# CS 510: Midterm – 18 October 2018

## 1 Extensions

Below you will find a description of all the extensions. You are required to complete only those extensions corresponding to the topic your were given.

A comment on notation. In our examples below, we will use an abbreviated syntax for lists in order to aid readability. For example, instead of writing `cons(1,cons(2,cons(3,emptylist)))` we will write `[1;2;3]`.

### 1.1 Topic 1

- `replace(item,list,index)`. Replaces item at given index from given list with given new item. For example, `replace(3,[1;3;4;3],7)` should produce the list `[1;7;4;7]`.
- `repetition(list,N)`. Repeats each item in `list` `N` times. For example, `repetition([1;2;3],3)` should produce `[1;1;1;2;2;2;3;3;3]`. Also, `repetition([],2)` should produce `[]`. Finally, `repetition([1;2;3],0)` should produce `[]`.
- `remove_dup(list)`. Removes all the duplicate items from the list. For example, `remove_dup([1;1;2;1;3;2])` should return `[1;2;3]`.
- `intersection(e1,e2)`. Computes a list resulting from the intersection of both lists. For example, `intersection([1;2;3],[4;5;2;3])` should return `[2;3]`. Also, `intersection([1;2;3], [4;5;6])` should return `[]`.

### 1.2 Topic 2

- `insr(item,list,index)`. Inserts given item at given index in given list.
- `remv(index,list)`. Removes item at given index from given list.
- `count(item,list)`. Counts the number of occurrences of given item in a given list.
- `reverse(list)`. Returns the reverse of given list.

### 1.3 Topic 3

- `remv(index,list)`. Removes item at given index from given list.

- `sublist(list,start_index,end_index)`. Returns the sublist of list starting at `start_index` and ending at `end_index`. For example, `sublist([1;2;3;4],0,2)` should produce `[1;2;3]`. Also, `sublist([1;2;3;4],1,1)` should produce `[2]`. Note that `sublist([1;2;3;4],0,5)` should produce an error message.
- `remove_dup(list)`. For example, `remove_dup([1;1;2;1;3;2])` should return `[1;2;3]`.
- `union(list1,list2)`. Returns the union of two lists. For example, `union([1;2;3],[4;5;2;3])` should return `[1;2;3;4;5]` and `union([1;2;3], [4;5;6])` should return `[1;2;3;4;5;6]`.

## 2 Submission instructions

Submit a file named `MidTerm_<SURNAME>.zip` through Canvas. Include all the files.