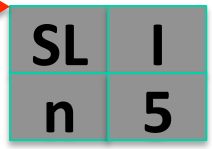
PROGRAMMAZIONE 2
24bis. Simulare il runtime

Un esempio

```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

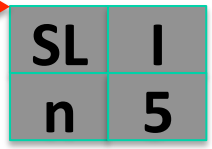
Runtime stack

A →




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack: simulazione

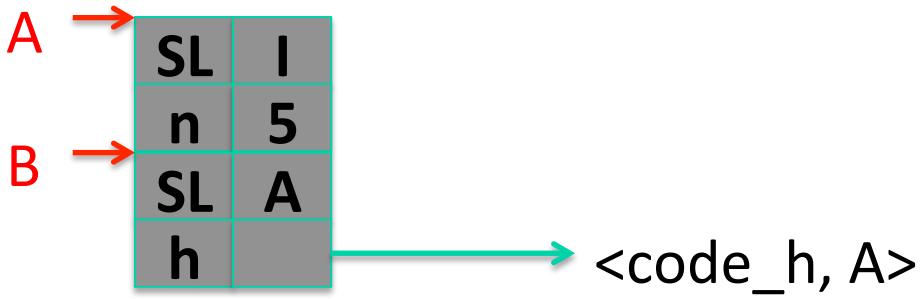
A → 

Env_A(n) = 5

Env_A(m) = unbound
for all m != n

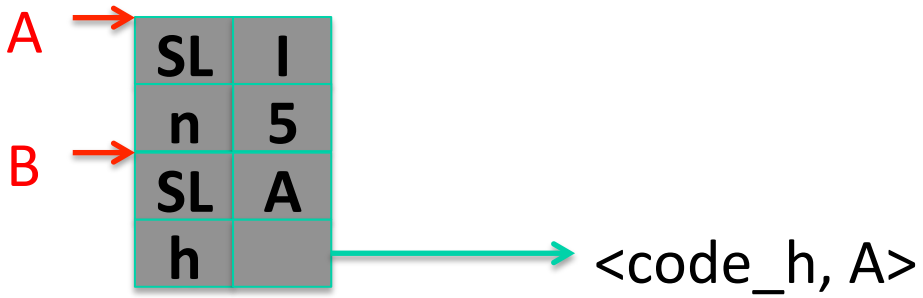
```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

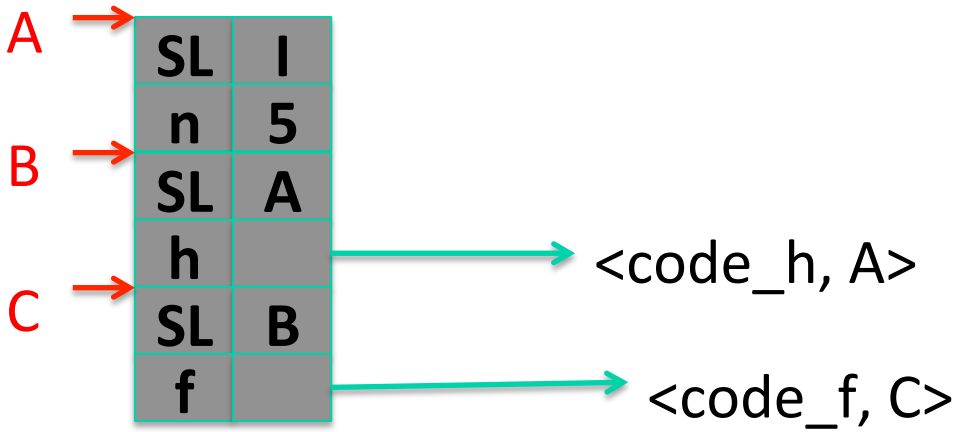
Runtime stack: simulazione



Env_A(n) = 5
Env_A(m) = unbound
for all m != n
Env_B (n) = 5
Env_B(h) = <code_h, Env_A>

