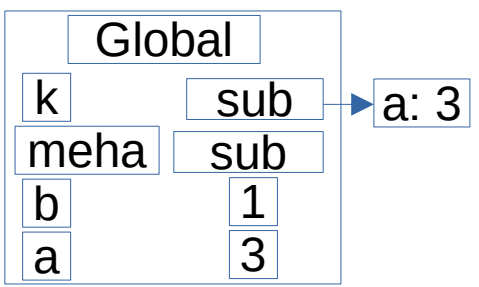
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

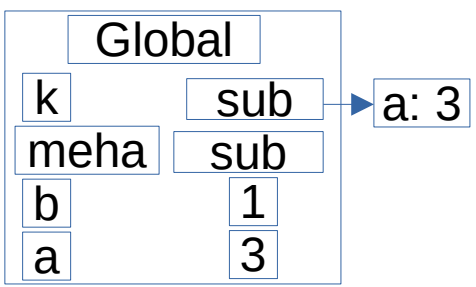
Alcance Estatico y Clausura con Asociacion Superficial

Inicialmente las variables Globales son declaradas.

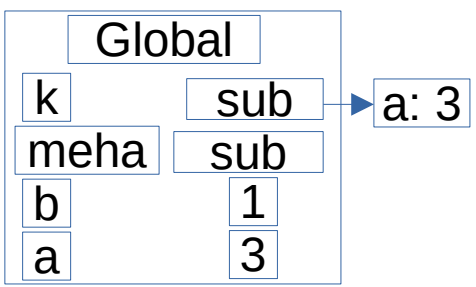
Las clausuras se realizan Al Ejecutar las subrutinas



```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

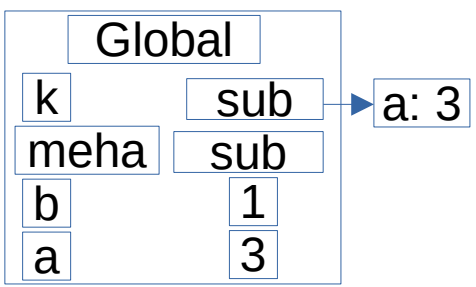


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



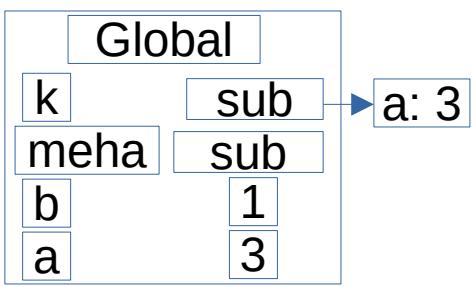
k0
4 a: 7
3 meha: sub
2 ku: meha(global)
1 go: meha(global)
0 b: 3

```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



k0
4 a: 7
3 meha: sub
2 ku: meha(global)
1 go: meha(global)
0 b: 3

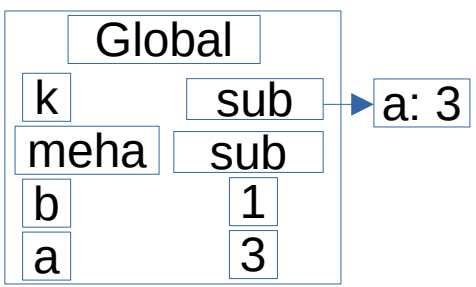
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



k1	9	a: 7
	8	meha: sub
	7	ku: meha(global)
	6	go: meha3
	5	b: 12

k0	4	a: 7
	3	meha: sub
	2	ku: meha(global)
	1	go: meha(global)
	0	b: 3

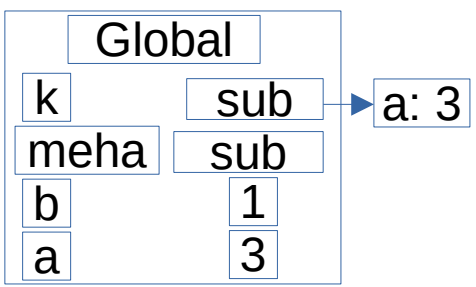
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



