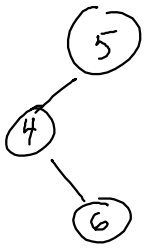
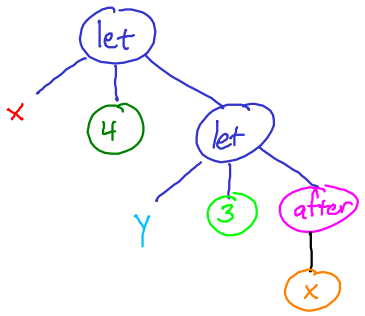

5
4
6

Auklet

<expr> ::= 0 | 1 | -1 | 2 | -2 | ...
| <expr> + <expr> | <expr> - <expr> | <expr> * <expr>
| after(<expr>) | before(<expr>)
| let <var> = <expr> in <expr> | <var>

Auklet	OCaml	AST	x86-64 assembly
5	EInt(5)	5	mov rax, 5
after(3)	EAfter(EInt(3))	after → 3	mov rax, 3 inc rax

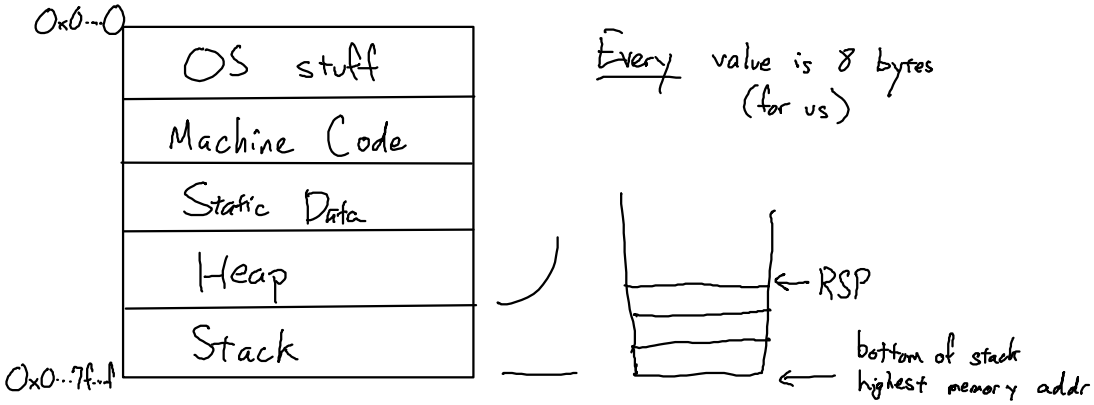
```
let x = 4 in
let y = 3 in
after(x)
```




```
mov rax, 4
mov [rsp-8], rax
mov rax, 3
mov [rsp-16], rax
mov rax, [rsp-8]
inc rax
```

mov %rax, -8(%rsp)

NOT TO SCALE



Every value is 8 bytes (for us)

```
let rec compile_expr (env:environment) (e:expr) : instruction list =
  match e with
  | EInt(n) -> [AsmMov(ArgRegister RAX, ArgConstant n)]
      [mov rax, n]
  | EAfter(e') ->
      compile_expr env e' @[inc rax]
  | EVar(x) ->
      let arg = lookup env x in
      [mov rax, arg]
```

when executed, the answer to e will be stored in RAX

$(-24, \{"x" \mapsto [rsp-8], "y" \mapsto [rsp-16]\})$

```
type environment = int * argument StringMap.t;;
```

pair<int, Dictionary<string, argument>*>