COMP 3958: Lab 5

Submit a zip file containing the two files `primes.ml` and `exp.ml`. Note that if your program does not build, you may not receive credit for it. Maximum score: 12

Consider the two different stream types – infinite streams and lazy streams – defined by

```
type 'a infstream = Cons of 'a * (unit -> 'a infstream)
type 'a lazystream = Cons of 'a * 'a lazystream Lazy.t
```

We have seen a number of functions to operate on such streams in class; functions like `hd`, `tl`, `take`, `drop`, `map` and `from n`. (`from n` returns the stream of integers starting from `n`.)

1. (a) Implement a function `filter` with signature

    ```
    val filter : ('a -> bool) -> 'a infstream -> 'a infstream
    ```

    so that `filter f s` is the sub-stream of `s` whose elements satisfy the predicate `f`. It is the analog of `List.filter` for infinite streams. Note that if no element of `s` satisfies `f`, `filter f s` does not terminate. Note that we are using `infstream` in this part.

   (b) Using `filter` with some of the functions we have seen in class, implement, based on the sieve of Eratosthenes, the stream `primes` that represents the infinite sequence of primes. Note that `primes` has type `int infstream`.

    Apply `take` on `primes` to find the first 100 primes.

   Put your code in a file named `primes.ml`. You will need to include the implementation of any function from class that you use.

2. (a) The exponential function can be defined by the infinite series

    $$\exp(x) = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \frac{x^4}{4!} + \cdots$$

    Implement a function `exp_terms` with signature

    ```
    val exp_terms : float -> float lazystream
    ```

    that returns a lazy stream consisting of the terms in the above infinite series, i.e, the terms are $1, x, \frac{x^2}{2!}, \frac{x^3}{3!}, \cdots$. Note that we are using `lazystream` in this part.

   (b) By summing the first 20 terms of the stream in (a), get an approximate value for $exp(1.1)$.

   Put your code in a file named `exp.ml`. Again, you will need to include the implementation of any function from class that you use.