k1	9	a: 7
	8	meha: sub
	7	ku: meha(global)
	6	go: meha3
	5	b: 12

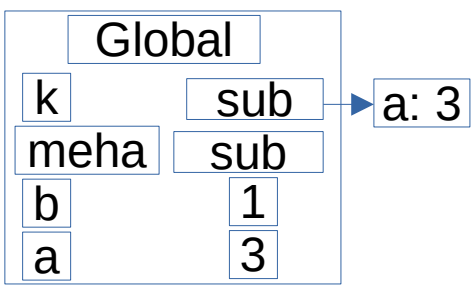
k0	4	a: 7
	3	meha: sub
	2	ku: meha(global)
	1	go: meha(global)
	0	b: 3

```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



k2	14	a: 7
	13	meha: sub
	12	ku: meha8
	11	go: ku7
	10	b: 21
k1	9	a: 7
	8	meha: sub
	7	ku: meha(global)
	6	go: meha3
	5	b: 12
k0	4	a: 7
	3	meha: sub
	2	ku: meha(global)
	1	go: meha(global)
	0	b: 3

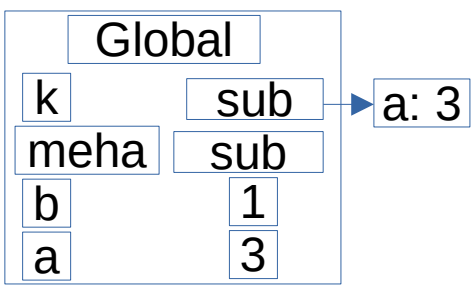
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```



k2	15	b: -15
	14	a: 7
	13	meha: sub
	12	ku: meha8
	11	go: ku7
	10	b: 21
k1	9	a: 7
	8	meha: sub
	7	ku: meha(global)
	6	go: meha3
	5	b: 12
k0	4	a: 7
	3	meha: sub
	2	ku: meha(global)
	1	go: meha(global)
	0	b: 3

```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

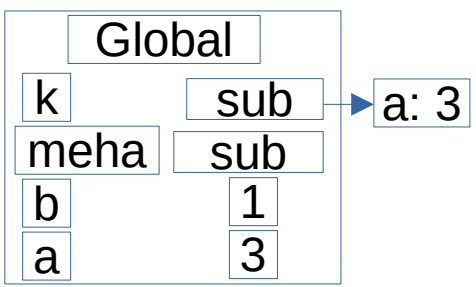
Como se esta ejecutando go se le crea la clausura
Recordando que go11 = ku7= meha(global)

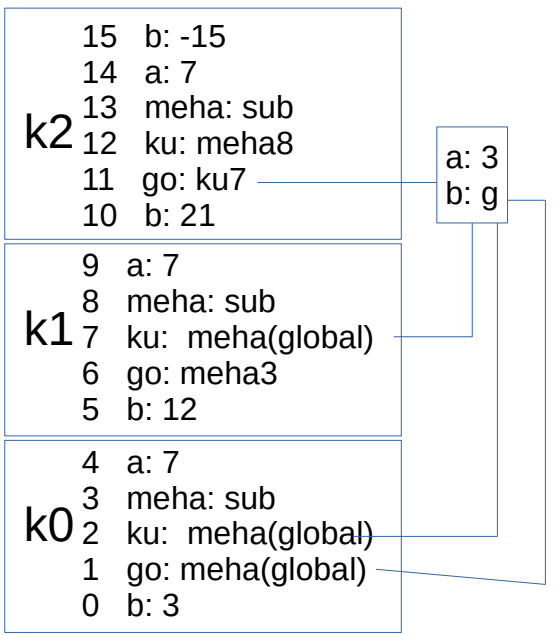


k2	15	b: -15
	14	a: 7
	13	meha: sub
	12	ku: meha8
	11	go: ku7
	10	b: 21
k1	9	a: 7
	8	meha: sub
	7	ku: meha(global)
	6	go: meha3
	5	b: 12
k0	4	a: 7
	3	meha: sub
	2	ku: meha(global)
	1	go: meha(global)
	0	b: 3

