Exercício 1

a) A = f(X) = $x^4 + 4$

```
λx : (x⁴ + 4)
```

b) B = f(a, b) = a + b

```
λ a, b: a + b
```

c) C= f(x) = x^(-2)

```
λx : x**(-2)
```

d) D = f(x) = x * x^(-1)

```
λx : x * x^(-1)     (λx : 1, para todo x ≠ 0)
```

Exercício 2

a) xy == yx

```
x e y são independentes em ambos os lados e as
funções não são alpha-equivalentes
```

b) λx.x(λy.xy) == λz.z(λx.zx)

```
- λx.x(λy.xy) {z/x} -> λz.z(λy.zy)
- λz.z(λy.zy) {x/y} -> λz.z(λx.zx)

x, y e z são dependentes em ambas e as funções
são alpha-equivalentes
```

c) ((λx.x(λy.xyzy)x)xy) == ((λy.y(λz.yzwz)y)yx)

```
   - ((λx.x(λy.xyzy)x)xy) {y/x} -> ((λx.x(λy.xyzy)x){y/x} (x) {y/x} (y){y/x})
        -  ((λx.x(λy.xyzy)x){y/x}  (x)  {y/x}  (y){y/x})  -
> ((λx.x(λy.xyzy)x){y/x} xy)
   - ((λx.x(λy.xyzy)x)[y -> y']{y/x} xy) -> ((λx.x(λy'.xy'zy')x){y/x} xy)
   - ((λy.y(λy'.yy'zy')y) xy) -> ((λy.y(λy'.yy'zy')y) xy){z/y'}
   - ((λy.y(λy'.yy'zy')y) xy){z/y'} -> ((λy.y(λy'.yy'zy')y) xy)[z -
> z']{z/y'}
          -  ((λy.y(λy'.yy'zy')y)  xy)[z  ->  w]{z/y'}  -
> ((λy.y(λy'.yy'wy')y) xy){z/y'}
   - ((λy.(y(λy'.(yy'wy'))y)) xy){z/y'} -> ((λy.(y(λz.(yzwz))y)) xy)

   y e x da segunda e os últimos x e y da primeira são independentes, e enquanto
     as demais são dependentes. As funções não são alpha-
equivalentes
```

Exercício 3

```
(x(λy.xy))[x→yz]

- x[x->yz] (λy.xy)[x->yz] -> yz (λy.xy){w/y}[x->yz]
- yz (λy.xy){w/y}[x->yz] -> yz (λw.xw)[x->yz]
- yz (λw.xw)[x->yz] -> yz(λw.yzw)
```

Exercício 4

a) (λx.(λy.y*y-(λz.z+x)4)3)2

```
(λy.y*y-(λz.z+2)4)3
3*3-(λz.z+2)4
3*3-4+2
9 - 4 + 2 = 7
```

b) (λx.x+ (λy.y*y)b)a

```
(λx.(x+ (λy.y*y)b)a)
a + (λy.y*y)b
a + (b*b)
a + b²
```

c (λx.(λy.x + (λx.8)-y)6)5 (λy.5 + (λx.8)-y)6 5 + (λx.8)-6 5 + 8 13

d) ((λx.(λy.x+y))3)7 (λy.3+y)7 (λy.3+y)7 3 + 7 10

Exercício 5

a) (λx.+x1)

```
(+x1)[x -> 2]
+ 2 1
3
```

b) (λx.xx)(λx.xx)

```
xx[x -> λx.xx]
(λx.xx)(λx.xx)
```

c) (λx.x(xy))(λu.u)

```
x(xy)[x -> (λu.u)]
x[x -> (λu.u)] (xy)[x -> (λu.u)]
(λu.u)(λu.u)y
(λu.u) u[u -> y]
(λu.u)y
```

```
u[u -> y]
y
```

d) (λy.(λx.y*y+x))z

```
(λx.y*y+x)[y -> z]
(λx.z*z+x)
```

e) (λx.((λy.(yx))(λi.i)))(λp.λq.p)

```
(λx.((λy.(yx))(λi.i)))(λp.(λq.p))
((λy.(yx))(λi.i)) [x -> (λp.(λq.p))]
(λy.(y(λp.(λq.p))))(λi.i)
(y(λp.(λq.p))) [y -> (λi.i)]
(λi.i)(λp.(λq.p))
i [i -> (λp.(λq.p))]
(λp.(λq.p))
```

