INFSEN02-1 Sample exam

The INFDEV@HR Team

1 Exam

1.1 Question 1

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

```
let apply = (λx f→(f x)) in ((apply 3) (λx→(3 + x)))
```

1.2 Relevant delta rules

Integer addition

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

Integer three (3)

```
(λs z→(s (s (s z))))
```

Integer size (6)

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

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

```
let apply = (λx f→(f x)) in ((apply 3) (λx→(3 + x)))
```

```
let apply = (λx f→(f x)) in ((apply 3) (λx→(3 + x)))
```

```
((λapply→((apply 3) (λx→(3 + x)))) (λx f→(f x)))
```

```
((λapply→((apply 3) (λx→(3 + x)))) (λx f→(f x)))
```

```
((( λx f→(f x)) 3) (λx→(3 + x)))
```

```
((((λx f→(f x)) 3) (λx→(3 + x)))
```

```
((((λx f→(f x)) (λs z→(s (s (s z)))) ) (λx→(3 + x)))
```

```
((((λx f→(f x)) (λs z→(s (s (s z))))) (λx→(3 + x)))
```

```
((λf→(f (λs z→(s (s (s z)))) )) (λx→(3 + x)))
```

```
((λf→(f (λs z→(s (s (s z)))))) (λx→(3 + x)))
```

```
( (λx→(3 + x)) (λs z→(s (s (s z)))))
```

```
((λx→(3 + x)) (λs z→(s (s (s z)))))
```

```
(3 + (λs z→(s (s (s z)))) )
```

```
((+ 3) (λs z→(s (s (s z)))))
```

```
(( (λm n→ (λs z→((m s) ((n s) z)))) 3) (λs z→(s (s (s z)))))
```

```
(((λm n→ (λs z→((m s) ((n s) z)))) 3) (λs z→(s (s (s z)))))
```

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

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

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

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

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

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

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

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

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

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

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

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

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

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

```
6
```

1.4 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))))
```

1.5 Relevant delta rules

Boolean type:

```
(∀α ⇒((α→α)→α→α))
```

1.6 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))))
```

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((\ \underline{((\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))\ \alpha)}\ (\alpha\rightarrow\alpha))\ \alpha))))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ ((\ \boxed{((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)}\ (\alpha\rightarrow\alpha))\ \alpha))))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ (\ \underline{(((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)\ (\alpha\rightarrow\alpha))}\ \alpha))))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ (\ \boxed{(\alpha\rightarrow\alpha)}\ \alpha))))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ \underline{((\alpha\rightarrow\alpha)\ \alpha)}))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow((\alpha\rightarrow\alpha)\ \boxed{\alpha})))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow\underline{((\alpha\rightarrow\alpha)\ \alpha)}))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\ (\texttt{z:}\alpha)\rightarrow\boxed{\alpha}))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\underline{(\texttt{z:}\alpha)\rightarrow\underline{\alpha}}))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ (\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\rightarrow\boxed{(\alpha\rightarrow\alpha)}))$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \underline{(\lambda(\texttt{s:}(\alpha\rightarrow\alpha))\rightarrow(\alpha\rightarrow\alpha))})$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\Lambda\alpha\Rightarrow\ \boxed{((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)})$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\underline{\Lambda\alpha\Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha)})$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\boxed{(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))})$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\underline{(\forall\alpha\ \Rightarrow((\alpha\rightarrow\alpha)\rightarrow\alpha\rightarrow\alpha))})$

$(\lambda(\texttt{m:Nat})\ (\texttt{n:Nat})\rightarrow\boxed{\texttt{Nat}})$

$(\lambda(\texttt{m:Nat})\underline{(\texttt{n:Nat})\rightarrow\underline{\texttt{Nat}}})$

$(\lambda(\texttt{m:Nat})\rightarrow\boxed{(\texttt{Nat}\rightarrow\texttt{Nat})})$

$\underline{(\lambda(\texttt{m:Nat})\rightarrow(\texttt{Nat}\rightarrow\texttt{Nat}))}$

$\boxed{(\texttt{Nat}\rightarrow\texttt{Nat}\rightarrow\texttt{Nat})}$