(require 2htdp/image)
(require spd/tags)

;; ========
;; Problem 1
;;
;; The area of a triangle is the (base * height)/2. Design a function
;; that consumes a triangle's base and height and produces its area.
;; Your solution must have all HtDF recipe elements include @tags.

(@Problem 1)
;; Problem 2
;;
;; The following program runs and all tests pass. But the program has a design
;; flaw. Correct that design flaw by NEATLY crossing out what is incorrect and
;; writing what is correct nearby. HINT: The answer calls for very little
;; crossing out and writing — if you want to cross out and write a lot then you
;; have not found the right answer.

@Problem 2
@HtDD Pos
(define-struct pos (x y))
;; Pos is (make-pos Natural Natural)
;; interp. An x,y position on a image, in pixels.
(define P1 (make-pos 3 4))

@dd-template-rules compound

(define (fn-for-pos p)
  (... (pos-x p)
       (pos-y p)))

@HtDF swap
;; Pos -> Pos
;; Produce new pos with x and y swapped.
(check-expect (swap (make-pos 1 1)) (make-pos 1 1))
(check-expect (swap (make-pos 2 2)) (make-pos 2 2))

;(define (swap p) ;stub

@template Pos

(define (swap p)
  (make-pos (pos-y p) (pos-x p)))
Problem 3

In each part of this problem you will be given a short description of problem domain information. You need to determine what form of data definition is most appropriate to representing that information.

In each part of the problem you must:
- first select how many data definitions are required to represent the problem domain information
- select the form of those data definitions

NOTE that if you select 1 data definition then you must only select one form of definition. If you select more than one form when you mark 1 you will lose all marks for that part of the problem.

3A:

A province in Canada. Note that there are 13 provinces and territories in Canada: Alberta, British Columbia, Manitoba, New Brunswick, Newfoundland and Labrador, Northwest Territories, Nova Scotia, Nunavut, Ontario, Prince Edward Island, Quebec, Saskatchewan, and Yukon.

How many data definitions? (circle one) 1 2

form of 1st data definition (circle one) form of possible 2nd data definition (circle one only if you selected 2 above)

simple atomic
interval
enumeration
itemization
compound
self-reference
reference
3B:

A simple transcript, with courses numbers taken and the grade in each course.

How many data definitions? (circle one) 1  2

form of 1st data definition
(circle one)

form of possible 2nd data definition
(circle one only if you selected 2 above)

simple atomic

interval

enumeration

itemization

compound

self-reference

reference
Problem 4

Write the complete (@dd-template-rules ...) tag and the template for the
following type comment. Be sure to follow ALL applicable rules, and simplify
the template to the extent that the rules allow. Be sure to keep the cond
Q/A pairs in the same order as the one-of subclasses.

(@Problem 4)

(@HtDD Farfle)

Farfle is one-of:

- "low"
- Number[0, 100]
- "high"
Problems 5 and 6 both use the following domain information.

Different courses in the computer science department have different enrolment numbers. For the 2016 winter term, those numbers were:

<table>
<thead>
<tr>
<th>Course</th>
<th>Enrolment</th>
</tr>
</thead>
<tbody>
<tr>
<td>100</td>
<td>211</td>
</tr>
<tr>
<td>103</td>
<td>210</td>
</tr>
<tr>
<td>110</td>
<td>1635</td>
</tr>
<tr>
<td>121</td>
<td>1042</td>
</tr>
<tr>
<td>210</td>
<td>971</td>
</tr>
<tr>
<td>213</td>
<td>671</td>
</tr>
<tr>
<td>221</td>
<td>912</td>
</tr>
</tbody>
</table>

One way to represent an individual course is as follows:

```scheme
(@HtDD Course)
(define-struct course (lvl num siz)) ;; Course is (make-course Natural Natural Natural)
;; interp. a course with size.
;; lvl (level) is 100, 200 etc.
;; num (num) is 03, 10, 21 etc.
;; siz (size) is number of students enrolled.
(define CPSC-100-2016 (make-course 100 00 211))
(define CPSC-103-2016 (make-course 100 03 210))
(define CPSC-110-2016 (make-course 100 10 1635))
(define CPSC-121-2016 (make-course 100 21 1042))
(define CPSC-210-2016 (make-course 200 10 971))
(define CPSC-213-2016 (make-course 200 13 671))
(define CPSC-221-2016 (make-course 200 21 912))

(@dd-template-rules compound) ; 3 fields

(define (fn-for-course c)
  (... (course-lvl c)
       (course-num c)
       (course-siz c)))
```
We would like to be able to work with an arbitrary number of such courses. Design a complete data definition for ListOfCourse. You must include type comment, interpretation, examples, @dd-template-rules tag, and a template.

(@Problem 5)
In the bubble chart above each circle corresponds to a single course. At the center of each circle is the course number. The y position of the circle comes from the course level — all 100 courses have the same y position for example. The x position comes from the course number. The size of the circle is proportional to the size of the course.

Design a function that consumes ListOfCourse and produces a SIMPLE bubble chart like the one above. Follow all design rules including helper rules and provide all HtDF recipe elements include @ tags.

More points will be awarded to solutions that follow the recipes and produce a rough visualization than to solutions that produce a great visualization but do not follow the recipes. Many more points.

HINT: relax and follow the recipes. If you do you will see that this problem is easier than it looks. Don't worry about the exact name of the 2htdp/image functions you will need to call, just make reasonable guesses about those.