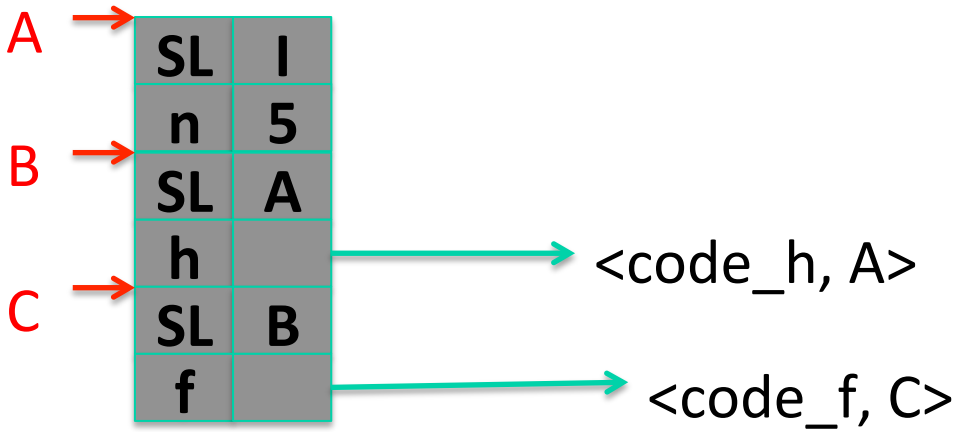
```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

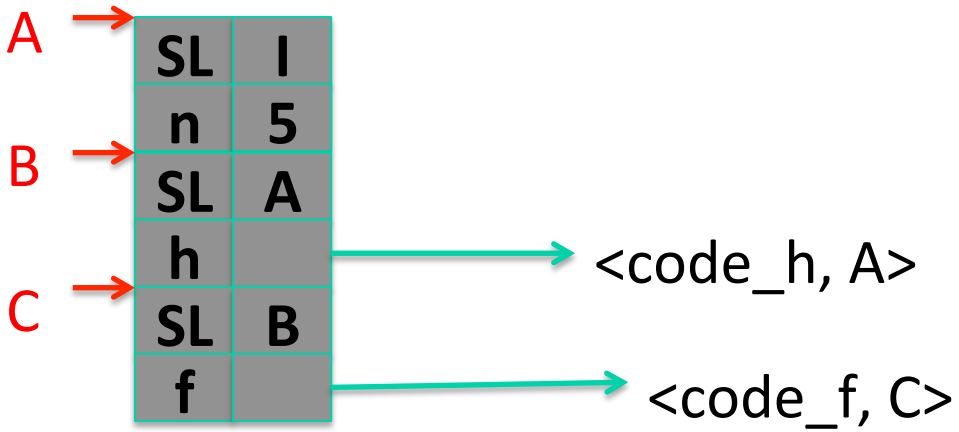
Runtime stack: simulazione



Env_A(n) = 5
Env_A(m) = unbound
for all m != n
Env_B (n) = 5
Env_B(h) = <code_h, Env_A>
Env_C(f) = <code_f, Env_C>
Env_C(h) = <code_h, Env_A>
Env_C(n) = 5

```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack: simulazione

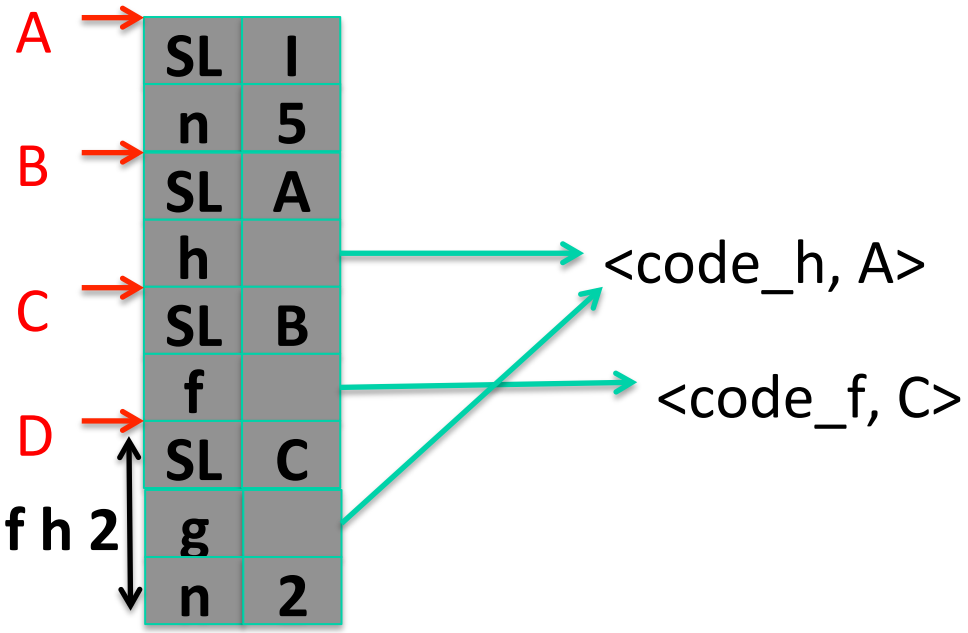


Env_A(n) = 5
Env_A(m) = unbound
for all m != n
Env_B (n) = 5
Env_B(h) = <code_h, Env_A>
Env_C(f) = <code_f, Env_C>
Env_C(h) = <code_h, Env_A>
Env_C(n) = 5

