



BU CAS CS 320 (FALL SEMESTER, 2006)
CONCEPTS OF PROGRAMMING LANGUAGES

Assignment 7

Out: Tuesday, 7 November 2006
Due: Thursday, 16 November 2006

Total: 60 points

Exercise 1 (10 points) Please implement a function `listAssign` of the following type

```
val listAssign : ('a ref) list * 'a list -> unit
```

When given a list of references r_1, \dots, r_n and a list of values v_1, \dots, v_n , the function `listAssign` assigns v_i to r_i for each $1 \leq i \leq n$. If the number of references does not match the number of values, no assignment is performed and an exception `UnequalLength` needs to be raised.

Exercise 2 (50 points) The following datatype is declared for representing singly-linked lists in SML:

```
datatype 'a sllist = Slnil | SLcons of 'a * ('a sllist) ref
```

Please implement a queue based singly-linked lists. The signature for this implementation is given below:

```
signature QUEUE = sig  
  type 'a queue
```

```
  exception EmptyQueue
```

```
  val make_with_list : 'a list -> 'a queue
```

```
  val empty : 'a queue
```

```
  val is_empty : 'a queue -> bool
```

```
  (* first-in-last-out *)
```

```
  val insert : 'a * 'a queue -> 'a queue
```

```
  val remove : 'a queue -> 'a * 'a queue
```

```
  val foreach : ('a -> unit) -> 'a queue -> unit
```

```
  val fprint :
```

```
    TextIO.outstream -> (TextIO.outstream -> 'a -> unit) -> 'a queue -> unit
```

```
end
```

```
structure Queue :> QUEUE = struct
```

```
  (* your code *)
```

```
end
```