Software Engineering 1

Regular exam OP3 - 2017

1 Question 1

Given the following lambda program, complete the empty beta reduction steps for this program.

```
(((λf g x y→((f x) y)) (λx y→y)) (λx y→x))
```

1.1 Answer 1

```
(((λf g x y→((f x) y)) (λx y→y)) (λx y→x))
```

```
( ((λf g x y→((f x) y)) (λx y→y)) (λx y→x))
```

```
((λg x y→(( (λx y→y) x) y)) (λx y→x))
```

```
((λg x y→(((λx y→y) x) y)) (λx y→x))
```

```
(λx y→(((λx y→y) x) y))
```

```
(λx y→( ((λx y→y) x) y))
```

```
(λx y→((λy→y) y))
```

```
(λx y→ ((λy→y) y) )
```

```
(λx y→ y )
```

2 Question 2

Given the following lambda calculus program complete the typing derivation for the program.

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→((f x) y))
```

2.1 Answer 2

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→((f x) y))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→((f x) y))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→(( (int→int→int) x) y))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int) → (((int→int→int) x) y) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→(((int→int→int) x) y))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int) → (((int→int→int) x) y) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→(((int→int→int) int ) y))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int) → (((int→int→int) int) y) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→(((int→int→int) int) int ))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→( ((int→int→int) int) int))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→( (int→int) int))
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→ ((int→int) int) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int)→ int )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) (y:int) → int )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int)→ (int→int) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) (x:int) → (int→int) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int))→ (int→int→int) )
```

```
(λ(f:(int→int→int)) (g:(int→int→int)) → (int→int→int) )
```

```
(λ(f:(int→int→int))→ ((int→int→int)→int→int→int) )
```

```
(λ(f:(int→int→int))→((int→int→int)→int→int→int))
```

```
((int→int→int)→(int→int→int)→int→int→int)
```