Exercise 1.18. Using the results of exercises 1.16 and 1.17, devise a procedure that generates an iterative process for multiplying two integers in terms of adding, doubling, and halving and uses a logarithmic number of steps.[1]

Answer. The way to capture the intuition of designing a procedure that generates an iterative process here is much similar to that of exercise 1.17. Once we grasp the technique of defining an *invariant quantity*, that is, the prediction of how the process evolves and the means of keeping the value of the combination of some particular elements remained unchanged from state to state, the designing of such a procedure becomes much more easier. Therefore, we can write down our `fast-mult` to generate an iterative process as the following:

```
(define (fast-mult a b)
  (mult-iter a b 0))

(define (mult-iter a b product)
  (cond ((= b 0) product)
        ((even? b) (mult-iter (double a) (halve b) product))
        (else (mult-iter a (- b 1) (+ product a)))))
```

*. Creative Commons  2013, Lawrence R. Amlord(颜世敏).

1. This algorithm, which is sometimes known as the "Russian peasant method" of multiplication, is ancient. Examples of its use are found in the Rhind Papyrus, one of the two oldest mathematical documents in existence, written about 1700 B.C. (and copied from an even older document) by an Egyptian scribe named A'h-mose