CSE216 – Programming Abstractions
Recitation 7

Objectives:

- Understanding lists in SML
- Understanding functions in SML
- Practice sample problems[1]

SML Problems

1. Define a function which computes the product of all integers between m and n (with n >= m) inclusive. Use this function to define the function $C_{n,k}$ (the number of combinations of n elements taken k by k), which is defined by

   $C_{n,k} = n!/(k!*(n-k)!)$

---

2. Define a function

   power : int * int -> int

   so that, for m >=0

   power(n,m) = n^m

   holds. Note: we assume that 0^0 is defined as 1.

---

[1] These exercises are taken from Prof. Catuscia Palamidessi's CSE428 course homepage http://www.lix.polytechnique.fr/~catuscia/teaching/cg428/

3. The positive integer square root of a non-negative integer is a function `introot` such that: if `introot` m = n, then n is the largest integer such that n^2 is less than or equal to m. Define the function `introot` in ML.

---

4. Define a function `copy: int * 'a -> 'a list` such that `copy(k,x)` gives the list containing `k` occurrences of `x`. Examples:

```
copy(0,5) = []
copy(1,5) = [5]
copy(3,"a") = ["a","a","a"]
copy(3,copy(1,8)) = [[8],[8],[8]]
```

---

5. Define a function `sumlists: int list * int list -> int list` which takes in input two lists of integers and gives as result the list of the sums of the elements in corresponding position in the input lists. The shortest list has to be seen as extended with 0's. Examples:

```
sumlists([],[]) = []
sumlists([1,2],[3,4]) = [4,6]
sumlists([1],[3,4,2]) = [4,4,2]
sumlists([1,6],[3]) = [4,6]
```

---

6. Define a function `remove_dup: ''a list -> ''a list` which takes in input a list and removes all the duplicates, Examples:

Is it possible to define `remove_dup` with a more generl type, i.e. `remove_dup: 'a list -> 'a list`?

---

7. Define a function `first_list: ('a * 'b) list -> 'a list` which takes in input a list of pairs and gives back the list consisting of the first elements only, Examples:

```
first_list [] = []
first_list [(1,2),(1,3)] = [1,1]
first_list [(1,"a"),(2,"b"),(3,"c")] = [1,2,3]
first_list [([],"a"),([1],"b"),([1,2],"c")] = [[],[1],[1,2]]
```

---

8. Define a function `flatten: 'a list list -> 'a list` which takes in input a list of lists and gives back the list consisting of all the elements, in the same order in which they appear in the argument. Examples:

```
flatten [] = []
flatten [[]] = []
flatten [[1,2],[2,3,4],[5],[],[6,7]] = [1,2,2,3,4,5,6,7]
flatten [["a"],["b","a"]] = ["a","b","a"]
```