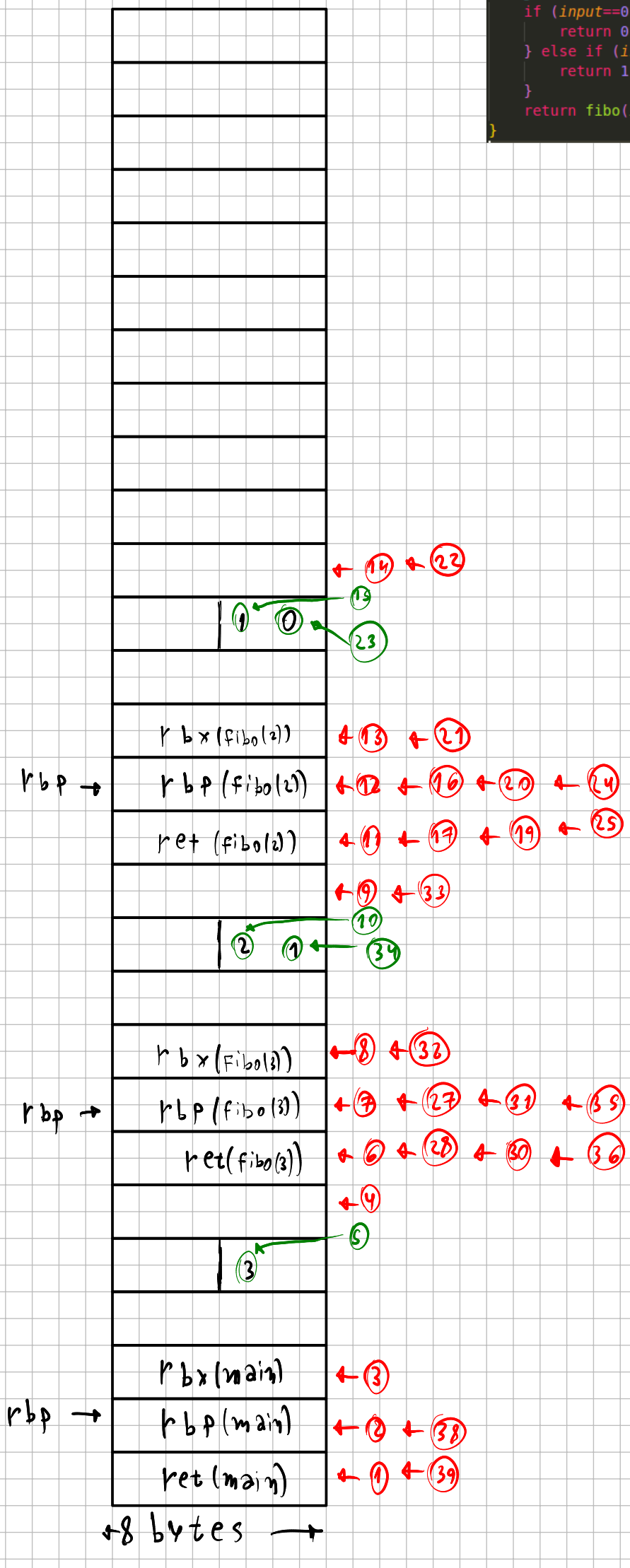
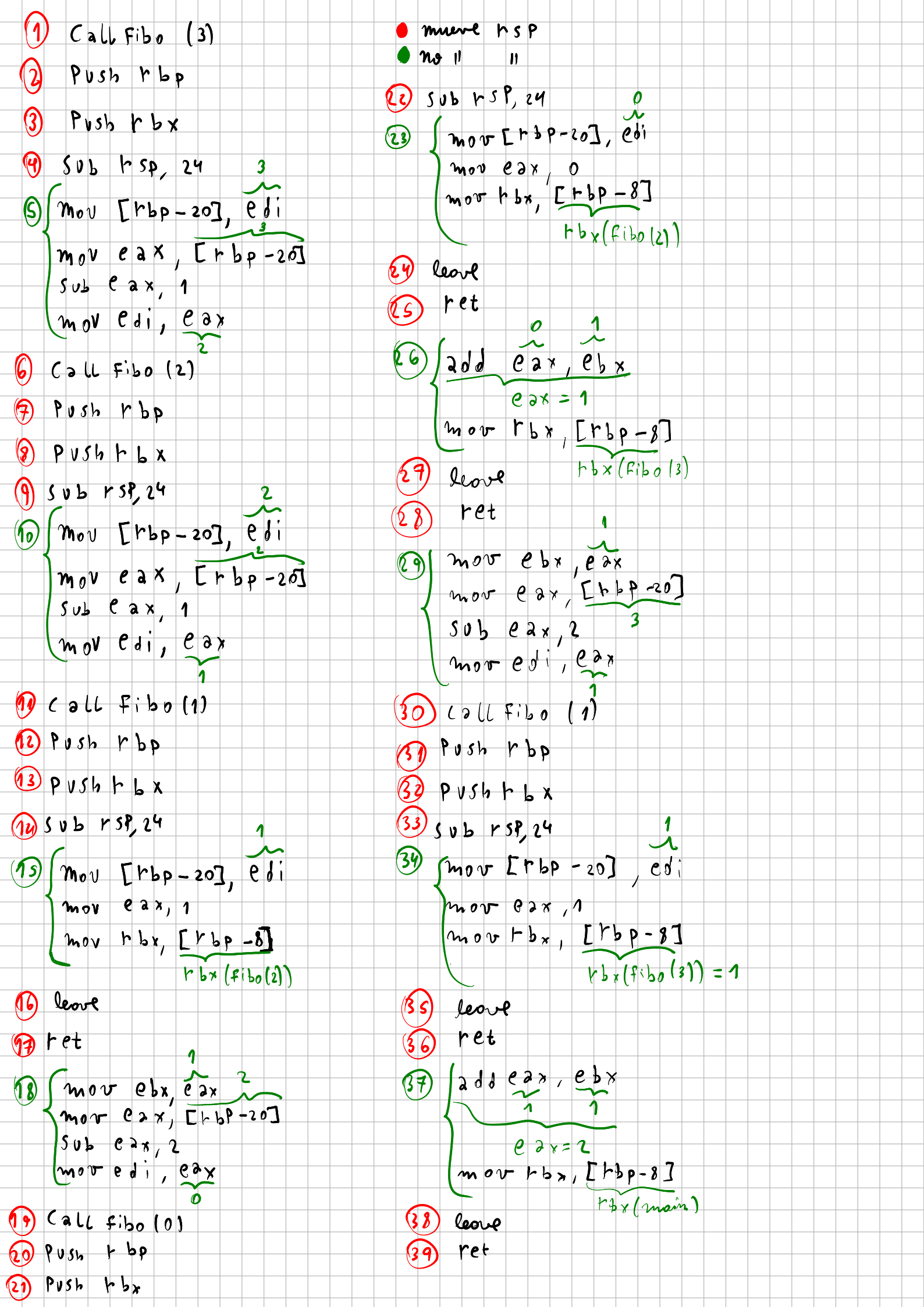
```asm
fibo:
    endbr64
    push    rbp
    mov rbp, rsp
    push    rbx
    sub rsp, 24
    mov DWORD PTR -20[rbp], edi
    cmp DWORD PTR -20[rbp], 0
    jne .L2
    mov eax, 0
    jmp .L3
.L2:
    cmp DWORD PTR -20[rbp], 1
    jne .L4
    mov eax, 1
    jmp .L3
.L4:
    mov eax, DWORD PTR -20[rbp]
    sub eax, 1
    mov edi, eax
    call    fibo
    mov ebx, eax
    mov eax, DWORD PTR -20[rbp]
    sub eax, 2
    mov edi, eax
    call    fibo
    add eax, ebx
.L3:
    mov rbx, QWORD PTR -8[rbp]
    leave
    ret
```

```c
unsigned int fibo(unsigned int input){
    if (input==0){
        return 0;
    } else if (input==1){
        return 1;
    }
    return fibo(input-1)+fibo(input-2);
}
```




1 Call Fibo (3)
2 Push rbp
3 Push rbx
4 Sub rsp, 24
