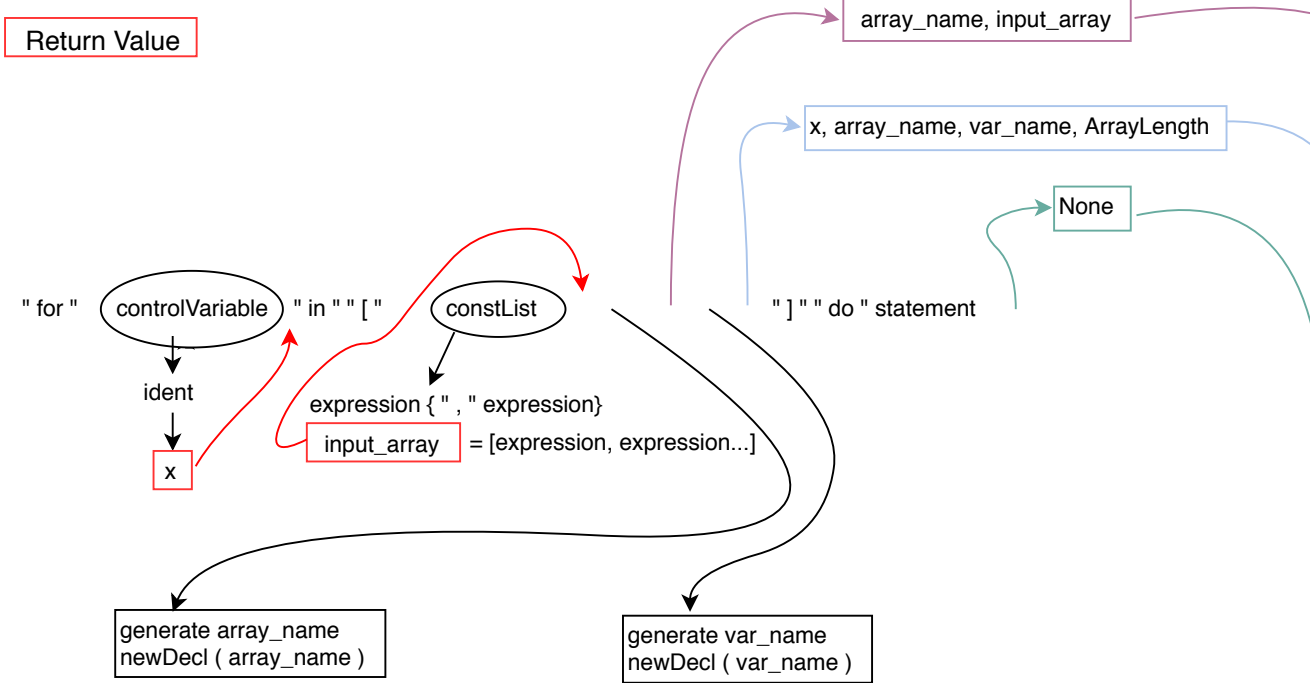
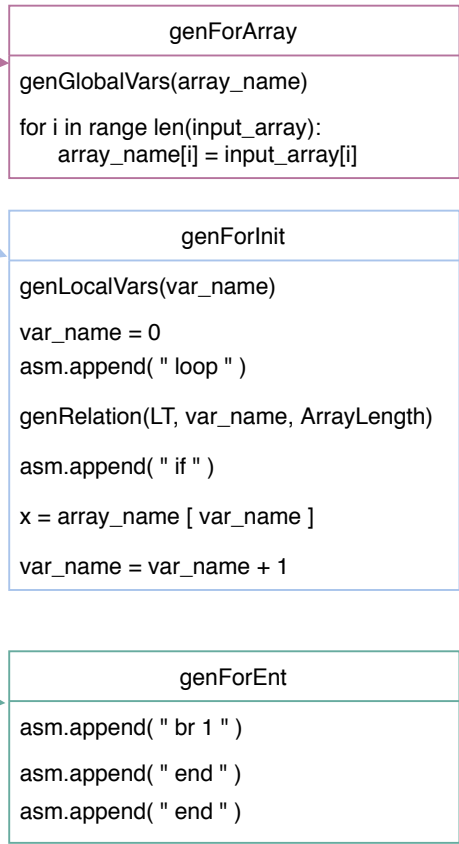