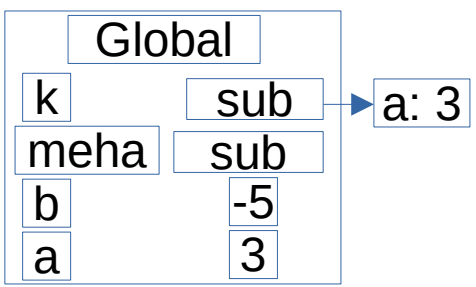
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

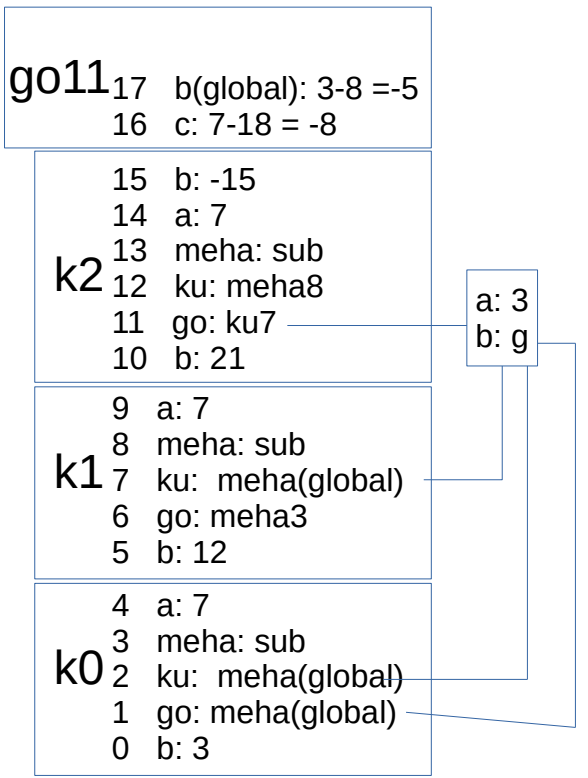
Como hay alcance estatico, en la clausura
Tenemos… a es 3 y b es global





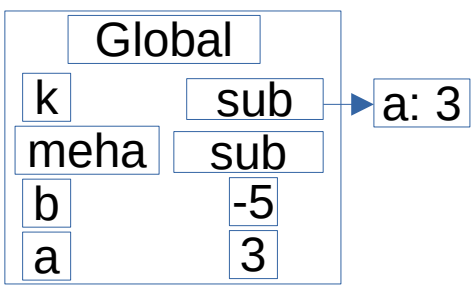
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

