Simulazione run-time

Α	SL=Init	CL=init
	х	1

let x =1;;

В

SL = init	CL = init
x	1
SL= A	CL=A
f1	M1

M1 <cf1, A>

В

C

 SL = init
 CL = init

 x
 1

 SL = A
 CL = A

 f1
 M1

 SL = B
 CL = B

 rev
 M2

M1	<cf1, a=""></cf1,>
M2	<crev, b=""></crev,>



D

SL = init CL = init Х 1 В SL= A CL=A f1 М1 SL = B CL = B С

rev

SL = C

Apply

M2

CL = C

М3

M1 <cf1, A> M2 <crev, B> М3 <capp, D>



```
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];;
```

В

C

D

Ε

Apply f1

x+1

[2;5;1]

SL = init	CL = init
X	1
SL= A	CL=A
f1	M1
SL = B	CL = B
rev	M2
SL = C	CL = C
Apply	М3
SL = D	CL=D
g	M1
n	2
lst	[2,5,1]
hd	0
lst	[]

result

<cf1, a=""></cf1,>
<crev, b=""></crev,>
<capp, d=""></capp,>



```
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];;
```

	Α	SL = init	CL = init
		x	1
	В	SL= A	CL=A
		f1	M1
	С	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	0
		lst	[]
		result	
rev	F	SL = B	CL= E
lst		I	[2,5,1]
		aux	M4

M1	<cf1, a=""></cf1,>
M2	<crev, b=""></crev,>
M3	<capp, d=""></capp,>
M4	<caux, f=""></caux,>



	Α	SL = init	CL = init
		х	1
	В	SL= A	CL=A
		f1	M1
	С	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	0
		lst	Ū
		result	
rev	F	SL = B	CL= E
lst		ı	[2,5,1]
		aux	M4
aux	G	SL= F	CL=F
[]		acc	[]
Ĭ		tmp	[2,5,1]
		h	2
		t	[5;1]
		result	L-, .
aux	н	SL=L	CL=G
h::a		acc	[2]
t		tmp	[5;1]
-		h	5
		t	[1]
		result	
aux		- 3 4.1 4	

M1	<cf1, a=""></cf1,>
M2	<crev, b=""></crev,>
M3	<capp, d=""></capp,>
M4	<caux, f=""></caux,>

```
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];;
```

Α
В
С
D

Ε

M1	<cf1, a=""></cf1,>
M2	<crev, b=""></crev,>
M3	<capp, d=""></capp,>
M4	<caux, f=""></caux,>
M5	<cf2, f=""></cf2,>

Apply f1 x+1 [2;5;1]

g F hd n

f2 G x*(y-z)

SL = init	CL = init
x	1
SL= A	CL=A
f1	M1
SL = B	CL = B
rev	M2
SL = C	CL = C
Apply	М3
SL = D	CL=D
g	M1
n	2
lst	[2,5,1]
hd	1
lst	[5;2]
res	[1;5;2]
SL=A	CL= E
у	1
Z	2
result	
f2	M5
SL= F	CL=F
Х	-1
result	