f) (λx.x)((λy.(λx.xy))x)

```
x [x -> ((λy.(λx.xy))x)]
(λy.(λx.xy))x
(λx.xy)[y -> x]
λx.xx
```

g) (λx.xx)(λy.y)

```
xx [x -> (λy.y)]
(λy.y)(λy.y)
y[y -> (λy.y)]
λy.y
```

Exercício 6

Exercício 7

Representação para números inteiros:

```
n ≡ λf.λx. f^n x
```

Exemplos

```
0 ≡ λf.λx. x
1 ≡ λf.λx. f x
2 ≡ λf.λx. f (f x) = λf.λx. f^2 x
3 ≡ λf.λx. f (f(x)) = λf.λx. f^3 x
4 ≡ λf.λx. f (f(f(x))) = λf.λx. f^4 x
5 ≡ λf.λx. f (f(f(f(x)))) = λf.λx. f^5 x
6 ≡ λf.λx. f (f(f(f(f(x))))) = λf.λx. f^6 x
```

```
 7 ≡ λf.λx. f (f(f(f(f(f(x)))))) = λf.λx. f^7 x
 8 ≡ λf.λx. f (f(f(f(f(f(f(x))))))) = λf.λx. f^8 x
 9 ≡ λf.λx. f (f(f(f(f(f(f(f(x)))))))) = λf.λx. f^9 x
10 ≡ λf.λx. f (f(f(f(f(f(f(f(f(x))))))))) = λf.λx. f^10 x
```

SUCC 0:

$SUCC := \lambda n.\lambda f.\lambda x.(f\ (n\ f\ x))$

$0 := \lambda f_0.\lambda x_0.x_0$

$\hookrightarrow (\lambda n.\lambda f.\lambda x.(f\ (n\ f\ x)))\ 0$

$\hookrightarrow \lambda f.\lambda x.(f\ (0\ f\ x))$

$\hookrightarrow \lambda f.\lambda x.(f\ ((\lambda f_0.\lambda x_0.x_0)\ f\ x))$

$\hookrightarrow \lambda f.\lambda x.(f(\lambda x_0.x_0)\ x)$

$\hookrightarrow \lambda f.\lambda x.f\ x = 1$ //

SUCC 1:

$1 := \lambda f_1.\lambda x_1.f_1\ x_1$

$\hookrightarrow \lambda f.\lambda x.(f\ (\lambda f_1.\lambda x_1.(f_1 x_1)\ f\ x))$

$\hookrightarrow \lambda f.\lambda x.(f\ (\lambda x_1.(f\ x_1)\ x))$

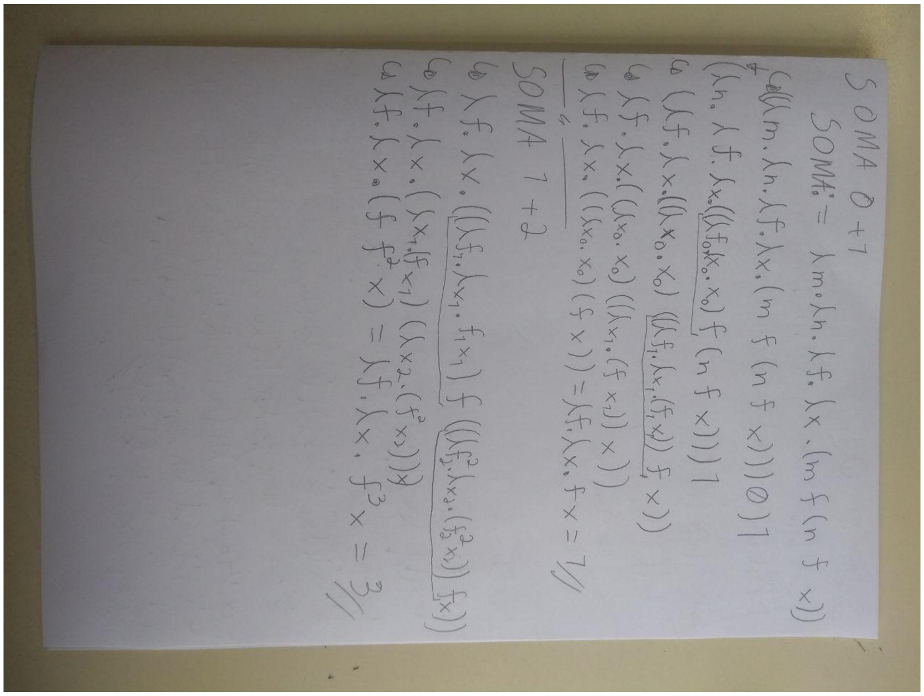
$\hookrightarrow \lambda f.\lambda x.(f(f(x))) = \lambda f x.f^2 x = 2$ //

