

60 points + 10 extra credit

Please submit your solution using the `handin` program. Submit the program as
`cs448b hw1`

This requires you to have an account on the UBC Computer Science undergraduate machines. If you need an account, go to: <https://www.cs.ubc.ca/students/undergrad/services/account> to request one.

Your solution should consist of an Erlang source file called `hw1.erl` and a second file with your responses to written questions (e.g. question 4). Your written response file can be any of:

`hw1.txt` – plain, ASCII text.

`hw1.ps` – PostScript.

`hw1.pdf` – PDF.

(Note: Word and other proprietary formats are not acceptable).

1. **Palindromes (10 points).** Write an Erlang function

```
is_palindrome(List) -> bool()
```

that returns `true` if its argument is a list that is a palindrome, and `false` if otherwise (i.e. if the list is not a palindrome, or the argument is not a list). A list is a palindrome if it is unchanged by reversing the order of its elements. For example:

```
is_palindrome([1,3,3,1]) -> true;
is_palindrome([cat,dog,potoroo,dog,cat]) -> true;
is_palindrome([cat,dog,potoroo,cat,dog]) -> false;
is_palindrome({1,3,3,1}) -> false.
```

For this problem, you may not use any functions from the Erlang libraries. In particular, you may not use functions from the `lists` module.

2. **Maxima (10 points).** Write an Erlang function

```
maxima(List) -> M
```

where `M` is a list of tuples of the form `{Pos, Value}` that indicate the positions, `Pos`, and values, `Value`, of the elements that are local maxima of `List`. An element is a local maxima iff

```
( it is the first element of the list
  or it is greater than or equal to the previous element
)
and( it is the last element of the list
     or it is greater than or equal to the next element
)
```

For example:

```
maxima([1, 2, 3, 2, 1, 2]) -> [{3,3}, {6,2}];
maxima([1, 42, -3, 17, 53, 92, -4, -4, -4, -4, 2, 1]) ->
  [{2,42}, {6,92}, {8,-4}, {9,-4}, {11,2}].
```

For this problem and the rest of this assignment, you may use any functions that you like from the Erlang libraries.

3. **Polynomials (40 points).**

(a) **Evaluation (10 points).** Write an Erlang function

```
poly_eval(Poly, X) -> number()
Types:
Poly = [ number() ] % a list of numbers
X = number()
```

The elements of list `Poly` are the coefficients of a polynomial. The function `poly_eval` evaluates this polynomial for the number `X`. In particular:

$$\text{poly_eval}(\text{Poly}, X) \rightarrow \sum_{I=1}^{\text{length}(\text{Poly})} \text{lists:nth}(I, \text{Poly}) * X^{I-1}$$

For example:

```
poly_eval([1,3,3,1], 2) -> 27;
poly_eval([0,1,2,3,4], 5) -> 2930.
```

(b) **Sum (10 points).** Write an Erlang function

```
poly_sum(P1, P2) -> PS()
Types:
P1,P2,PS = [ number() ] % polynomials, as for question 3a.
```

Where `PS` is the list that represents the polynomial that is the sum of the polynomials represented by `P1` and `P2`. Thus,

$$\text{poly_eval}(\text{PS}, X) = \text{poly_eval}(\text{P1}, X) + \text{poly_eval}(\text{P2}, X)$$

(c) **Product (20 points).** Write an Erlang function

```
poly_prod(P1, P2) -> PP()
Types:
P1,P2,PP = [ number() ] % polynomials, as for question 3a.
```

Where `PP` is the list that represents the polynomial that is the product of the polynomials represented by `P1` and `P2`. Thus,

$$\text{poly_eval}(\text{PP}, X) = \text{poly_eval}(\text{P1}, X) * \text{poly_eval}(\text{P2}, X)$$

```
poly_prod([1,2], [3,5]) -> [3,11,10].
```

4. **(10 points, extra credit).** For each problem on this assignment:

- (a) How long did it take you to solve the problem?
- (b) How long do you estimate that it would take you to solve a similar problem now that you have some Erlang programming experience?
- (c) Please rate each problem on a scale of 0 to 5 where
 - 0 – Worthless tedium.
 - 1 – Too much work, and I little learned.
 - 2 – A typical homework problem.
 - 3 – Definitely had a favorable learning/effort ratio.
 - 4 – I learned a lot and had fun doing so.
 - 5 – Wow! I’ve discovered a new way to think!