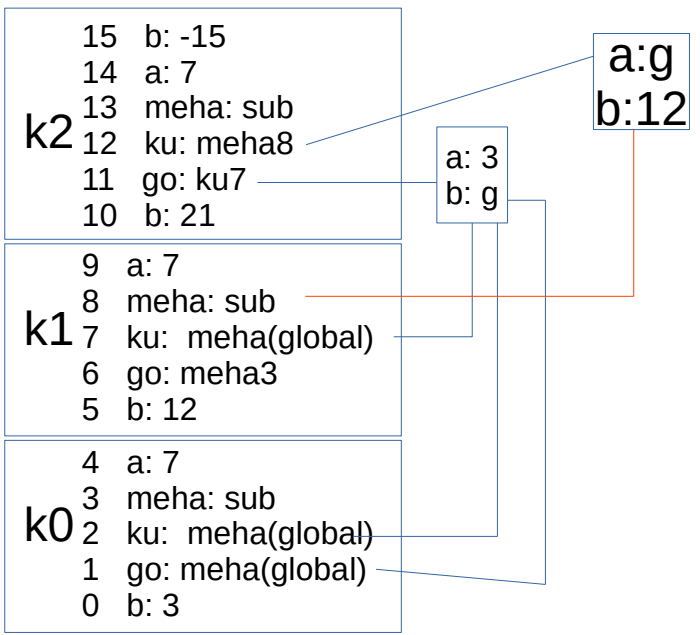




```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

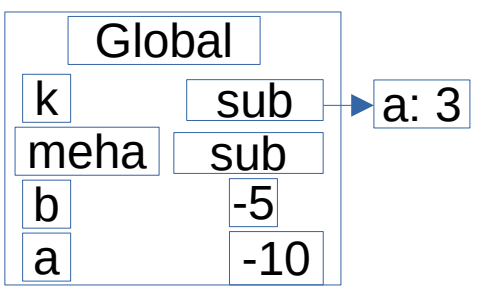
Hacemos la clausura de ku12

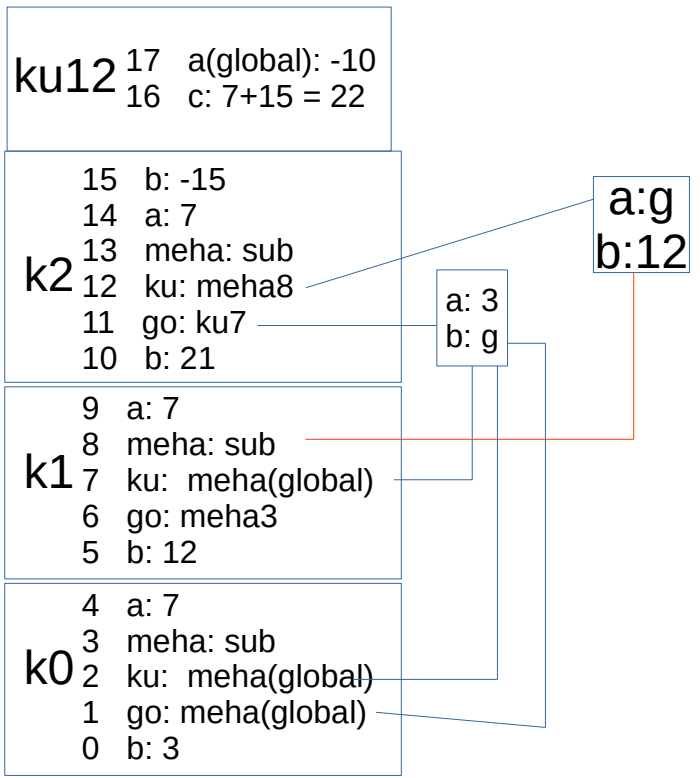




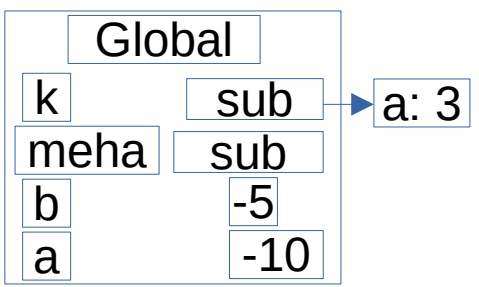
```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

    sub meha(int c) {
            a := b - c;
    }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