Definizione ricorsiva

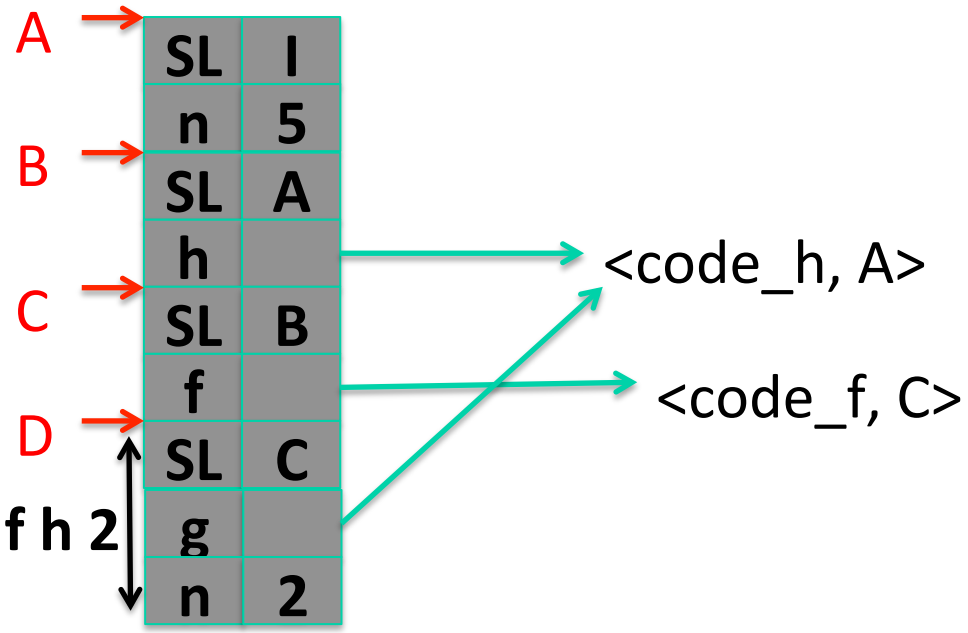
```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

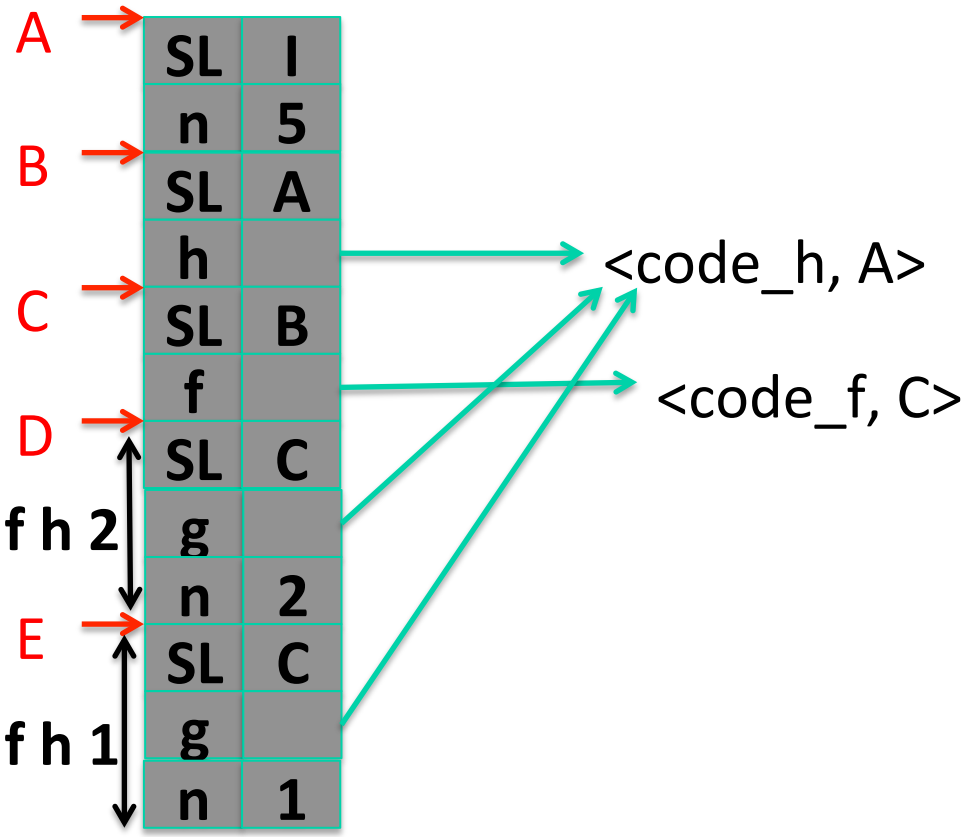
Runtime stack: simulazione



Env_A(n) = 5
Env_A(m) = unbound
for all m != n
Env_B (n) = 5
Env_B(h) = <code_h, Env_A>
Env_C(f) = <code_f, Env_C>
Env_C(h) = <code_h, Env_A>
Env_C(n) = 5
Env_D(g) = <code_h, Env_A>
Env_D(n) = 2
Env_D(f) = <code_f, Env_C>
Env_D(h) <code_h, Env_A>

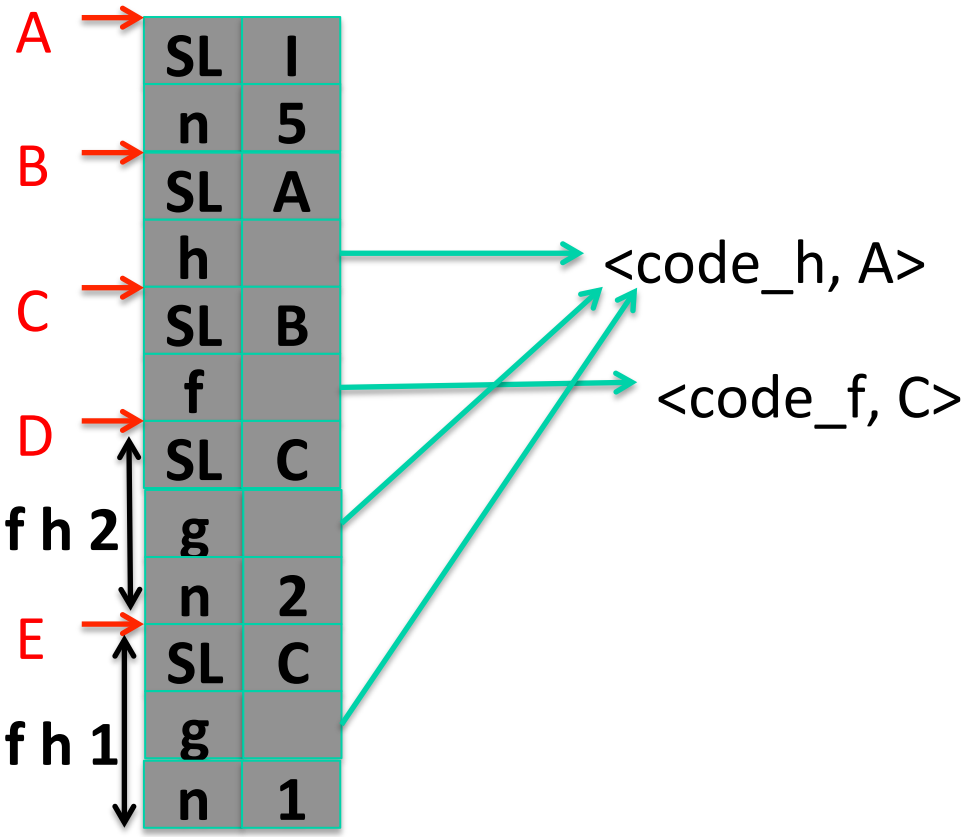
```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Runtime stack: simulazione




```
let n = 5;;
let h = fun x -> n + x ;;
let rec f g n = if n = 1 then g(n)
                else n * f g (n-1);;
f h 2;;
```

Env_A(n) = 5
Env_A(m) = unbound
for all m != n
Env_B (n) = 5
Env_B(h) = <code_h, Env_A>
Env_C(f) = <code_f, Env_C>
Env_C(h) = <code_h, Env_A>
Env_C(n) = 5
Env_D(g) = <code_h, Env_A>
Env_D(n) = 2
Env_D(f) = <code_f, Env_C>
Env_D(h) <code_h, Env_A>
Env_E(g) = <code_h, Env_A>
Env_E(n) = 1
Env_E(f) = <code_f, Env_C>
Env_E(h) <code_h, Env_A>