Software Engineering 1

Lambda Calculus exercises
Typing

1 Exercise 1

Given the following lambda program, complete the typing derivation for this program.

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

1.1 Answer 1

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

2 Exercise 2

Given the following lambda program, complete the typing derivation for this program.

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((f g) l))
```

2.1 Answer 2

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((f g) l))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((f g) l))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((
    ((int→int)→List<int>→List<int>) g) l))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→
    ((((int→int)→List<int>→List<int>) g) l) )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((((int→int)→List<int>→
    List<int>) (int→int) ) l))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→
    ((((int→int)→List<int>→List<int>) (int→int)) l) )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→((((int→int)→List<int>→
    List<int>) (int→int)) List<int> ))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→(
    (((int→int)→List<int>→List<int>) (int→int))  List<int>))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→( (List<int>→List<int>)  List<
    int>))
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→
    ((List<int>→List<int>) List<int>) )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>)→ List<int> )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) (l:List<int>) → List<int> )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int))→ (List<int>→List<int>) )
```

```
(λ(f:((int→int)→List<int>→List<int>)) (g:(int→int)) → (List<int>→List<int>) )
```

```
(λ(f:((int→int)→List<int>→List<int>))→ ((int→int)→List<int>→List<int>) )
```

```
(λ(f:((int→int)→List<int>→List<int>))→((int→int)→List<int>→List<int>))
```

```
(((int→int)→List<int>→List<int>)→(int→int)→List<int>→List<int>)
```

3 Exercise 3

Given the following lambda program, complete the typing derivation for this program.

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(((f
    g) s) l))
```

3.1 Answer 3

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(((f
    g) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(((f g) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(((
    ((int→int→int)→int→List<int>→List<int>) g) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) g) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) (int→int→int)) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) (int→int→int)) s) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) (int→int→int)) int) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) (int→int→int)) int) l))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    (((((int→int→int)→int→List<int>→List<int>) (int→int→int)) int) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→((
    (((int→int→int)→int→List<int>→List<int>) (int→int→int)) int) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→((
    (int→List<int>→List<int>) int) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(
    ((int→List<int>→List<int>) int) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→(
    (List<int>→List<int>) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    ((List<int>→List<int>) List<int>))
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    List<int>)
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) (l:List<int>)→
    List<int>)
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int)→
    (List<int>→List<int>) )
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) (s:int) →
    (List<int>→List<int>) )
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int))→ (int→List<int>→List<int>) )
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>)) (g:(int→int→int)) → (int→List<int>→List<int>) )
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>))→ ((int→int→int)→int→List<int>→List<int>) )
```

```
(λ(f:((int→int→int)→int→List<int>→List<int>))→((int→int→int)→int→List<int>→List<int>))
```

```
(((int→int→int)→int→List<int>→List<int>)→(int→int→int)→int→List<int>→List<int>)
```

4 Exercise 4

Given the following lambda program, complete the typing derivation for this program.

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

4.1 Answer 4

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

5 Exercise 5

Given the following lambda program, complete the typing derivation for this program.

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y
     (x a)))
```

5.1 Answer 5

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y
     (x a)))
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y (x a)))
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y
     (x a)))
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string)) → (y (x a)) )
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y
     ( (int→string)  a)))
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string)) →
     (y ((int→string) a)) )
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(y
     ((int→string) int )))
```

```
(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→
    (y ((int→string) int)))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→(
    (string→int→string) ((int→string) int)))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→((
    string→int→string) ((int→string) int)))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→((
    string→int→string) string))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→
    ((string→int→string) string))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→
    (int→string))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int) (y:(string→int→string))→
    (int→string))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int)→
    ((string→int→string)→int→string))

(λ(f:((string→int→string)→int→string)) (x:(int→string)) (a:int)→
    ((string→int→string)→int→string))

(λ(f:((string→int→string)→int→string)) (x:(int→string))→
    (int→(string→int→string)→int→string))

(λ(f:((string→int→string)→int→string)) (x:(int→string))→(int→(string→int→string)→int→string))

(λ(f:((string→int→string)→int→string))→((int→string)→int→(string→int→string)→int→string))

(λ(f:((string→int→string)→int→string))→((int→string)→int→(string→int→string)→int→string))

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