

## Assignment # 3A

---

### Details

- Topics: ML
- Weight: 5%
- Due Date: this assignment is due on April 7<sup>th</sup>, 2011 @ 11:59pm
- This assignment is to be done individually.

### Marking

- Your code will be marked for correctness (80%) and code documentation (20%).
- Since we plan to test your code using an auto-marker, you must use the exact function names/arguments as specified and use the same file names and folder name as specified.
- **You will lose 25% of the total mark if you use wrong file names or function names.**
- **You will lose 10% of the total mark if your submission fails to load.**

### Submission

Place all your files in a folder named after your UTORID and submit that folder zipped through the portal (if your UTORID is abcd, the file should be named abcd.zip). A link for submission is provided in the assignments folder in the portal (<http://portal.utoronto.ca>)

### Resources

- ML <http://www.smlnj.org/>
- Robert Harper. *Programming in Standard ML*. Available online: <http://www.cs.cmu.edu/~rwh/smlbook/online.pdf>

### Description

This assignment is to get you used to ML syntax. Problems from Assignment 2 have been picked to make the assignment straight forward for you! Since you know the algorithm already for solving these questions, you will be in a better position to learn ML.

- (a) **[20 marks]** Write a function **intersect** that computes the intersection of two lists. In other words, given two lists as arguments, it returns a list of elements contained in both lists. Note that you cannot use the built-in function `member`. *Sample invocations:*

- **intersect**( [1, 2, 3, 4] , [10, 2, 4, 100] )  
(2 4)

**Write your answer in a file called intersect.sml**

- (b) [20 marks] Define a function **select-even** that takes a list of numbers as input and returns a list that contains all even numbers from the input list, in the same order as they appear in the input list. You can use built-in predicate **even?** *Sample invocation:*

```
-select-even([1, 2, 3, 4])  
(2 4)
```

**Write your answer in a file called select-even.sml**

- (c) [20 marks] Define a function **mult-num** that takes a list as input and returns the product of all numbers in the list. *Sample invocations:*

```
- mult-num([1, 2, 3, 4])  
24
```

**Write your answer in a file called mult-num.sml**

- (d) [20 marks] Define a function **select** that takes a unary predicate and a list as input and returns a list that contains all elements from the input list, of which the predicate is true, in the same order as they appear in the input list. You can assume that the input predicate is applicable to every element of the input list. *Sample invocation:*

```
- select(even, [1, 2, 3])  
[2]
```

**Write your answer in a file called select.sml**