Introduction to Programming: Quiz 1 (Make-up)

Date: November 23, 2026 **Marks:** 10

Use the space provided after each question to write your answer.
Write only the final solution here. Use your own blank paper for rough work.
The first three questions carry 2 marks each. The fourth is worth 4 marks.

---

[2] 1. Using `filter`, write a function `dropMultiples` that takes an integer `n` and a list of integers `xs`, and removes every multiple of `n` from `xs`. For example, `dropMultiples 3 [1..10] = [1, 2, 4, 5, 7, 8, 10]`.

[2] 2. Using `takeWhile`, write a function `sumUntilNeg` that takes a list of integers and returns the sum of the elements that appear before the first negative number. For example, `sumUntilNeg [3, 5, -2, 7] = 8` and `sumUntilNeg [-1, 4] = 0`.

[2] 3. Using `foldr`, write a function `countOcc` that takes a value `x` and a list `xs`, and returns the number of times `x` occurs in `xs`. For example, `countOcc 'a' "banana" = 3`. Also give the most general type of `countOcc`.

4. A value `xss` of type `[[a]]` is a *matrix* if it is non-empty, its first row is non-empty, and every row has the same length as the first row.

[1] (a) Write a function `isMatrix :: [[a]] → Bool` that checks whether its input is a matrix. For example, `isMatrix [[1, 2], [3, 4], [5, 6]] = True`, while `isMatrix [[1, 2], [3]] = False` and `isMatrix [] = False`.

[3] (b) Write a function `transpose :: [[a]] → [[a]]` that computes the transpose of its input, assuming that the input is a matrix. Row i of the output consists of the i-th elements of the input rows, in order. For example, `transpose [[1, 2, 3], [4, 5, 6]] = [[1, 4], [2, 5], [3, 6]]`.