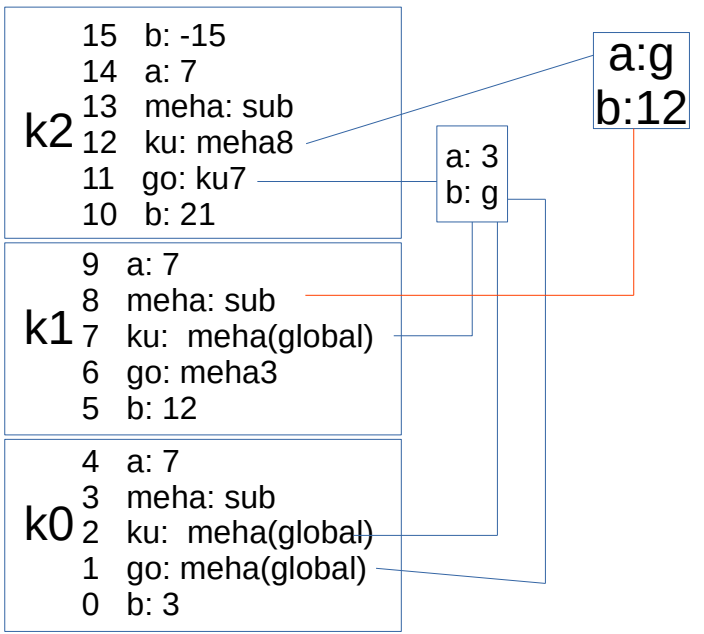
a(global) se convierte en -10





```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

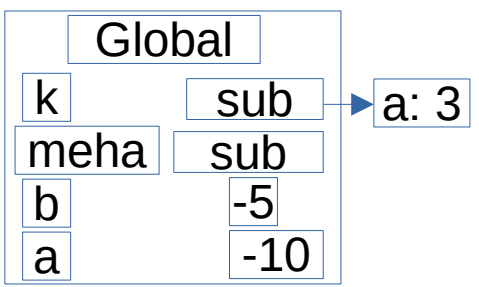


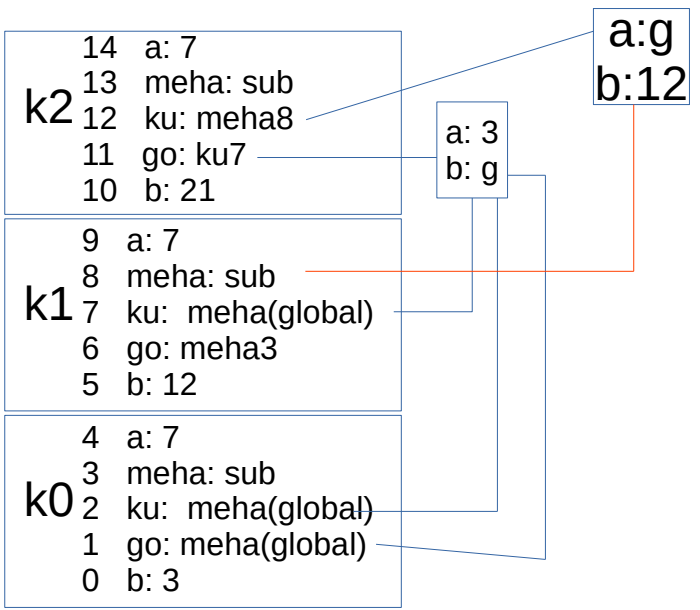


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

7 21

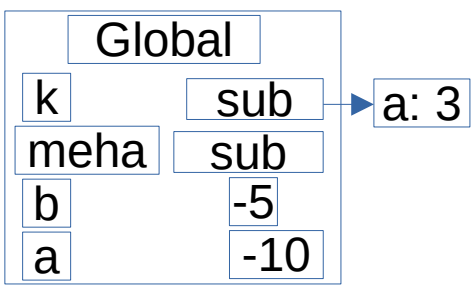


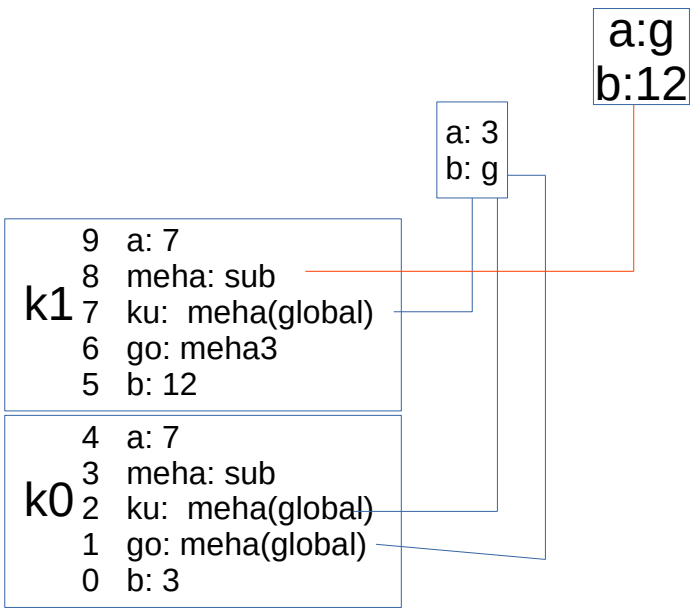


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

7 21

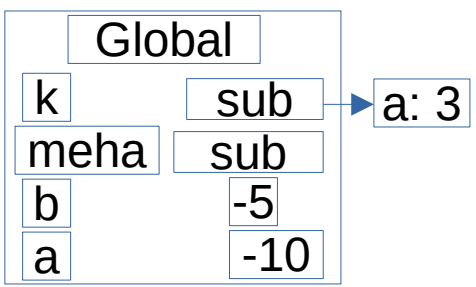


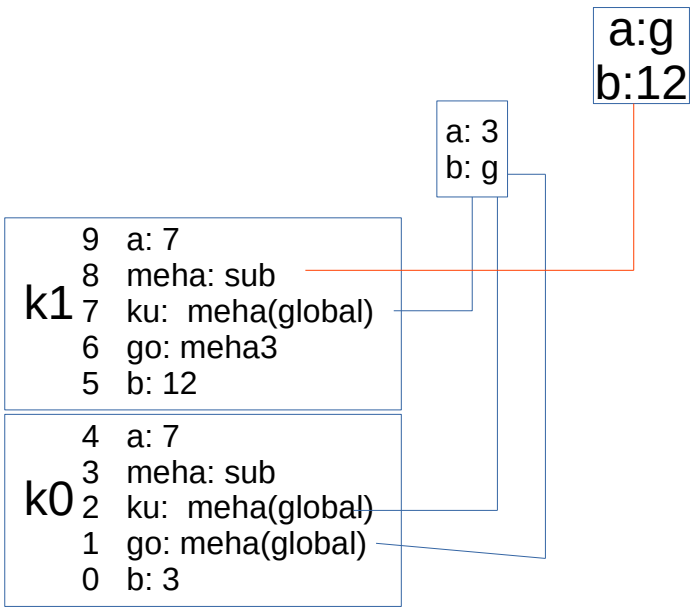


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

