Homework 4

October 3, 2023

1 Pattern Matching

Use **Pattern Matching** and let expression (as needed) to solve the following problems. Note that you can still use the if-then-else expressions.

In this assignment, the matrices are represented by two dimensional lists. For example,

```
val x = [[1,2,3],[4,5,6],[7,8,9]];
```

represents the matrix.

```
1 2 3
4 5 6
7 8 9
```

```
val y = [[1,1,1],[2,2,2],[3,3,3]];
```

represents the matrix.

```
1 1 1
2 2 2
3 3 3
```

Note that you may reuse the functions you defined in homework 3 by including them in this submission.

1. Write a function `matrixAdd : int list list * int list list -> int list list` that adds two matrices of the same dimension. For example `matrixAdd (x, y)` should return the matrix below:

   ```
   2 3 4
   6 7 8
   10 11 12
   ```

2. Write a function `transpose: 'a list list -> 'a list list` that transposes a matrix. For example `transpose x` should return the matrix below:

```
1 4 7
2 5 8
3 6 9
```

3. Write a function `dotProduct : int list * int list -> int` that returns the dot product of two vectors. For example `dotProduct ([1,2,3], [4,5,6])` should return `32`.

4. Write a function `scalarMatrixProduct: int * int list list -> int list list` that multiples a scalar with a matrix. For example `scalarMatrixProduct (10, x)` should return the following matrix:

```
10 20 30
40 50 60
70 80 90
```

5. Write a function `matrixProduct: int list list * int list list -> int list list` that multiples two matrices. For example `matrixProduct (x, y)` should return the following matrix:

```
14 14 14
32 32 32
50 50 50
```

Note that you may get "Pattern matching is not exhaustive" warning for some functions, which is acceptable.

2 Test cases

The following test code assumes you have the functions in homework 3 included.

```
val show = print o matrixToString;
val x  = [[1,2,3], [4,5,6], [7,8,9]];
val y = [[1,1,1],[2,2,2],[3,3,3]];

show (matrixAdd(x,y));
dotProduct ([1,2,3],[4,5,6]);
show (transpose x);
show (scalarMatrixProduct (10, x));
show (matrixProduct (x, y));
```

3 Submission

Submit your SML program in a ASCII text file by the name of `hwk4.sml` to the dropbox. The `sml` file extension doesn't change the file format, which is still a plain text file. If you want to submit `hwk4.txt` that is fine as well. Other formats of submissions will be rejected.