CSci 2041:

Advanced Programming Principles

University of Minnesota,
Prof. Van Wyk,
Spring 2018

Exercise #1:Circle area

Write an OCaml function named `circle_area` with type `float -> float` that computes (surprise) the area of a circle given its radius.

Exercise #2:

With a partner, write an OCaml function named `power` with the type `int -> float -> float`.

`power 3 3.0` should return `27.0`.

`power 3 3.2` should return `32.768`.

Exercise #3:

We previously wrote cube as follows:

```
let cube x = x * x * x
```

Write another version that uses power, preferably taking advantage of the curried nature of functions in OCaml.

What is the minimal number of characters needed to do this?

Exercise #4:

Working in pairs, write an OCaml function named all that returns true is all elements of the list are true.

That is, if there are no elements that are false.

Recall sum:

```
let rec sum xs =
  match xs with
  | [ ] -> 0
  | x::rest -> x + sum rest
```

Exercise #5:Even 2 ways

Write a function even2ways that checks if an integer list only contains even values and has an even number of elements.

```
let rec even2ways (xs :  int list) :  bool = ...
```

Exercise #6:

Working in pairs, write an OCaml function named `is_empty` that returns `true` is the list is empty, and `false` otherwise.

Exercise #7:

Working with a *different* partner, write an OCaml function named `head` that returns the front element of the list.

This specification is intentionally incomplete - what decisions must you make to complete this function?

What is it's type?