```
7  21
7  12
```

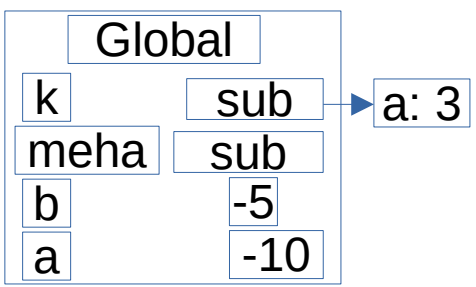


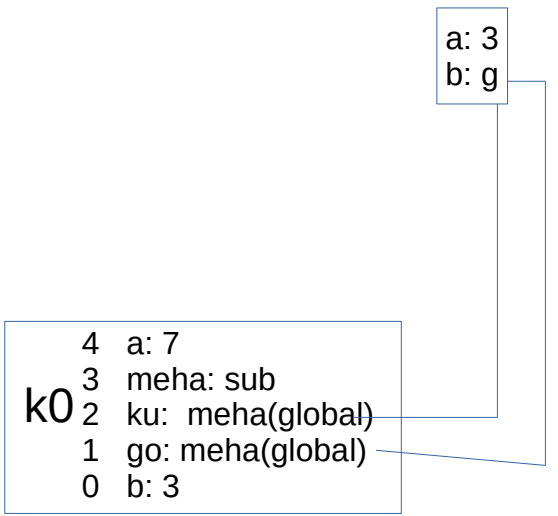


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

