```
(- (+ 2 input) 1)
```

```
mov [rsp - 8], rdi

mov rax, 5
mov [rsp - 16], rax
mov rax, [rsp - 8]
mov [rsp - 24], rax
and rax, [rsp - 16]
and rax, 1
cmp rax, 1
jne near error_non_int
mov rax, [rsp - 24]
sub rax, 1
add rax, [rsp - 16]
jo near overflow_check
mov [rsp - 16], rax
mov rax, 3
mov [rsp - 24], rax
and rax, [rsp - 16]
and rax, 1
cmp rax, 1
jne near error_non_int
mov rax, [rsp - 24]
sub rax, 1
mov [rsp - 24], rax
mov rax, [rsp - 16]
sub rax, [rsp - 24]
jo near overflow_check
```

Can we get rid of all error checks in the generated code if we type-check first?

A: Yes
B: No

Say we implement `calc_type : expr * typ_env -> typ`. Where do we use it?
What do `typ` and `typ_env` look like?

```
(def (g y)
  (+ y 1))

(def (f x)
  (+ (g (+ x 2)) 3))

(def (main input)
  (f (+ input 4))
```

```
(def (g y       )
  (+ y 1))

(def (f x       )
  (+ (g (+ x 2)) 3))

(def (main input        )
  (f (+ input 4))
```