

## CS 101 Computer Science I (Spring 2001) Assignment 7

1. Write a Scheme procedure called "**each-matched-pair?**" that takes two lists called "**lst1**" and "**lst2**" and a binary predicate called "**pred?**" as inputs. The **each-matched-pair?** procedure applies **pred?** to each corresponding pair of members of **lst1** and **lst2**, i.e., it applies **pred?** to the 1st member of **lst1** and the 1st member of **lst2**; it also applies **pred?** to the 2nd member of **lst1** and the 2nd member of **lst2**, etc. The **each-matched-pair?** procedure returns the boolean value **#t** if **pred?** returns the value **#t** each and every time. Otherwise, **each-matched-pair?** returns the boolean value **#f**. If **lst1** and **lst2** have different lengths, **each-matched-pair?** ignores the extra members at the end of the longer list.

```
(each-matched-pair? '(1 5 3 9) '(2 7 4 9) <=)      ==> #t
(each-matched-pair? '(1 7 3 9) '(2 5 4 9) <=)      ==> #f
(each-matched-pair? '()          '()          equal?) ==> #t
(each-matched-pair? '(a b c)    '()          equal?) ==> #t
(each-matched-pair? '()          '(a b c)    equal?) ==> #t
```

2. Write a scheme procedure called "**each-within-n?**" that takes two lists called "**lst1**" and "**lst2**" and an integer called "**n**" as inputs. It returns the boolean value **#t** if the two numbers in each corresponding pair of members of **lst1** and **lst2** differ by **n** or less, regardless of which number is greater than the other. Otherwise **each-within-n?** returns the boolean value **#f**. You should use the procedure **each-matched-pair?** in your definition.
3. Write a Scheme procedure called "**map-successive-pairs**" that takes a list called "**lst**" and a binary procedure called "**fun**" as inputs. The procedure **map-successive-pairs** applies **fun** to each successive pair of members of **lst**, i.e., it applies **fun** to the 1st and 2nd members of **lst**, the 2nd and 3rd members, the 3rd and 4th members, etc. The procedure **map-successive-pairs** returns a list composed of the results of these applications of **fun**. The length of the output list will be one less than the length of the input list.

```
(map-successive-pairs '(2 4 6 8) -)    ==> (-2 -2 -2)
(map-successive-pairs '(2 4 6 8) +)    ==> (6 10 14)
```

4. Write a Scheme predicate called "**uniformly-spaced?**" that takes a list of numbers as input. It returns the boolean value **#t** if each number on the list differs from the previous one by the same amount. If the input list has fewer than two members, then **uniformly-spaced?** returns **#t**. Otherwise, it returns the boolean value **#f**. You should use the procedures **map-successive-pairs** and **each-successive-pair?** in your definition.

### Due Dates

- Section 51 (Professor Welty): Monday April 2, 2001
- Section 52 (Professor Ellman): Tuesday April 3, 2001