5 mov [rbp-20], edi
mov eax, [rbp-20]
sub eax, 1
mov edi, eax
6 Call Fibo (2)
7 Push rbp
8 Push rbx
9 Sub rsp, 24
10 mov [rbp-20], edi
mov eax, [rbp-20]
sub eax, 1
mov edi, eax
11 Call Fibo (1)
12 Push rbp
13 Push rbx
14 Sub rsp, 24
15 mov [rbp-20], edi
mov eax, 1
mov rbx, [rbp-8]
rbx (fibo(2))
16 leave
17 ret
18 mov ebx, eax
mov eax, [rbp-20]
sub eax, 2
mov edi, eax
19 Call fibo (0)
20 Push rbp
21 Push rbx
mueve rsp
no " "
22 Sub rsp, 24
23 mov [rbp-20], edi
mov eax, 0
mov rbx, [rbp-8]
rbx (fibo(2))
24 leave
25 ret
26 add eax, ebx
eax = 1
mov rbx, [rbp-8]
rbx (fibo(3))
27 leave
28 ret
29 mov ebx, eax
mov eax, [rbp-20]
sub eax, 2
mov edi, eax
30 Call fibo (1)
31 Push rbp
32 Push rbx
33 Sub rsp, 24
34 mov [rbp-20], edi
mov eax, 1
mov rbx, [rbp-8]
rbx (fibo(3)) = 1
35 leave
36 ret
37 add eax, ebx
eax = 2
mov rbx, [rbp-8]
rbx (main)
38 leave
39 ret