```
7  21
7  12
```

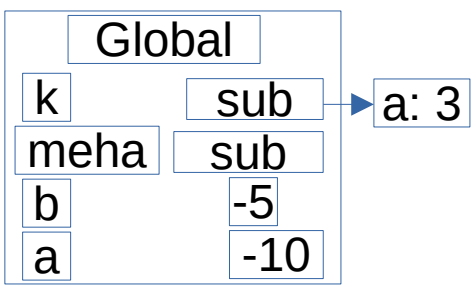


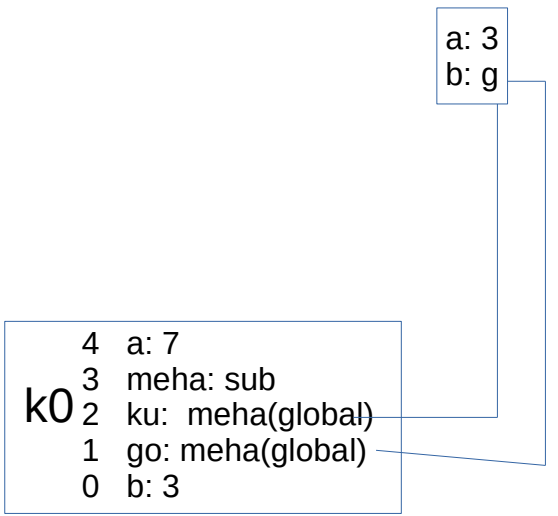


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

```
7  21
7  12
7   3
```

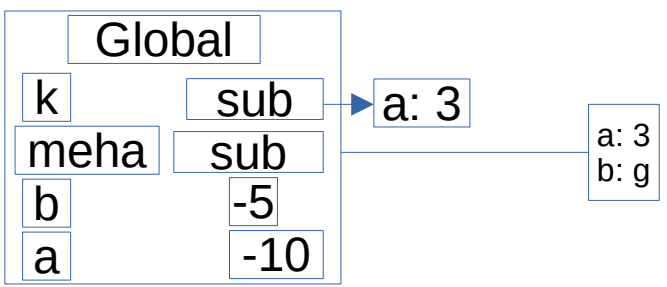




```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

```
7  21
7  12
7   3
```

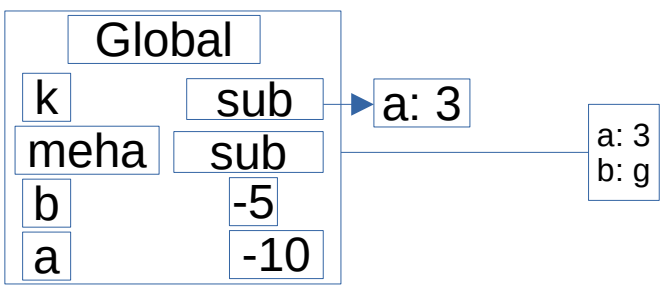


```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
```

```
print(a, b)
```

Salida

```
7  21
7  12
7   3
-10 -5
```



```
int a = 2 + 1, b = 1;

sub meha(int c)  {
        b := a + c;
}
sub k (int b, sub go, sub ku) {

        sub meha(int c) {
                a := b - c;
        }

        int a = 6 + 1;
        if (b < 3 * (2 + 1)) {
                k(b + 3 * (2 + 1), meha, go);
        } else if (b < 6 * (2 + 1)) {
                k(b + 3 * (2 + 1), ku, meha);
        } else {
                int b = 6 - b
                go(a + b);
                ku(a - b);
        }
        print(a, b)
}
k(a, meha, meha);
print(a, b)
```

Salida

```
 7  21
 7  12
 7   3
-10 -5
```

Finaliza la ejecucion