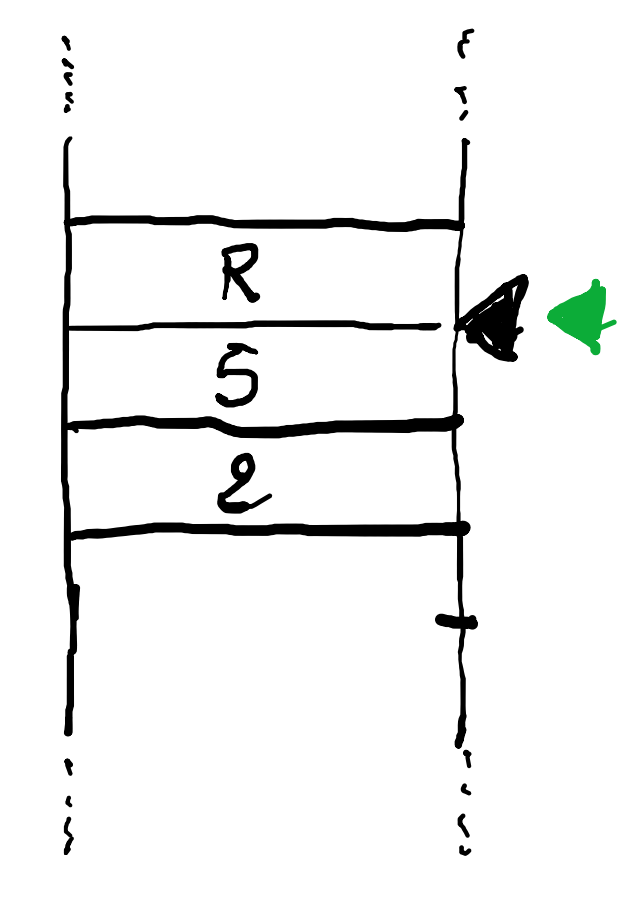


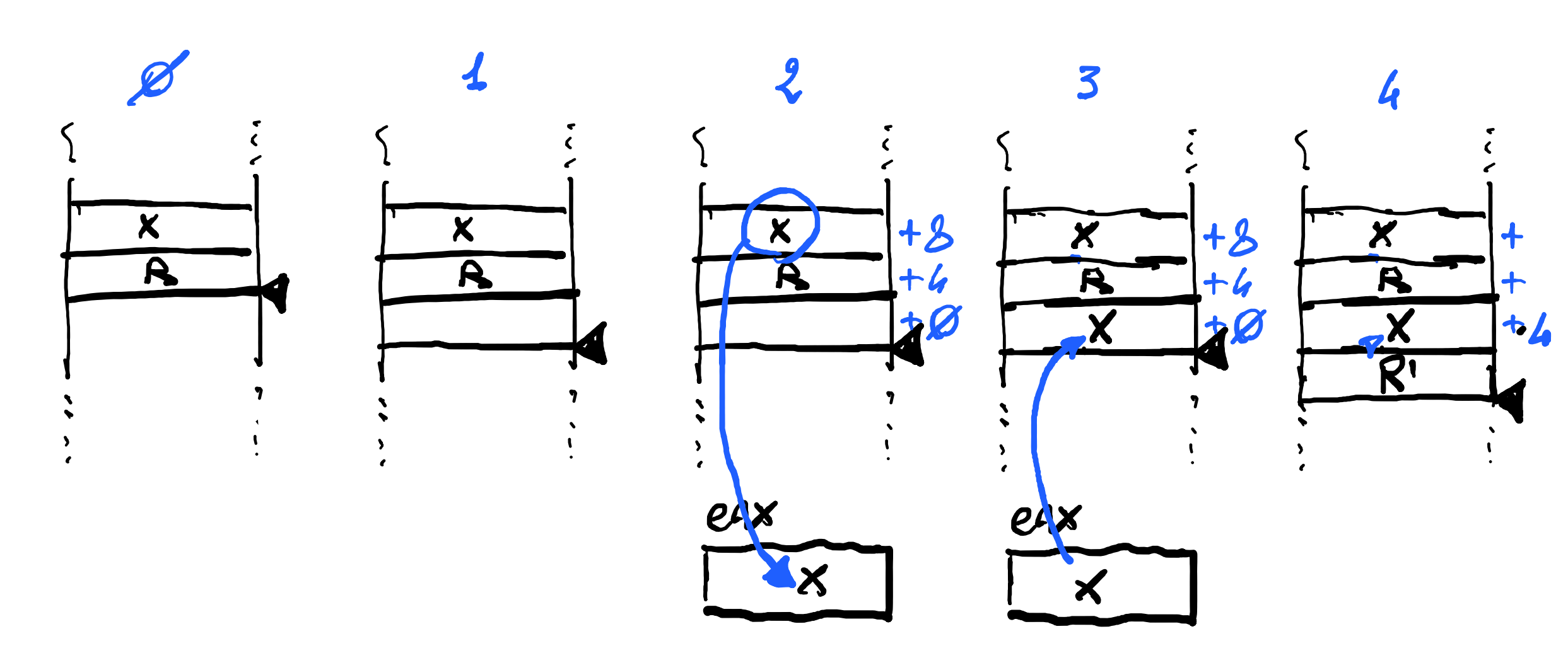
```
int f() {
    int x = g(2, 5);
    x = x + 1;
    return x;
}
```

```
.globl f
f:
    subl $8, %esp
    movl $2, (%esp)
    movl $5, 4(%esp)
    call g
    incl %eax
    addl $8, %esp
    ret
```



```
int f(int x) {
    int a = g(x);
    a = a + 1;
    return a;
}
```

```
.globl f
f:
    ① subl $4, %esp
    ② movl 8(%esp), %eax
    ③ movl %eax, (%esp)
    ④ call g
    ⑤ incl %eax
    ⑥ addl $4, %esp
    ret
```



```
int f() {
    return g() + h();
}
```