SUCC 2:

$\hookrightarrow \lambda f.\lambda x.(f\ (\lambda f_2.\lambda x_2.(f_2^2 x_2))f\ x)$

$\hookrightarrow \lambda f.\lambda x.(f\ (\lambda x_2.f^2 x_2)\ x)$

$\hookrightarrow \lambda f.\lambda x.(f\ (f^2 x)) = \lambda f x.f^3 x = 3$ //



SOMA 0 + 1

$SOMA := \lambda m.\lambda n.\lambda f.\lambda x.(m\ f\ (n\ f\ x))$

$\hookrightarrow ((\lambda m.\lambda n.\lambda f.\lambda x.(m\ f\ (n\ f\ x)))\ 0)\ 1$

$(\lambda n.\lambda f.\lambda x.((\lambda f_0.\lambda x_0.x_0)\ f\ (n\ f\ x)))\ 1$

$\hookrightarrow (\lambda f.\lambda x.((\lambda x_0.x_0)\ ((\lambda f_1.\lambda x_1.(f_1\ x_1))\ f\ x)))$

$\hookrightarrow \lambda f.\lambda x.((\lambda x_0.x_0)\ ((\lambda x_1.(f\ x_1))\ x))$

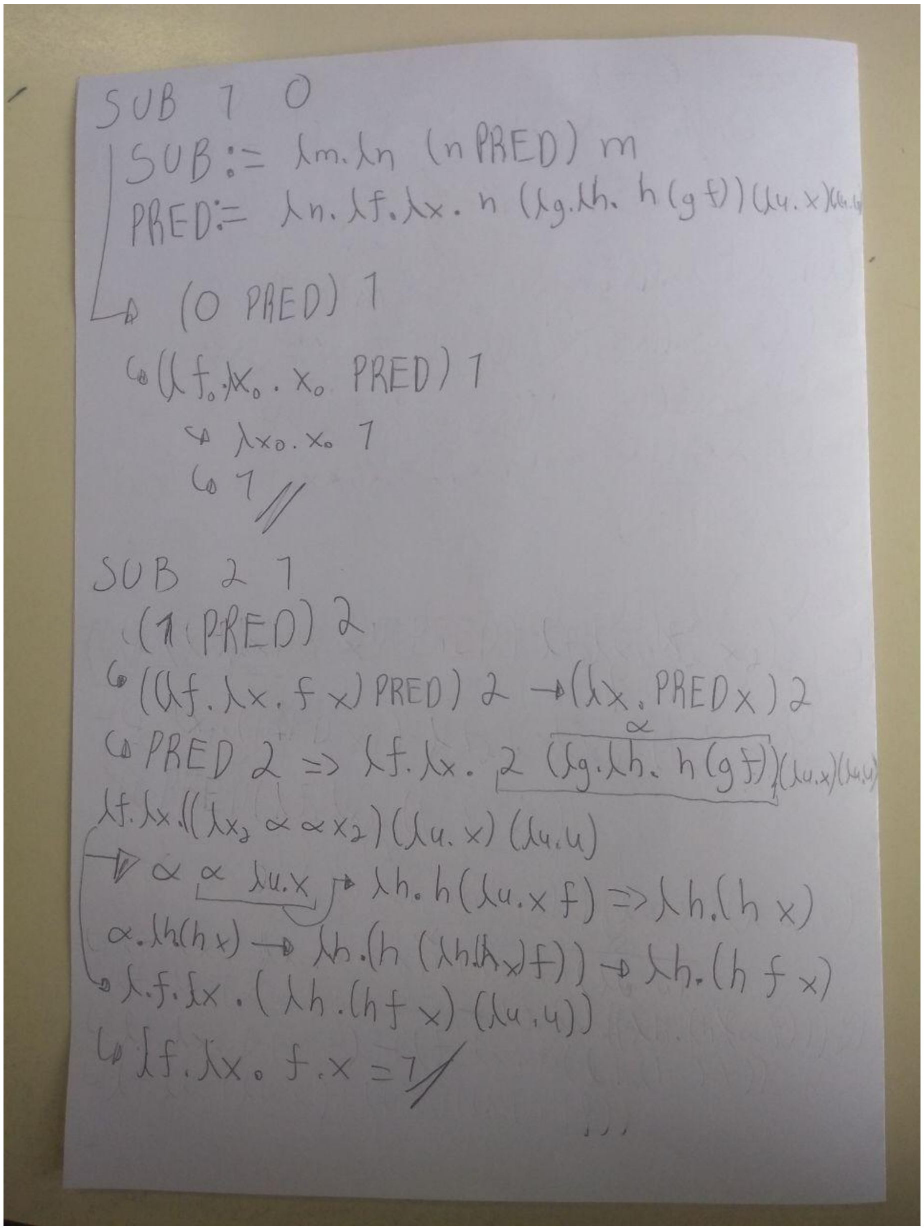
$\hookrightarrow \lambda f.\lambda x.((\lambda x_0.x_0)\ (f\ x)) = \lambda f.\lambda x.f\ x = 1$ //

