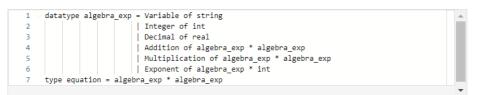
## **Exam for Part A**

1.	Check a box if and only if it is an accurate description of ML	8分
	✓ ML uses lexical scope for the semantics of looking up variables in the environment	
	ML has no language constructs for creating mutable data	
	ML has a REPL as part of the definition of the language	
	✓ ML is statically typed	
2.	Here is a particular list of pairs in ML:	12分
	[(4,19), (1,20), (74,75)]	
	For each $pattern$ below, check the box if and only if this pattern matches the value above.	
	✓ x::y	
	▼ x::(y::z)	
	(a,b,c)::d	
	√ (a,b)::(c,d)::(e,f)::[]	
	√ (a,b)::(c,d)::(e,f)::g	

- mystery uses currying to take two arguments.
- mystery uses tupling to take two arguments.
- ✓ If the second argument to mystery is a zero-element list, then whenever mystery produces a result, the result is NONE.
- If the second argument to mystery is a one-element list, then whenever mystery produces a result, the result is
- If the second argument to mystery is a two-element list, then whenever mystery produces a result, the result is MONE.
- ightharpoonup The argument type of  ${f f}$  can be any type, but it must be the same type as the element type of  ${f xs}$ .
- $\Box$  The result type of **f** can be any type, but it must be the same type as the element type of **xs**.
- If you replace the first line of the code with fun mystery f = fn xs =>, then some callers of mystery might no longer type-check.
- If you replace the first line of the code with fun mystery f = fn xs =>, then some callers of mystery might get a different result.
- g is a tail-recursive function.
- For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is always the same as the length of someList (for any someFun and someList).
- ▼ For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is sometimes the same as the length of someList (depending on someFun and someList).
- For the entire computation of a call like mystery someFun someList, the total number of times someFun is called is never the same as the length of someList (for any someFun and someList).

4.	The null function is predefined in ML's standard library, but can be defined in many ways ourselves. For each suggested8分 definition of null below, check the box if and only if the function would behave the same as the predefined null function whenever the function below is called.						
	Note: Consider only situations where calls to the functions below type-check.						
	✓ fun null xs = case xs of [] => true   _ => false						
	✓ fun null xs = xs=[]						
	fun null xs = if null xs then true else false						
	$\checkmark$ fun null xs = ((fn z => false) (hd xs)) handle List.Empty => true						
5.	The next four questions, including this one, relate to this situation: Suppose somebody has written a library for a collection of strings (perhaps implemented as some sort of linked list of strings or tree of strings, but the details do not matter). The library includes higher-order functions map, filter, and fold that operate on these collections and have their conventional meanings. For each problem below, decide which of these library functions is the best to use for implementing the desired function.						
	(For those needing a precise definition of best: On this exam, the best function, given appropriate arguments, returns the final result you need, meaning you need no more computation after calling the function. If multiple functions can do this, choose the one that can be used by passing it the function argument that itself does the least amount of work.)						
	Desired function: Take a collection of strings and produce a new collection where each string in the output is like a string in the input except the string has any space characters removed.						
	• map						
	○ filter						
	○ fold						
6.	Desired function: Take a collection of strings and return a string that is the concatenation of all the strings in the collection.						
	O map						
	○ filter						
	fold						
7.	Desired function: Take a collection of strings and a number n and return how many strings in the collection have a length						
	O map						
	( ) filter						
	O fold						
8.	Desired function: Take a collection of strings and return a collection containing the strings in the input collection that start 3分 with a capital letter.						
	O map						
	• filter						
	O fold						

a -	This datatype binding and type s	vnanum ara ucaful f	or representing	cartain agustions from algabras



Which of the mathematical equations below could not be elegantly represented by a value of type equation?

- $\bigcirc x + y = z$
- $\bigcirc (x+4) + z = 7 \cdot y$
- $\bigcirc \ x^3 \cdot y^2 = z^0$
- $\bigcirc$  14.2 + 3 = 17.2

10. Here is a particular polymorphic type in ML:

10分

4分

```
1 'a * 'b -> 'b * 'a * 'a
```

For each type below, check the box if and only if the type is an instantiation of the type above, which means the type above is more general.

