INFSEN02-1 Sample exam

The INFDEV@HR Team

1 Question 1

Given the following lambda program, and a series of relevant delta rules, show the beta reductions for this program.

```
(1 + 2)
```

1.1 Relevant delta rules

Integer addition:

```
(λm n→ (λs z→((m s) ((n s) z))))
```

Integer one (1)

```
(λs z→(s z))
```

Integer two (2)

```
(λs z→(s (s z)))
```

1.2 Answer 1 (note: you do not need to write all this detail yourself, it is only included for completeness)

```
(1 + 2)
```

```
((+ 1) 2)
```

```
(( (λm n→ (λs z→((m s) ((n s) z)))) 1) 2)
```

```
(((λm n→ (λs z→((m s) ((n s) z)))) 1) 2)
```

```
(((λm n→ (λs z→((m s) ((n s) z)))) (λs z→(s z))) 2)
```

```
( ((λm n→ (λs z→((m s) ((n s) z)))) (λs z→(s z))) 2)
```

```
((λn s z→(( (λs z→(s z)) s) ((n s) z))) 2)
```

```
((λn s z→(((λs z→(s z)) s) ((n s) z))) 2)
```

```
((λn s z→(((λs z→(s z)) s) ((n s) z))) (λs z→(s (s z))))
```

```
((λn s z→(((λs z→(s z)) s) ((n s) z))) (λs z→(s (s z))))
```

```
(λs z→(((λs z→(s z)) s) (( (λs z→(s (s z))) s) z)))
```

```
(λs z→( ((λs z→(s z)) s) (((λs z→(s (s z))) s) z)))
```

```
(λs z→((λz→(s z)) (((λs z→(s (s z))) s) z)))
```

```
(λs z→((λz→(s z)) (((λs z→(s (s z))) s) z)))
```

```
(λs z→((λz→(s z)) ((λz→(s (s z))) z)))
```

```
(λs z→((λz→(s z)) ((λz→(s (s z))) z)))
```

```
(λs z→((λz→(s z)) (s (s z))))
```

```
(λs z→((λz→(s z)) (s (s z))))
```

```
(λs z→(s (s (s z))))
```

```
(λs z→(s (s (s z))))
```

```
3
```

2 Question 2

Given the following lambda calculus program, and a series of relevant delta rules, give the full typing derivation for the program.

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (((n α) s) z))))
```

2.1 Relevant delta rules

Integer type:

```
(∀α ⇒((α→α)→α→α))
```

2.2 Answer 2 (note: you do not need to write all this detail yourself, it is only included for completeness)

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((m α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒(λ(s:(α→α)) (z:α)→(((Nat α) s) (((n α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) s) (((Nat α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) s) (((Nat α) s) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) (α→α)) (((Nat α) (α→α)) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) (α→α)) (((Nat α) (α→α)) z))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) (α→α)) (((Nat α) (α→α)) α))))
```

```
(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→(((Nat α) (α→α)) (((Nat α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((((∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α) (α→α))
    α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((((∀α ⇒((α→α)→α→α)) α) (α→α)) (((Nat α) (α→α)) α
    ))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((((α→α)→α→α) (α→α)) (((Nat α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) (((Nat α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) (((Nat α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) (((∀α ⇒((α→α)→α→α)) α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) (((∀α ⇒((α→α)→α→α)) α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) ((((α→α)→α→α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) ((((α→α)→α→α) (α→α)) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) ((α→α) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) ((α→α) α))))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) α)))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→((α→α) α)))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→α))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α)) (z:α)→α))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α))→(α→α)))

(λ(m:Nat) (n:Nat)→Λα⇒ (λ(s:(α→α))→(α→α)))

(λ(m:Nat) (n:Nat)→Λα⇒ ((α→α)→α→α))

(λ(m:Nat) (n:Nat)→Λα⇒((α→α)→α→α))

(λ(m:Nat) (n:Nat)→(∀α ⇒((α→α)→α→α)))

(λ(m:Nat) (n:Nat)→(∀α ⇒((α→α)→α→α)))

(λ(m:Nat)→(Nat→(∀α ⇒((α→α)→α→α))))

(λ(m:Nat)→(Nat→(∀α ⇒((α→α)→α→α))))

(Nat→Nat→(∀α ⇒((α→α)→α→α)))
```