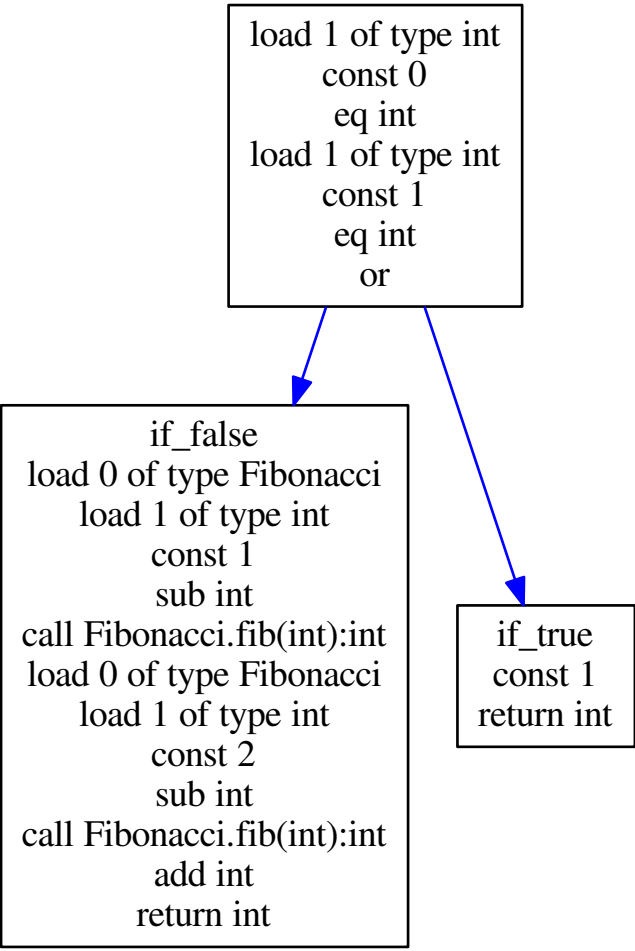


load 1 of type int
const 0
eq int
load 1 of type int
const 1
eq int
or



```
graph TD; Entry["load 1 of type int<br/>const 0<br/>eq int<br/>load 1 of type int<br/>const 1<br/>eq int<br/>or"] --> IfFalse["if_false<br/>load 0 of type Fibonacci<br/>load 1 of type int<br/>const 1<br/>sub int<br/>call Fibonacci.fib(int):int<br/>load 0 of type Fibonacci<br/>load 1 of type int<br/>const 2<br/>sub int<br/>call Fibonacci.fib(int):int<br/>add int<br/>return int"]; Entry --> IfTrue["if_true<br/>const 1<br/>return int"];
```

if_false
load 0 of type Fibonacci
load 1 of type int
const 1
sub int
call Fibonacci.fib(int):int
load 0 of type Fibonacci
load 1 of type int
const 2
sub int
call Fibonacci.fib(int):int
add int
return int

if_true
const 1
return int