

Esercizi Assembly

Provare a ricostruire le istruzioni originali in C

```
push    %ebp
mov     %esp,%ebp
sub     $0x8,%esp
call    80483e9 <bar>
leave
ret
```

La prima parte da push a sub indica la creazione di uno stack
Successivamente call esegue una chiamata della funzione bar

Leave indica rimozione dello stack
Ret = return 0

```
push    %ebp
mov     %esp,%ebp
sub     $0x8,%esp
call    80483fb <baz>
call    8048400 <quux>
leave
ret
```

```
push    %ebp
mov     %esp,%ebp
pop     %ebp
ret
```

```
push    %ebp
mov     %esp,%ebp
mov     $0x0,%eax
movl    $0x1,(%eax)
pop     %ebp
ret
```

```
push    %ebp
mov     %esp,%ebp
and     $0xffffffff,%esp
call    80483dc <foo>
mov     $0x0,%eax
leave
ret
```

```
void foo();
void bar();
void baz();
void quux();
```

```
void foo() {
    bar();
}
```

```
void bar() {
    baz();
    quux();
}
```

```
void baz() {
    //do nothing
}
```

```
void quux() {
    *(int*)(0) = 1;
}
```

```
int main() {
    foo();
    return 0;
}
```

Indica un puntatore ad un
altro puntatore che resetta a
0, per poi impostarli a 1