⇒
EQUIV. (Equivalent)

```
int f() {
    int a = g();
    int c = a;
    a = h();
    a = a + c;
    return a;
}
```

```
.globl f
f:
    ① call g
    ② movl %eax, %ecx
    ③ call h
    ④ addl %ecx, %eax
    ⑤ ret
```

ERRORE

ABI IA32 - REGISTRI

① REGISTRI A, C, D

- Sono registri **CALLER-SAVED**
- La funzione chiamante (caller) deve salvare il valore di questi registri se chiama una funzione.
 - La funzione chiamata (callee) può modificare liberamente il valore (o "sporcare il registro")

② REGISTRI B, DI, SI, BP

- Sono registri **CALLER-SAVED**
- La funzione chiamata (callee) deve salvare il valore di questi registri per preservarne il valore.

COME SALVARE IL VALORE DI UN REGISTRO

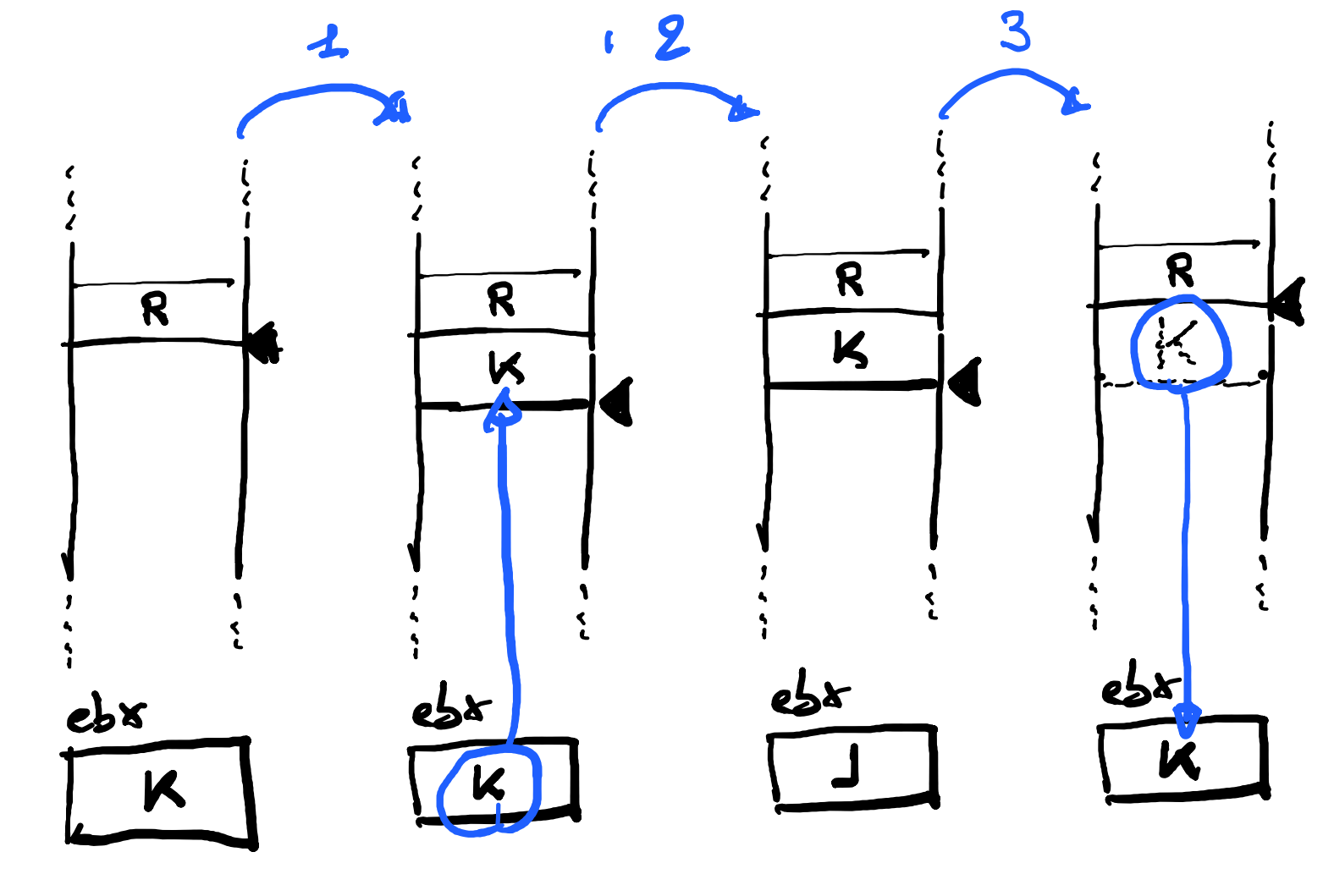
- salviamo il valore di %reg in stack
- OPERAZIONI CHE SPORCANO %reg
- Recuperiamo da stack il valore di %reg

