Exercise 1
JavaScript: x => x
Lambda Calculus: λx.x

Exercise 2

1. **λx.x**
   Sub x = a:
   λa.a
   Therefore, answer is **b) λa.a**

2. **λxy.yx**
   = λx(λy.yx)
   Sub x = a, y = b:
   λa(λb.ba)
   Therefore, answer is **b) λa(λb.ba)**

3. **λxy.xz**
   Sub x = m, y = n:
   λmn.mz
   Therefore, answer is **b) λmn.mz**

Exercise 3

1. **(λx.x)y**
   = x [x := y]
   = y
2. **λx.xx**
   This is already the normal form
3. **(λz.zz)(λy.yy)**
   = zz [z := (λy.yy)]
   = (λy.yy)(λy.yy)
   = yy [y := (λy.yy)]
   = (λy.yy)(λy.yy)
   = ...
   This will continue infinitely.
   Therefore, this is a divergence.
4. **(λx.xx)y**
   = xx [x := y]
   = yy

Exercise 4

1. **(λy.zy)a**
   = zy [y := a] = za
2. **(λx.x)(λx.x)**
   = x [x := λx.x] = λx.x
3. **(λx.xy)(λx.xx)**
   = xy [x := λx.xx] = (λx.xx)y
   = xx [x := y] = yy
4. **(λz.z)(λa.aa)(λz.zb)**
   = ((λz.z)(λa.aa))(λz.zb)
   = (z [z := λa.aa])(λz.zb)
   = (λa.aa)(λz.zb)
   = aa [a := λz.zb]
   = (λz.zb)(λz.zb)
   = zb [z := λz.zb]
   = (λz.zb)b
   = zb [z := b]
   = bb

Exercise 5

1. **λx.zx**
   = z
2. **λx.xz**
   This expression cannot be simplified further
3. **(λx.bx)(λy.ay)**
   = ba

Exercise 6

1. **λx.xxx**
   This lambda expression takes in a parameter x and only uses x in its output. Therefore, there are no free variables. Hence, it is a combinator.
2. **λxy.zx**
   Since z is not a parameter but is used in the output, there is a free variable. Therefore, the expression is not a combinator.
3. **λxyz.xy(zx)**
   This lambda expression takes in parameters x, y and z and only uses all x, y and z in its output. Therefore, there are no free variables. Hence, it is a combinator.

4. **λxyz.xy(zxy)**
   This lambda expression takes in parameters x, y and z and only uses all x, y and z in its output. Therefore, there are no free variables. Hence, it is a combinator.

Exercise 7

$Y = \lambda f.(\lambda x.f(xx))(\lambda x.f(xx))$

$Y\ g = (\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))(g)$

$= (\lambda x.f(xx))(\lambda x.f(xx))\ [f := g]$

$= (\lambda x.g(xx))(\lambda x.g(xx))$

$= g(xx)\ [x := \lambda x.g(xx)]$

$= g((\lambda x.g(xx))(\lambda x.g(xx)))$

$= g((\lambda f.(\lambda x.f(xx))(\lambda x.f(xx)))g)$

$= g(Y\ g)$

Exercise 8

TRUE = λxy.x

FALSE = λxy.y

IF = λbtf. b t f

AND = λxy. IF x y FALSE

OR = λxy. IF x TRUE y

NOT = λx. IF x FALSE TRUE

1. **NOT FALSE**
   = (λx. IF x FALSE TRUE)(FALSE)
   = IF x FALSE TRUE [x := FALSE]
   = IF FALSE FALSE TRUE
   = (λbtf. b t f)(FALSE FALSE TRUE)
   = b t f [b := FALSE, t := FALSE, f := TRUE]
   = FALSE FALSE TRUE
   = (λxy.y) (FALSE TRUE)
   = y [x := FALSE, y := TRUE]
   = **TRUE**

2. **IF (OR TRUE FALSE)**
   = λbtf. b t f (OR TRUE FALSE)
   = b t f (b := OR, t := TRUE, f := FALSE)
   = OR TRUE FALSE
   = λxy. IF x TRUE y (TRUE FALSE)
   = IF x TRUE y (x := TRUE, y := FALSE)
   = IF TRUE TRUE FALSE
   = λbtf. b t f (TRUE TRUE FALSE)
   = b t f (b := TRUE, t := TRUE, f := FALSE)
   = TRUE TRUE FALSE
   = λxy.x (TRUE FALSE)
   = x [x := TRUE, y := FALSE]
   = **TRUE**

3. **IF (AND TRUE TRUE)**
   = λbtf. b t f (AND TRUE TRUE)
   = b t f (b := AND, t := TRUE, f := TRUE)
   = AND TRUE TRUE
   = λxy. IF x y FALSE (TRUE TRUE)
   = IF x y FALSE (x := TRUE, y := TRUE)
   = IF TRUE TRUE FALSE
   = λbtf. b t f (TRUE TRUE FALSE)
   = b t f (b := TRUE, t := TRUE, f := FALSE)
   = TRUE TRUE FALSE
   = λxy.x (TRUE FALSE)
   = x [x := TRUE, y := FALSE]
   = **TRUE**