Show one page at a time
Finish review

Question **1**
Correct
Mark 1.00 out of 1.00
Flag question

Would you like to get the default mark?

- a. No thanks
- b. Yes please ✔

Information
Flag question

The next section contains basic questions.

Read each question carefully.

Question **2**
Correct
Mark 0.50 out of 0.50
Flag question

Select the appropriate operators for each list operation:

Append an element to the head of the list: a::b ✔

Append two lists: a ++ b ✔

Question **3**
Partially correct
Mark 0.33 out of 0.50
Flag question

Select the appropriate description for each type defintion:

type Fruit = Apple | Banana | Cherry: Sum ✔

type Tree = Node Tree Int Tree | Nil: Product ✘

type Point = Point Int Int: Product ✔

Question **4**
Correct
Mark 0.50 out of 0.50
Flag question

For each of the following language constructs, select its type:

case ... of — Expression ✔
if ... then ... else — Expression ✔
let ... in — Expression ✔

Question **5**
Correct
Mark 0.50 out of 0.50
Flag question

Match the functions with their corresponding recursion type. Choose Head (non-tail) or Tail.

add a b = if b == 0 then a else add (a+1) (b-1) — Tail ✔
add a b = if b == 0 then a else 1 + add a (b-1) — Head ✔
add a b s = if (b == 0) then s + a else if (a==0) then s + b else add (a-1) (b-1) (s+2) — Tail ✔

Question **6**
Correct
Mark 0.50 out of 0.50
Flag question

Select the types that have Cardinality 2.

- [x] a. Bool ✔
- [ ] b. Maybe
- [x] c. type State = On | Off ✔
- [ ] d. Order

Question **7**

Select all true statements:

Question **7**
Correct
Mark 0.50 out of 0.50
Flag question

Select all **true** statements:

- [ ] a. We can use `local ... in ... end` to declare local variables
- [ ] b. Elm has no concept of local declarations, everything has to be declared globally
- [x] c. Local declarations must end with an expression that can use the function and constant declarations ✔
- [ ] d. Only constants can be declared in local declarations, functions must always be declared globally
- [x] e. We can use `let ... in` to declare local variables ✔

Question **8**
Partially correct
Mark 0.33 out of 0.50
Flag question

Select all the **true** statements about records:

- [x] a. Records are a collection of fields ✔
- [ ] b. Names of records fields are optional
- [x] c. Fields can be accessed using the record["fieldName"] syntax ✘
- [ ] d. In Elm records can't be nested
- [x] e. Records can be pattern matched ✔

Question **9**
Correct
Mark 0.50 out of 0.50
Flag question

What is type inference?

- ( ) a. A way of specifying complex types in functional languages without runtime errors
- (•) b. A language characteristic which means that types are deduced automatically ✔
- ( ) c. A feature of the language to have complex data structures

Question **10**

Partially correct

Mark 0.33 out of 1.00

Flag question

Select the correct option for each code snippet:

```
max fallback l = case l of
   [] -> fallback
   x::xs -> if x > fallback then max x xs else max fallback xs
```

Can crash at runtime ✗

```
max l = case l of
   [x] -> x
   x::xs -> let m = max fallback xs in
      if x > m then x else m
```

Fails to compile ✓

```
max fallback l = case l of
   [] -> fallback
   x::xs -> let m = max fallback xs in
      if x > m then x else m
```

Compiles and never crashes ✗

Question **11**

Correct

Mark 1.00 out of 1.00

Flag question

The inferred signature of the following function is:

```
add2 ax ay bx by = {x = ax + bx, y = ay + by}
```

a. `number1 -> number -> number1 -> number -> { x : number1, y : number }` ✓

b. `Int -> Int -> Int -> Int-> { x : Int, y : Int }`

c. `Int -> Int -> Int -> Int-> { a | x : Int, y : Int }`

d. `number -> number -> number -> number -> { x : number, y : number }`

e. `number1 -> number -> number1 -> number -> { a | x : number1, y : number }`

Question **12**

Given the following type definitions:

Question **12**
Correct
Mark 1.00 out of 1.00
Flag question

Given the following type definitions:

```
type Size = S | M | L
type Color = Red | Yellow | White | Blue
type Shirt = Shirt Color Size
```

The cardinality of Shirt is:

Answer: 12 ✔

Information
Flag question

The next 2 questions are advanced questions.

Read each question carefully.

Question **13**
Incorrect
Mark 0.00 out of 1.00
Flag question

Select the result of each function:

> List.foldl (/) 8.0 [8.0, 4.0, 2.0]

0.125 ✖

> List.foldr (/) 8.0 [8.0, 4.0, 2.0]

0.125 ✖

Question **14**
Partially correct
Mark 0.33 out of

Given the following definitions:

```
something lx ly =
```

Question **14**

Partially correct

Mark 0.33 out of 1.00

Flag question

Given the following definitions:

```
something lx ly =
    case (lx, ly) of
        ([], _) -> ly
        (_, []) -> lx
        (x::xs, y::ys) -> if x > y then x::something xs ly else y::something lx ys

mystery l =
    case l of
        [] -> []
        [x] -> [x]
        _ ->
            let
                h = (List.length l) // 2
                a = (mystery (List.drop h l))
                b = (mystery (List.take h l))
            in
               something a b
```

mystery [10, 5, 2, 8, 4, 15] evaluates to [15 ✔ , 4 ✖ , 8 ✔ , 2 ✖ , 5 ✖ , 10 ✖]