1. [5]How many elements are in the following type?

   ```
   SomeType ::= Red
              | Blue Bool
              | Green Bool Bool
   ```

   A. 8
   [✓ B. 7]
   C. 6
   D. 3

2. [5]How many elements are in the following type?

   ```
   Etyb ::= Etyb Bool Bool Bool Bool Bool Bool Bool Bool
   ```

   [✓ A. 256]
   B. 8
   C. 512
   D. 1

3. [5]What is the type of the following function?

   ```
   h f g (x :: xs) = f g
   h f g []        = g
   ```

   A. $(\mathrm{A} \to A) \to (A \to B) \to \text{List } B \to A$
   B. $(A \to B) \to B \to \text{List } B \to B$
   [✓ *C.* $(A \to A) \to A \to \text{List } B \to A$]
   D. $A \to A \to \text{List } B \to A$

4. [5]What is the type of the following function?

   ```
   f Zero        = True
   f (Succ Zero) = False
   ```

   A. $\mathbb{B} \to \mathbb{N}$
   B. $\mathbb{B}$
   C. $\mathbb{N}$
   [✓ D. $\mathbb{N} \to \mathbb{B}$]

5. [5]What is the result of `baz (< 2) [0,1,2,3,4,5]`?

   ```
   baz : (A -> Bool) -> List A -> List A
   baz f []        = []
   baz f (x :: xs) = f x match
                       True  -> x :: baz f xs
                       False -> baz f xs
   ```

A. []
B. [2,3,4,5]
[✓ C. [0,1]]
D. [0,1,2]

6. [5]What is the result of `bar (not) [True, False, True, False]`. Given:

```
bar : (A -> B) -> List A -> List B
bar f []        = []
bar f (x :: xs) = (f x) :: bar f xs
```

A. [True, True, True, True]
B. [False, False, False, False]
C. [True, False, True, False]
[✓ D. [False, True, False, True]]

7. [5]Which of these types represent the counting numbers?
A. $\mathbb{Z}$
B. $\mathbb{B}$
C. **List**
[✓ D. $\mathbb{N}$]

8. [5]What is the type of the following function?

```
f 0 p = p 0
f n p = not (p n)
```

[✓ A. $\mathbb{N} \to (\mathbb{N} \to \mathbb{B}) \to \mathbb{B}$]
B. $\mathbb{N} \to (\mathbb{B} \to \mathbb{N}) \to \mathbb{B}$
C. $\mathbb{B}$
D. $\mathbb{N} \to \mathbb{B} \to \mathbb{B}$

9. [5]What is the type of the following list? xs = (Cons 5 (Cons 4 (Cons "Three" (Cons 2.0 Empty))))?
A. List $\mathbb{N}$
B. List $\mathbb{R}$
[✓ C. None, It is not well-typed.]
D. List **String**

10. [5]What is the result of `foo (+) 1 [1,2,3,4])`. Given the following definition of foo

```
foo : (A -> B -> B) -> B -> List A -> B
foo f b []        = b
foo f b (x :: xs) = foo f (f x b) xs
```

A. 24
B. 10

[✓ C. 11]
D. 4

11. [5]What is the type of the following function?

```
g True (x :: xs) = xs
g n xs = xs
```

A. $\mathbb{B} \to \text{List } \mathbb{B} \longrightarrow \text{List } \mathbb{B}$
B. $\mathbb{B} \to \text{List } A \to A$
C. $\mathbb{B} \to \text{List } \mathbb{N} \longrightarrow \text{List } \mathbb{N}$
[✓ D. $\mathbb{B} \to \text{List } A \to \text{List } A$]

12. [5]How many elements are in the following type?

```
SomeType ::= Red SomeType
            | Blue
```

A. 1
B. 0, It is not a valid type
[✓ C. Infinity]
D. 2

13. [5]What is the name of the following function?

```
foo : (A -> B -> B) -> B -> List A -> B
foo f b []        = b
foo f b (x :: xs) = foo f (f x b) xs
```

A. map
[✓ B. fold]
C. prod
D. reduce

14. [5]A function which always returns y when given the same x is:
A. tail-recursive
[✓ B. pure]
C. polymorphic
D. higher-ordered

15. [5]A function is called this when it's type may change depending on what arguments it is given:
A. dependent
B. tail-recursive
C. total
[✓ D. polymorphic]

16. [5]The following function is tail-recursive:

```
foo : (A -> B -> B) -> B -> List A -> B
foo f b []        = b
foo f b (x :: xs) = foo f (f x b) xs
```

[✓ A. True]
B. False

17. [5]This is the type of functions which only make recursive calls when therecursive call is the only computation being performed on the right-hand sideof the definition
A. total
B. dependent
C. polymorphic
[✓ D. tail-recursive]

18. [5]What is the type of `Just 15`?
A. ℕ
[✓ B. Maybe ℕ]
C. None, it is not well-typed
D. Z

19. [5]What is the name of the type we use to handle error in functions?
A. **List**
[✓ B. **Maybe**]
C. 𝔹
D. **Tree**

20. [5]Which of the following types always has exactly two elements?
A. **List**
B. ℕ
C. **Maybe**
[✓ D. 𝔹]

21. [5]A function which takes another function as an argument is:
A. compositional
B. tail-recursive
[✓ C. higher-ordered]
D. dependent

22. [5]What is the name of the following function?

```
baz : (A -> Bool) -> List A -> List A
baz f []        = []
baz f (x :: xs) = f x match
                      True  -> x :: baz f xs
                      False -> baz f xs
```

A. sum

[✓ B. filter]
C. reduce
D. fold

23. [5]Which is the correct way of encoding the *integer* 3?
A. Cons (Cons (Cons Empty))
[✓ B. Positive (Succ (Succ (Succ Zero)))]
C. Positive (Succ (Succ (Succ (Positive Zero))))
D. Succ (Succ (Succ Zero))

24. [5]Is the following function tail recursive?

```
bar : (A -> B) -> List A -> List B
bar f []        = []
bar f (x :: xs) = (f x) : bar f xs
```

[✓ A. No]
B. Yes

25. [5]Which of these types is polymorphic?
A. $\mathbb{N}$
B. $\mathbb{Z}$
[✓ C. **Maybe**]
D. $\mathbb{B}$

26. [5]A function which is defined over the entire input domain is called:
[✓ A. total]
B. higher-ordered
C. tail-recursive
D. pure

27. [5]What is the name of the following function?

```
bar : (A -> B) -> List A -> List B
bar f []        = []
bar f (x :: xs) = (f x) :: bar f xs
```

A. sum
B. fold
C. reduce
[✓ D. map]