Exercise 1.10.

The following procedure computes a mathematical function called Ackermann's function.

```
(define (A x y)
 (cond ((= y 0) 0)
       ((= x 0) (* 2 y))
       ((= y 1) 2)
       (else (A (- x 1)
                (A x (- y 1))))))
```

What are the values of the following expressions?

Solution

Using substitution (or a Scheme REPL) we can see that

(A 1 10) = **1024**
(A 2 4) = **65536**
(A 3 3) = **65536**

Then we are supposed to define the following functions.

```
(define (f n) (A 0 n))

(define (g n) (A 1 n))

(define (h n) (A 2 n))
```

For functions f and g we can simply calculate the values and derive the pattern

(f 0) = 0	(f 1) = 2	(f 2) = 4
(f 3) = 6	(f 4) = 8	(f 5) = 10

Which is function $\boldsymbol{2n}$

(f 0) = 0	(f 1) = 2	(f 2) = 4
(f 3) = 8	(f 4) = 16	

Which is function $\boldsymbol{2^n}$

The last function is slightly more difficult so I will have to use substitution. Apologies for the pseudo-formal notation and for mixing up mathematical and notation and code in Scheme. This is used to try to describe my approach.

```
(define (A x y)
  (cond ((= y 0) 0)
        ((= x 0) (* 2 y))
        ((= y 1) 2)
        (else (A (- x 1)
                 (A x (- y 1))))))
```

I am not expanding y to keep the investigation general. In the first iteration (except for special cases of (h 0) and (h 1)), we only need to further work with the else clause. Therefore the further expansion of the else branch looks like this:

(A 1 (A 2 (- y 1)))

This function was basically introduced in previous example as function g, where it was defined as (define (g n) (A 1 n)).

Therefore we can use simple mathematical substitution and put (A 2 (- y 1)) = n and we get (A 1 n) which is exactly function g from previous example. As mentioned above, g was mathematical function 2^n.

After applying n = (A 2 (- y 1)) from our previous substitution we can describe the function as

(A 2 y) => $2^{(\mathrm{A}\ 2\ (-\mathrm{y}\ 1))}$ => $2^{2^{(\mathrm{A}\ 2\ (-\mathrm{y}\ 2))}}$

Function h is therefore defined as:
(h 0) = 0
(h 1) = 2
(h 2) = 2^2
(h 3) = 2^{2^2} = 16
(h 4) = $2^{2^{2^2}}$ = 2^{16} = 65536

Or in other words (h n) = $2^{(h\ (-n\ 1))}$ with the exception of (h 0) and (h 1) where the branches that do not use recursion of Ackermann function are used.