- $\square$  string \* int -> string \* int \* int
- ✓ int \* string -> string \* int \* int
- ✓ int \* int -> int \* int \* int
- $\square$  {foo : int, bar : string} -> {a : string, b : int, c : int}
- \_\_\_ 'a \* 'a -> 'a \* 'a \* 'a

3分

```
structure NoNegativeCounter :> COUNTER =
2
     struct
     exception InvariantViolated
    type t = int
    fun newCounter i = if i <= 0 then 1 else i
10 fun increment i = i + 1
11
12
    fun first_larger (i1,i2) =
        if i1 <= 0 orelse i2 <= 0
13
        then raise InvariantViolated
14
       else (i1 - i2) > 0
15
16
17
    end
18
```

In each problem, the definition of COUNTER matches the structure definition NoNegativeCounter, but different signatures allow clients to use the structure in different ways. You will answer the same question for each COUNTER definition by choosing the best description of what it allows clients to do.In this question, the definition of COUNTER is:

```
1 signature COUNTER =
2 sig
3 type t = int
4 val newCounter : int -> t
5 val increment : t -> t
6 val first_larger : t * t -> bool
7 end
```

- This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.
- This signature makes it impossible for any client to call NoNegativeCounter.first\_larger at all (in a way that causes any part of the body of NoNegativeCounter.first\_larger to be evaluated).
- This signature makes it possible for clients to call NoNegativeCounter.first\_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

12.In this question, the definition of COUNTER is:

1 signature COUNTER =

type t = int

val newCounter : int -> t

val first\_larger : t \* t -> bool

sig

5

6 end

- This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.
- This signature makes it impossible for any client to call NoNegativeCounter.first\_larger at all (in a way that causes any part of the body of NoNegativeCounter.first\_larger to be evaluated).
- This signature makes it possible for clients to call NoNegativeCounter.first\_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

3分

3分

```
1 signature COUNTER =
2 sig
3 type t
4 val newCounter: int -> int
5 val increment: t -> t
6 val first_larger: t * t -> bool
7 end
8
```

- This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be
- This signature makes it impossible for any client to call NoNegativeCounter.first\_larger at all (in a way that causes any part of the body of NoNegativeCounter.first\_larger to be evaluated).
- This signature makes it possible for clients to call NoNegativeCounter.first\_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

14.In this question, the definition of COUNTER is:

1 signature COUNTER =
2 sig
3 type t
4 val newCounter : int -> t
5 val increment : t -> t
6 val first\_larger : t \* t -> bool
7 end

- This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised
- This signature makes it impossible for any client to call NoNegativeCounter.first\_larger at all (in a way that causes any part of the body of NoNegativeCounter.first\_larger to be evaluated).
- This signature makes it possible for clients to call NoNegativeCounter.first\_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

15.In this question, the definition of COUNTER is:

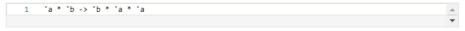
1 signature COUNTER =
2 sig
3 type t = int
4 val newCounter : int -> t
5 val increment : t -> t
6 end

- This signature allows (some) clients to cause the NoNegativeCounter.InvariantViolated exception to be raised.
- This signature makes it impossible for any client to call NoNegativeCounter.first\_larger at all (in a way that causes any part of the body of NoNegativeCounter.first\_larger to be evaluated).
- This signature makes it possible for clients to call NoNegativeCounter.first\_larger, but never in a way that leads to the NoNegativeCounter.InvariantViolated exception being raised.

## ERROR:

10. Here is a particular polymorphic type in ML:

8/10分



For each type below, check the box if and only if the type *is* an instantiation of the type above, which means the type above is more general.

**锁**误

14. In this question, the definition of COUNTER is:

```
0/3 分
```

```
1  signature COUNTER =
2  sig
3  | type t
4   val newCounter : int -> t
5   val increment : t -> t
6   val first_larger : t * t -> bool
7  end
```

```
错误
```