CSCI–4150 Introduction to Artificial Intelligence, Fall 2004
Assignment 1 (64 points), out Thursday September 2, due Thursday September 9

Notes on this assignment

- You should download the `assign1.scm` file from the assignment web page and write your code in this file. It will contain stubs for the procedures that you need to write.
- Problems 1–3 are to be turned in on paper. The rest are to be turned in electronically. You will find a link to the web tester from the assignment web page. See the web page for submission policy and details.
- The standard late deadlines apply. (See the syllabus for details.) Remember, any late written work should be turned into Owen Kellet's mailbox in the CS lounge on the first floor of the Amos Eaton building (or turned in in class).
- You may always assume (on all assignments in this class) that your procedures will be given valid inputs.

Questions

1. (written; 4 points) Section 3.6, Exercise 1 [math expressions]
2. (written; 4 points) Section 8.5, Exercise 2, parts d, e, f, g [list creation]

   Note that you must write these expressions using only the `list`, `cons`, and `append` procedures!
3. (written; 4 points) Section 8.5, Exercise 4, parts d, e, f, g [list exercises]

   Note that you must write these expressions using only procedures from Section 8!
4. (6 points) Write the procedure `(make-change amount)` where `amount` is an integer between 0 and 99, inclusive. It should calculate the minimum number of coins (quarters, dimes, nickels, and pennies) necessary to produce `amount` cents. Your answer should be returned in a list of the form: This procedure should return a list of nonnegative integers of the form: `(Q D N P)`. For example:

   ```
   > (make-change 89)
   ;Value: (3 1 0 4)
   ```
5. (10 points) Section 6.3, Exercise 3 [`income-tax`]

   There will be a hint on the assignment web page for this problem, but try to figure it out yourself first.
6. (8 points) Write the procedure `(swap Lst j k)` that returns a list identical to `Lst` except that elements `j` and `k` are swapped. `j` and `k` are integers between 0 and `(- (length Lst) 1)` inclusive. For example:

   ```
   > (swap '(a b c d e f) 0 4)
   ;Value: (e b c d a f)
   ```

   You may use the MIT/GNU Scheme `sublist` procedure for this problem.
7. (6 points) Write the procedure `(element-number e Lst)` which returns the position of the first occurrence of `e` in the list `Lst` (starting from the front of the list). If `e` does not appear in `Lst`, it should return the length of `Lst`. For example:

```
> (element-number 'c '(a b c d b c a))
;Value: 2
> (element-number 'f '(a b c d b c a))
;Value: 7
```

8. (6 points) In preparation for plus/minus grading, I need you to write me a procedure to automatically assign plus/minus grades. Write the procedure `(plus/minus-ize grade)` that takes the `grade` argument (which must be one of the symbols `A`, `B`, `C`, `D`, and `F`) and returns a randomly generated plus- or minus-version of that grade. Recall that there is no A+, D-, F+, or F-. Therefore, the relation between input and output symbols for this procedure is:

`grade` argument	possible return values (select one randomly)
`A`	`A`, `A-`
`B`	`B+`, `B`, `B-`
`C`	`C+`, `C`, `C-`
`D`	`D+`, `D`
`F`	`F`

Note: this is not an exercise in string manipulation or symbol creation; you will have to hardcode all the symbols above. Instead, this is an exercise in using conditional expressions and selecting random elements from a list. Hint: you should use a `case` statement in this problem.

9. (6 points) Write the procedure `(meters->distance m)` where `m` is a distance in meters. It should convert this distance into a number of microns, millimeters, centimeters, meters, or kilometers. The unit should be chosen so that the corresponding number is the smallest number greater than or equal to 1, e.g., 1.5 centimeters instead of 15 millimeters and 90 centimeters instead of 0.9 meters.

The result should be returned in a list where the first element is the number and the second element is either: `microns`, `millimeters`, `centimeters`, `meters`, or `kilometers`. For example:

```
> (meters->distance 0.0001)
;Value: (100. microns)
> (meters->distance 0.0035)
;Value: (3.5 millimeters)
> (meters->distance 0.013)
;Value: (1.3 centimeters)
> (meters->distance 1876)
;Value: (1.876 kilometers)
```

Hint: you should use a `cond` statement for this problem.

10. (10 points) A series that converges reasonably quickly to π is given by:

$$\pi = \frac{3\sqrt{3}}{2} \sum_{i=0}^{\infty} \frac{(i!)^2}{(2i+1)!}$$

Write a recursive procedure `(approx-pi N)` which sums the first (`N`+1) terms of this series (i.e., terms for $i = 0$ through $i =$`N`). For example:

```
> (approx-pi 0)
;Value: 2.598076211353316
```

You should get an approximation accurate to four decimal places for `N`=6.