MODO ESPLICITO

```
① subl $4, %esp
   movl %ebx, (%esp)
② ---
③ movl (%esp), %ebx
   addl $4, %esp
```

MODO PUSH + POP

```
① pushl %ebx
② ---
③ popl %ebx
```



```
int f() {
    int a = g();
    int c = a;
    a = h();
    a = a + c;
    return a;
}
```

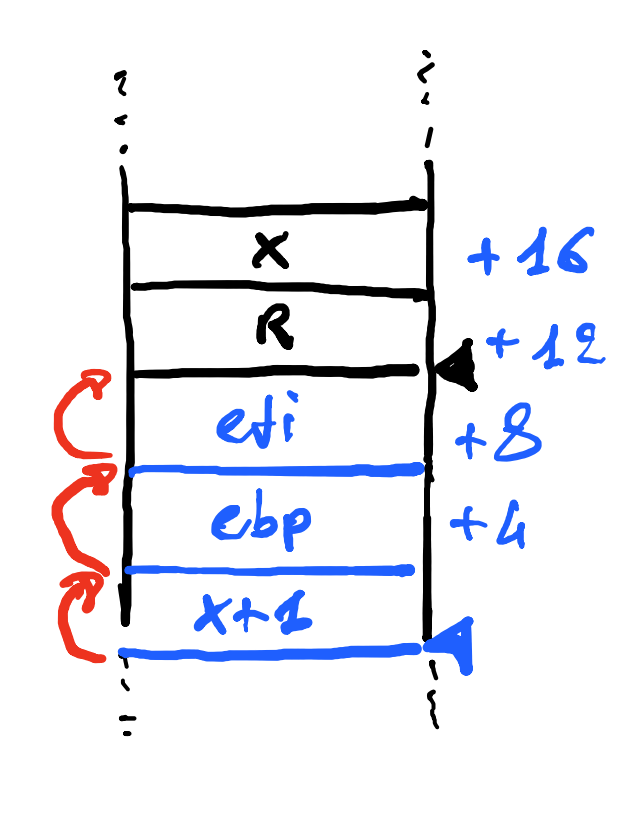
```
.globl f
f:
    ① call g
    ② pushl %ebx
    ③ movl %eax, %ebx
    ④ call h
    ⑤ addl %ebx, %eax
    ⑥ popl %ebx
    ⑦ ret
```

CORRETTA!

ESEMPIO

```
int f(int x) {
    int y = 1;
    int z = 2;
    int a = g(x + 1);
    a = a + y + z;
    return a;
}
```

```
.globl f
f:
    prologo:
        pushl %edi
        pushl %ebp
        subl $4, %esp
        movl $1, %edi
        movl $2, %ebp
        movl 16(%esp), %eax
        incl %eax
        movl %eax, (%esp)
        call g
        addl %edi, %eax
        addl %ebp, %eax
    epilogo:
        addl $4, %esp
        popl %ebp
        popl %edi
        ret
```



QUALI REGISTRI POSSO USARE ?

- No una funzione che **NON** chiama altre funzioni
 - Uso A, C, D
 - Se non bastano uso B, DI, SI, BP ma devo preservarne il valore (prologo + epilogo)
- No una funzione che chiama altre funzioni
 - Uso B, DI, SI, BP preservandone il valore (prologo + epilogo)
 - Uso A, C, D per valori temporanei in blocchi di istruzioni **SENZA** call
 - Uso A, C, D ma preservo il valore prima di qualsiasi chiamata a funzione

```
pushl %ecx
subl $4, %esp
movl ?, (%esp)
call g
addl $4, %esp
popl %ecx
```