Simulazione run-time

```
let x =1;;

let f1  = fun y z  ->  let  f2 = fun x  ->  x * (y+ z)  in f2 x * (y-z);;

let rev lst =
   let rec aux acc = function
     | [] -> acc
     | h::t -> aux (h::acc) t in aux [] lst

let rec apply g  n  (lst : int list) =
                          match (rev lst) with
                           |[] -> []
                           | hd::ls ->  (g hd n):: apply g n ls;;

let res  = apply  f1  (x+1 )  [2;5;1];;
```

	A	SL = init	CL = init
		x	1
	B	SL= A	CL=A
		f1	M1
	C	SL = B	CL = B
		rev	M2
	D	SL = C	CL = C
		Apply	M3
Apply	E	SL = D	CL=D
f1		g	M1
x+1		n	2
[2;5;1]		lst	[2,5,1]
		hd	1
		lst	[5;2]
		res	[1;5;2]
g	F	SL=A	CL= E
hd		y	1
n		z	2
		result	
		f2	M5
f2	G	SL= F	CL=F
x*(y-z)		x	-1
		result	

M1	<cf1, A>
M2	<crev, B>
M3	<capp, D>
M4	<caux, F>
M5	<cf2, F>

```
let x =1;;
let f1  = fun y z  ->  let  f2 = fun x  ->  x * (y+ z)  in f2 x * (y-z);;
let rev lst =
   let rec aux acc = function
     | [] -> acc
     | h::t -> aux (h::acc) t in aux [] lst
let rec apply g  n  (lst : int list) =
                          match (rev lst) with
                          |[] -> []
                          | hd::ls ->   (g hd n):: apply g n ls;;
```

let res = **apply f1 (x+1) [2;5;1];;**