P0

Return Value



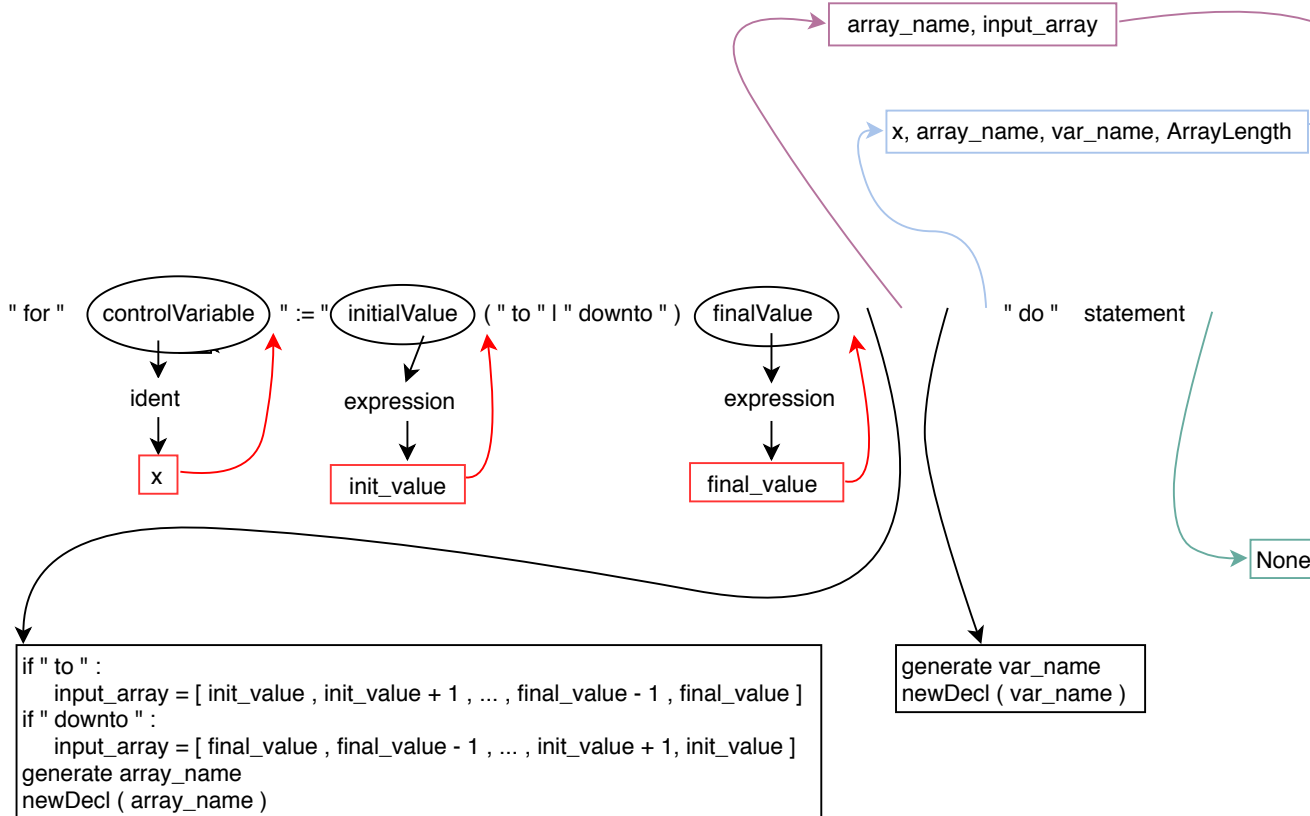
CGwat



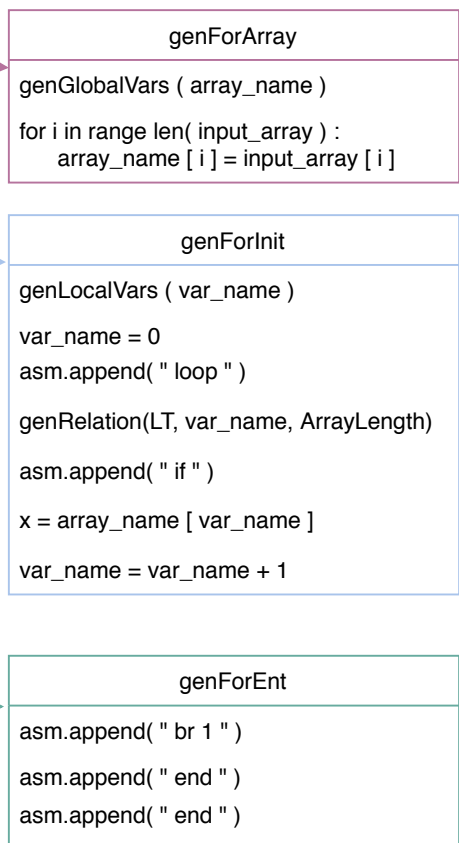
CODE

```
(module
(import "P0lib" "write" (func $write (param i32)))
(import "P0lib" "writeln" (func $writeln))
(import "P0lib" "read" (func $read (result i32)))
(global $x (mut i32) i32.const 0)
(func $program
(local $counter_0 i32)
i32.const 0
i32.const 1
i32.store
i32.const 4
i32.const 3
i32.store
i32.const 0
i32.const 0
local.set $counter_0
loop
local.get $counter_0
i32.const 2
i32.lt_s
if
local.get $counter_0
i32.const 4
i32.mul
i32.const 0
i32.add
i32.load
global.set $x
local.get $counter_0
i32.const 1
i32.add
local.set $counter_0
global.get $x
call $write
br 1
end
end
call $writeln
)
(memory 1)
(start $program)
)
```

