1. Beta-reduce the following expressions to their normal form:
   1. (λa λy . y a) (z z)

in λy.ya

🡪 β λy.y(z z)

* 1. (λx λy . (x y)) (λz.y)

in λy . (x y)

🡪 β λy.(λz.y) y

in (λz.y)

🡪 β λy.y

* 1. (λx.(x x)) (λy.(y y))

// This one works similarly to the Y combinator, it will continuously spit out:

(λy.(y y)) (λy.(y y))

(λx.(x x)) (λy.(y y))

in (x x)

🡪 β (λy.(y y)) (λy.(y y))

// Then it will continue to output (λy.(y y)) every time

* 1. K x y

≡(λxy.x)xy

α= (λxy.x)ab

in λy.x

🡪 β (λy.a)b

in a

🡪 β a

* 1. S K

≡(λxyz.xz(yz))K

in (λyz.xz(yz))

🡪 β (λyz.Kz(yz))

// Partial Evaluation

// The expanded form of S K is:

(λxyz.xz(yz))( λxy.x)

* 1. (S K) y y z

// As seen in part e, S K ≡(λxyz.xz(yz))( λxy.x)

(λxyz.xz(yz))( λxy.x) y y z

in (λyz.xz(yz))(λxy.x)

🡪 β (λyz.yz(yz))(λyy.y) y z

in (λz.yz(yz))(λyy.y)

🡪 β in (λz.yz(yz))(λyy.y) z

in (yz(yz))(λyy.y)

🡪 β (yz(yz))(λyy.y)

* 1. K’ y y z

≡(λxy.y) y y z

in (λy.y)

🡪 β (λy.y) y z

in y

🡪 β y z // Since K’ takes the 2nd of *two* values, the third stays

1. What is the normal form of: S (K S) (K I)

≡ (λxyz.xz(yz)) (K S) (K I)

in (λyz.xz(yz))

🡪 β (λyz.(K S)z(yz)) (K I)

in (λz.(K S)z(yz))

🡪 β (λz.(K S)z((K I) z))

1. Prove the following equivalencies by reducing each side to its normal form.
   1. I = S K K

S K K

≡ (λxyz.xz(yz)) K K

in (λyz.xz(yz))

🡪 β (λyz.Kz(yz)) K

in (λz.Kz(yz))

🡪 β (λz.Kz(Kz)) S K K ≡ (λz.Kz(Kz))

K z

≡(λxy.x) z

in (λy.x)

🡪 β (λy.x) K z ≡ (λy.x)

λz.Kz(Kz)

≡λz. (λy.x) (λy.x)

🡪 β (λz.z)

α=(λx.x)

≡I

So I = S K K

* 1. S K K = K I I

// As seen in 3a, S K K = I

K I I

≡(λxy.x) I I

In λy.x

🡪 β (λy.I) I

in I

🡪 β I

//So S K K = I and K I I = I which means that S K K = K I I