SOMA 1 + 2

$\hookrightarrow \lambda f.\lambda x.((\lambda f_1.\lambda x_1.f_1\ x_1)\ f\ (((\lambda f_2.\lambda x_2.(f_2^2\ x_2))\ f\ x))$

$\hookrightarrow \lambda f.\lambda x.(\lambda x_1.(f\ x_1))\ (\lambda x_2.(f^2\ x_2))\ x)$

$\hookrightarrow \lambda f.\lambda x.(f\ f^2\ x) = \lambda f.\lambda x.\ f^3\ x = 3$ //



SUB 1 0

$SUB := \lambda m.\lambda n\ (n\ PRED)\ m$

$PRED := \lambda n.\lambda f.\lambda x.\ n\ (\lambda g.\lambda h.\ h\ (g\ f))\ (\lambda u.x)\ (\lambda u.u)$

$\hookrightarrow (0\ PRED)\ 1$

$\hookrightarrow ((\lambda f_0.\lambda x_0.\ x_0\ PRED)\ 1$

$\hookrightarrow \lambda x_0.x_0\ 1$

$\hookrightarrow 1$ //

SUB 2 1

$(1\ PRED)\ 2$

$\hookrightarrow ((\lambda f.\lambda x.\ f\ x)\ PRED)\ 2 \rightarrow (\lambda x.\ PRED\ x)\ 2$

$\hookrightarrow PRED\ 2 \Rightarrow \lambda f.\lambda x.\ 2\ \overset{\alpha}{\boxed{(\lambda g.\lambda h.\ h\ (g\ f))}}\ (\lambda u.x)\ (\lambda u.u)$

$\lambda f.\lambda x.((\lambda x_2\ \alpha\ \alpha\ x_2)\ (\lambda u.x)\ (\lambda u.u)$

$\hookrightarrow \alpha\ \alpha\ \lambda u.x \rightarrow \lambda h.\ h(\lambda u.x\ f) \Rightarrow \lambda h.(h\ x)$

$\alpha.\lambda h(h\ x) \rightarrow \lambda h.(h\ (\lambda h(h\ x)\ f)) \rightarrow \lambda h.(h\ f\ x)$

$\hookrightarrow \lambda f.\lambda x.\ (\lambda h.(h\ f\ x)\ (\lambda u.u))$

$\hookrightarrow \lambda f.\lambda x.\ f.x = 1$ //

Exercício 8

O Combinador Y é uma forma de criar recursividade em funções anônimas, útil em cálculo lambda por uma função não precisar chamar a si mesma.

É basicamente uma função que recebe de argumentos uma função e um valor, e essa primeira função é passada junto de um argumento para uma segunda função que chama a primeira, passando a própria primeira como argumento. A primeira função tem o papel de parar a sequência ou de chamar a sia própria (recebida como argumento) novamente.