P0



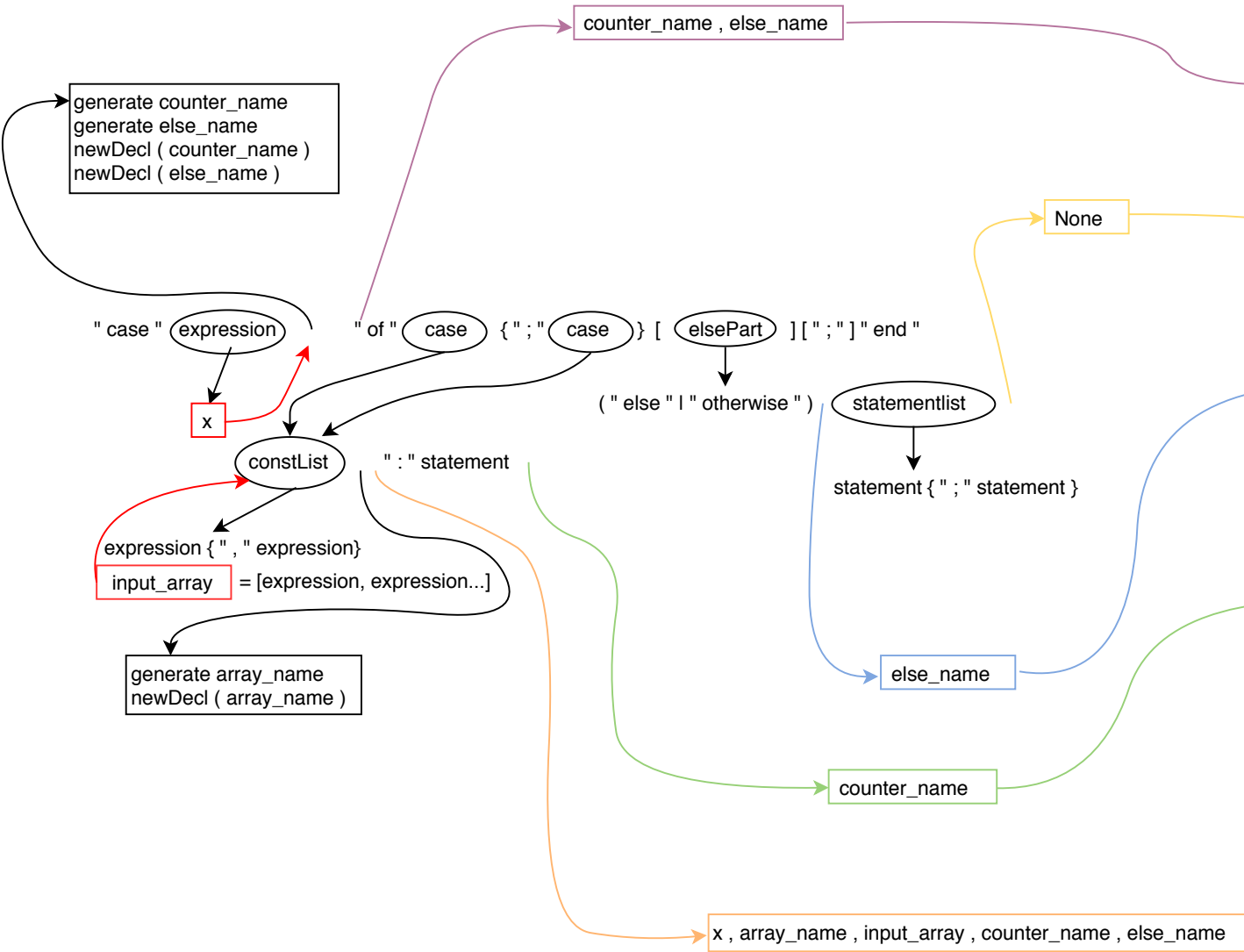
CGwat



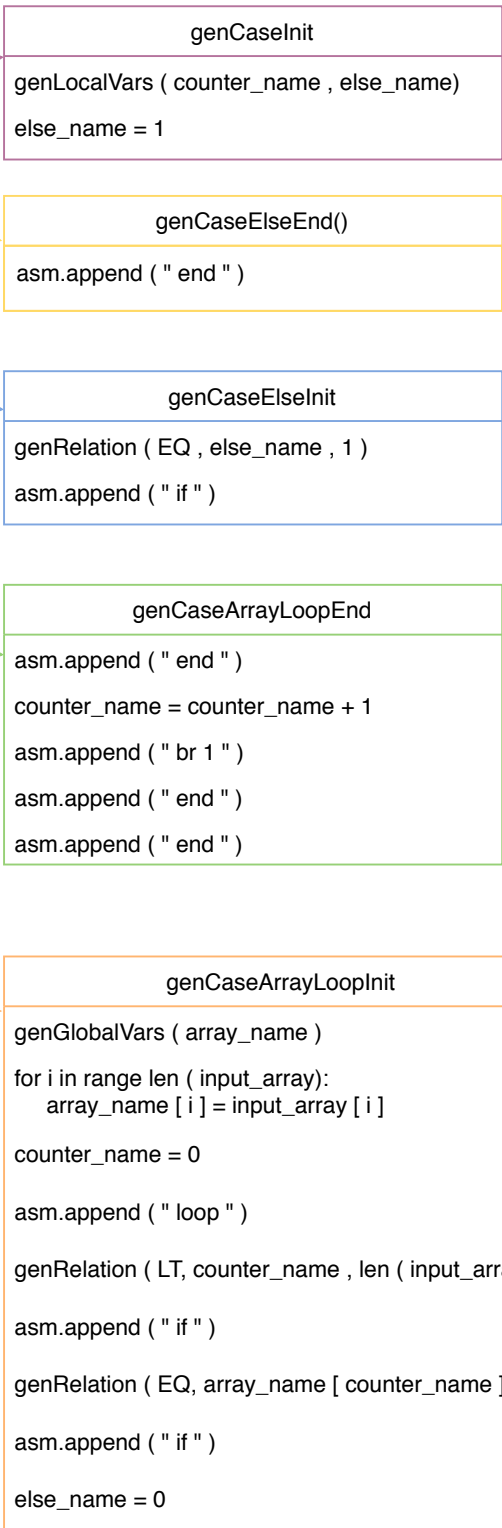
CODE

```
(module
(import "P0lib" "write" (func $write (param i32)))
(import "P0lib" "writeln" (func $writeln))
(import "P0lib" "read" (func $read (result i32)))
(global $x (mut i32) i32.const 0)
(func $program
(local $counter_0 i32)
i32.const 0
i32.const 1
i32.store
i32.const 4
i32.const 2
i32.store
i32.const 0
i32.const 0
local.set $counter_0
loop
local.get $counter_0
i32.const 2
i32.lt_s
if
local.get $counter_0
i32.const 4
i32.mul
i32.const 0
i32.add
i32.load
global.set $x
local.get $counter_0
i32.const 1
i32.add
local.set $counter_0
global.get $x
call $write
br 1
end
end
call $writeln
)
(memory 1)
(start $program)
)
```

P0



CGwat



CODE

```
(func $program
(local $counter_0 i32)
(local $else_0 i32)
i32.const 3
global.set $x
i32.const 1
local.set $else_0
i32.const 0
i32.const 3
i32.store
i32.const 0
local.set $counter_0
loop
local.get $counter_0
i32.const 1
i32.lt_s
if
local.get $counter_0
i32.const 4
i32.mul
i32.const 0
i32.add
i32.load
global.get $x
i32.eq
if
i32.const 0
local.set $else_0
call $writeln
end
local.get $counter_0
i32.const 1
i32.add
local.set $counter_0
br 1
end
end
i32.const 4
i32.const 2
i32.store
i32.const 0
local.set $counter_0
loop
local.get $counter_0
i32.const 1
i32.lt_s
if
local.get $counter_0
i32.const 4
i32.mul
i32.const 4
i32.add
i32.load
global.get $x
i32.eq
if
i32.const 0
local.set $else_0
call $writeln
end
local.get $counter_0
i32.const 1
i32.add
local.set $counter_0
br 1
end
end
local.get $else_0
i32.const 1
i32.eq
if
global.get $x
call $write
end
)
)
```