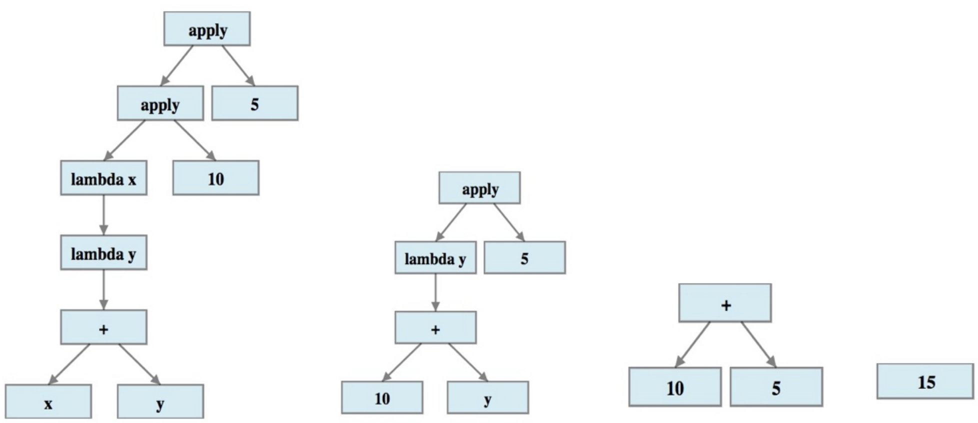
To solve these problems, you need to understand the following concepts:

(1) Lambda-term definition
(2) Parentheses convention
(3) Free Variables
(4) alpha-equivalence
(5) substitutions
(6) beta-reductions

Problem 1: Reduce the following expressions to values:

```
  (((lambda x (lambda y (+ x y))) 10) 5)
= ((lambda y (+ 10 y)) 5)
= (+ 10 5)
= 15
```



Problem 2: For each of the following terms, identify the free variables in each term and identify which terms are closed:

```
t4 = ((lambda x x) x)
FREE: 3rd x

t7 = ((lambda y x) y)
FREE: x, 2nd y
```

Problem 3: Using the terms from above, apply the following substitutions and show the resulting expression

```
  t4[x := t3]
= ((lambda x x) x)[x := (lambda x (x x))]
= ((lambda x x)[x := (lambda x (x x))] x[x := (lambda x (x x))])
= ((lambda x x) (lambda x (x x)))
```

Problem 4: Make all parentheses explicit in the following expressions:

```
λx.xz λy.xy
(λx.xz λy.xy)
(λx.(xz λy.xy))
(λx.(xz (λy.xy)))
(λx.((xz) (λy.xy)))
(λx.((xz) (λy.(xy))))
```

Problem 5. Apply β-reductions to the following λ expressions as much as possible:

```
  (λz.z) (λy.y y)(λx.x a)
= (λy.y y)(λx.x a)
= ((λx.x a) (λx.x a))
= ((λx.x a) a)
= (a a)
```

Problem 6: Do it yourself. Plug in the values of `plus`, `two`, and `three` in (`plus two three`) and then solve as in problem 5.