INFSEN02-1 Exam

Herkansing

1 Question 1

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

```
((((λu w z→((u w) z)) (λf g x→(f x))) (λy→y)) (λy→y))
```

1.1 Answer 1

```
((((λu w z→((u w) z)) (λf g x→(f x))) (λy→y)) (λy→y))
```

```
(( ((λu w z→((u w) z)) (λf g x→(f x))) (λy→y)) (λy→y))
```

```
(((λw z→(( (λf g x→(f x)) w) z)) (λy→y)) (λy→y))
```

```
( ((λw z→(((λf g x→(f x)) w) z)) (λy→y)) (λy→y))
```

```
((λz→(((λf g x→(f x)) (λy→y) ) z)) (λy→y))
```

```
((λz→(((λf g x→(f x)) (λy→y)) z)) (λy→y))
```

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

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

```
((λg x→( (λy→y) x)) (λy→y))
```

```
((λg x→((λy→y) x)) (λy→y))
```

```
(λx→((λy→y) x))
```

```
(λx→ ((λy→y) x) )
```

```
(λx→ x )
```

2 Question 2

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

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→(snd ((t y) x)))
```

2.1 Answer 2

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→(snd ((t y) x)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→(snd ((t y) x)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→(snd ((string→int→Tuple) y) x)
    ))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→(snd (((string→int→Tuple) y) x)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) (((string→int→
    Tuple) y) x)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→
    ((Tuple→int) (((string→int→Tuple) y) x)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) (((string→int→
    Tuple) y) int)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→
    ((Tuple→int) (((string→int→Tuple) y) int)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) (((string→int→
    Tuple) string) int)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) (
    ((string→int→Tuple) string) int)))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) ((int→Tuple) int)
    ))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) ((int→Tuple) int))
    )
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) Tuple))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→((Tuple→int) Tuple))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→int)
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int) (y:string)→int)
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int)→(string→int))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int)) (x:int)→(string→int))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int))→(int→string→int))
```

```
(λ(t:(string→int→Tuple)) (snd:(Tuple→int))→(int→string→int))
```

```
(λ(t:(string→int→Tuple))→ ((Tuple→int)→int→string→int) )
```

```
(λ(t:(string→int→Tuple))→((Tuple→int)→int→string→int))
```

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