Problem 1)

1 { int x;	[x:?]
2 int y;	[y:?, x:?]
3 y := 1;	[y:1, x:?]

4 { int f(int x) {
...
10 };
	[f{}, y:1, x:?]

first f call
11 x := f(2);
x=2
4 { int f(int x) {
x is not 0, so
7 else { 2-1 y=1
8 y := f(x–1)∗y+1 };
	[f{}, y:1, x:?]

x still not 0, so 2nd call for f
11 x := f(2); :) {
7 else {
8 y := f(x–1)∗y+1 };
	[[x:2, f{}, y:1],[f{}, y:1, x:?]]

x == 0 here 3rd call for f
4 { int f(int x) {
5 if x=0 then {
6 y := 1 }
	[[x:1, f{}, y:1],
	[x:2, f{}, y:1],
	[f{}, y:1, x:?]]

9 return y;	[[x:0, f{}, y:1], [x:1, f{}, y:1], [x:2, f{}, y:1], [f{}, y:1, x:?],] what we have right before the return y
back to 2nd call 8 y := f(x–1)∗y+1 }; 9 return y; 2	[[y:1, x:1, f{}, y:1], [x:2, f{}, y:1], [f{}, y:1, x:?] f(x-1)*y+1 = 2
back at first call of f 8 y := f(x–1)∗y+1 }; 9 return y; 4	[[y:2, x:2, f{}, y:1], [f{}, y:1, x:?]] f(...) = 4
11 x := f(2);	[x:4]

Problem 2)

In static scoping, z would be assigned 21. (x*y, 3*7)

The earliest x, 3, would be referred to, as well as the earliest y.

In dynamic scoping, z would be assigned 26. (2*13)

The latest definition of x would be in function g(int x), which a 2 was passed into.

Latest definition of y would be the 13 in where the block of the function g exists.

Problem 3)

Call by value:
[x:14, g{}, f{}, z:?, y:7]
[y:16, a:15, x:14, g{}, f{}, z:?, y:7]   at first f
[y:32, x:14, g{}, f{}, z:?, y:7] after first function call
[a:-15, y:32, x:14, g{}, f{}, z:?, y:7] in second f
[z:-29, y:32, x:14, g{}, f{}, z:? y:7] after second f
[z:-28, y:7] returned to the first call of z

Call By Name:

[g{}, f{}, z:?, y:7], before first function call

[x:(y*2), g{}, f{}, z:g(y*2), y:7],   at g

[a:(x+1), x:y*2, g{}, f{}, z:g(y*2), y:7]   in first f call

[y:(a+1), a:(x+1), x:y*2, g{}, f{}, z:g(y*2), y:7]   in first f call

_y =_
_(a+1)_
_((x+1)+1)_
_(((y*2)+1)+1)_
_(((14)+1)+1)   =   16_

return =
_16+a_
_16 + x+1_
_16 + ((y*2) + 1)_
_16 + (14 + 1) = 31_

[y:32, a:(x+1), x:y*2, g{}, f{}, z:g(y*2), y:7]   After first f call
[a:(x-y+3), y:32, x:y*2, g{}, f{}, z:g(y*2), y:7]   in second f call
_[y:(a+1), a:(x-y+3), x:y*2, g{}, f{}, z:g(y*2), y:7]_

_y = (a+1) = (x-y+3)+1 = (y*2 - 32 + 3) + 1 = (64 - 32 + 3) + 1_
_= 36_

return = (36+(x-y+3)) = (36+(64-32+3)) = 71

[z:71, y:32, x:y*2, g{}, f{}, z:g(y*2), y:7], return z + 1

[z: 72, y:32, y:7]

Z = 72, newer y = 32, original y = 7

Call by Need:

[g{}, f{}, z:?, y:7], before first function call

[x:y*2, g{}, f{}, z:?, y:7]

[a:x+1, x:y*2, g{}, f{}, z:?, y:7]

[y:(a+1), a:x+1, x:y*2, g{}, f{}, z:?, y:7]

[y:16, a:15, x:14, g{}, f{}, z:?, y:7]   evaluated according to call by need

return y+a from first call of f

[y:32, x:14, g{}, f{}, z:?, y:7]

[a:(x-y+3), y:32, x:14, g{}, f{}, z:?, y:7]

[y:(a+1), a:(x-y+3), y:32, x:14, g{}, f{}, z:?, y:7]

[y:-14, a:-15, y:32, x:14, g{}, f{}, z:?, y:7]   evaluated according to call by need

y = (x-y+3+1) = (14-32+4) = -14

[z:-29, y:32, x:14, g{}, f{}, z:?, y:7]

z = -28, local y in g = 32, global y = 7