```
def TestHalt(G, x):
    def Q(z):
        G(x)
        return 0
    return P(Q, x, 0)
```

```
def foo(x):
    if x > 3:
        while True: pass
    else:
        return None
```

```
def bar(x):
    if x > 0:
        while True: pass
    else:
        return None
```

TestHalt(foo, 10) ✓

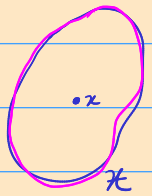
TestHalt(foo, 0) ✗

```
def TestHalt(Q, x):
    """Returns if Q(x) halts or not."""

    def F(y):
        loop

    def G(y):
        if y == x:
            Q(x)
        if y ≠ x:
            loop

    return not P(F, G)
```




$\mathcal{H}(looper) = \phi$

If $Q(x)$ halts, $\mathcal{H}(G) = \{x\}$

If $Q(x)$ loops, $\mathcal{H}(G) = \phi$

$\mathcal{L}(looper) = \mathcal{H}$

$\mathcal{L}(G) = \mathcal{H}$ if $Q(x)$ halts

$\mathcal{L}(G) \neq \mathcal{H}$ if $Q(x)$ loops

```
def looper(x):
    while True: pass
```

$\forall x \in \mathcal{H}$, looper(x) does not halt.

P(F, G) == true if Q(x) halts

P(F, G) == False if Q(x) loops

P(M, looper) == true

P(M, looper) == false