CS312 Fall 2001
Prelim 2
November 20, 2001

Name____

NetID__________**@cornell.edu**

Instructions

Write your name and netID on the lines above. There are seven questions on eight numbered pages. Check now that you have all the pages. Write your answers in the boxes provided. Ambiguous answers will be considered incorrect. The exam is closed book except for the handout provided. Do not begin until instructed. You have 90 minutes. **Good luck!**

1	2	3	4	5	6	7	Σ
/14	/12	/9	/7	/9	/7	/3	/61

1. (14 points) Determine the value and type of each of the following expressions as it would be printed by the interpreter. Don't forget to include the type.

```
- val (s,t) = ([1,2,3],[4,5,6]);
val s = [1,2,3] : int list
val t = [4,5,6] : int list
```

(a) `- s@t;`

`val it =` ____________________

(b) `- s::[t];`

`val it =` ____________________

(c) `- map (fn z => z::t) s`

`val it =` ____________________

(d) `- map (fn z => map (fn w => w+z) s) t;`

`val it =` ____________________

(e) `- let val s = tl t val t = hd s in t end;`

`val it =` ____________________

(f) `- let val x = ref 2 fun f y = x := !x + y in app f s end;`

`val it =` ____________________

(g) `- [[[]]];`

`val it =` ____________________

2. (12 points) Supply a value for `zardoz` that causes the entire expression to evaluate to 312. If impossible, say so and explain why. Example:

```
let
  val zardoz = (312,313)
in
  #1 zardoz
end
```

(a)
```
let
  val zardoz = 
in
  zardoz (zardoz zardoz)
end
```

(b)
```
let
  val zardoz = 
in
  hd (hd (map zardoz [zardoz]))
end
```

(c)
```
let
  val zardoz = 
in
  zardoz ()()()
end
```

(d)
```
let
  val zardoz = 
  fun f n = 5*n + zardoz
in
  List.foldl (op * ) 1 (List.tabulate (zardoz,f))
end
```

3. (9 points)

(a) (8 points) Write a function `split` that, given an integer `i` and a list of elements `s`, splits the list into a triple consisting of the list of elements in `s` before the `i`$^{\text{th}}$ element (starting at 0), the `i`$^{\text{th}}$ element, and the list of elements in `s` after the `i`$^{\text{th}}$ element. Raise `Subscript` if `i` is out of bounds. Examples:

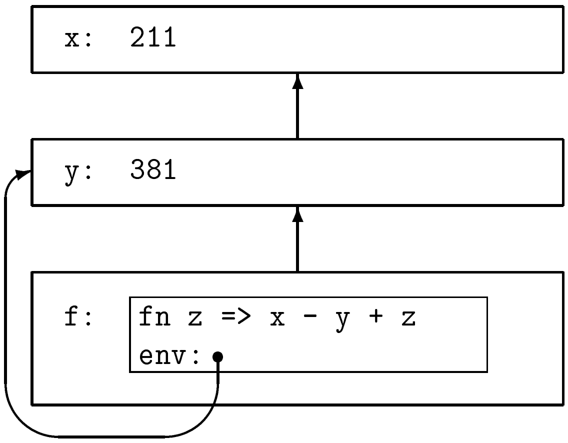
```
- split 2 [5,6,7,8,9];
val it = ([5,6],7,[8,9]) : int list * int * int list
- split 0 ["purple","dinosaur"];
val it = ([],"purple",["dinosaur"]) : string list * string * string list
- split 3 [(),(),()];
uncaught exception subscript out of bounds
  raised at: stdIn:22.23-22.32
```

```
fun split i s =
```

(b) (1 point) What is the type of `split`?

4. (7 points)

(a) (3 points) Give a definition of `f` that results in the following environment.




```
val f =
```

(b) (3 points) Suppose we were to apply `f` to `482` in this environment. Draw the box-and-pointer diagram at the point immediately before the body of the function is evaluated.

(c) (1 point) What would be the value of `f 482`?

```
- f 482;
val it =
```

5. (9 points) Here is a program that computes all subsets of a set. The set is given by a list of distinct elements.

```
fun subsets (x:'a list) : 'a list list =
  case x of
    [] => [[]]
  | x::t =>
      let val st = subsets t
      in st @ (map (fn z => x::z) st)
      end

- subsets ([]:int list);
val it = [[]] : int list list
- subsets [1];
val it = [[],[1]] : int list list
- subsets [1,2];
val it = [[],[2],[1],[1,2]] : int list list
- subsets [1,2,3];
val it = [[],[3],[2],[2,3],[1],[1,3],[1,2],[1,2,3]] : int list list
```

Rewrite this program in tail recursive form. The order of the subsets or the elements in each subset does not matter—your program may produce them in a different order than above if more convenient. You may assume that all the elements of the input list are distinct.

```
fun subsets (x:'a list) : 'a list list = let
  fun subsets' (y:'a list) (out:'a list) : 'a list list =







in
  subsets' x [[]]
end
```

6. (7 points) Prove by induction on n that the program given to you in Problem 5 (not the one you have written!) produces a list of length 2^n when given an input list of length n (by convention, $2^0 = 1$). You may use without proof the facts that `@` produces a list whose length is the sum of the lengths of its two arguments and that `map` produces a list of the same length as its second argument.

7. (3 points) Let `f,g,a,b` be function and constant symbols and `X,Y,Z` variables. Say whether the following pairs of terms unify, and if so, give their most general unifier. Example:

`f(X,a)`	`f(b,Y)`	$\boxed{\texttt{X}\leftarrow\texttt{b},\ \texttt{Y}\leftarrow\texttt{a}}$

(a)	`f(X,g(X,Y))`	`g(X,f(X,Y))`	
(b)	`f(g(X,Y),g(Y,X))`	`f(g(a,a),Z)`	
(c)	`f(X,g(a,X))`	`f(g(b,a),X)`	