

2005 Paper 12 Question 10

Introduction to Functional Programming

(a) Define a polymorphic datatype `'a seq` for lazy sequences, and define functions `head` and `tail` to return the first element and the rest of the sequence respectively. [1 mark each]

(b) Define a function `pick` with the following type:

```
'a pick : 'a list -> ('a * 'a list) seq
```

which returns a sequence of pairs (x, xs) as in these examples:

(i) `pick [1,2,3,4]` returns a sequence with elements $(1, [2,3,4])$, $(2, [1,3,4])$, $(3, [1,2,4])$, $(4, [1,2,3])$.

(ii) `pick [1,2,1,2]` returns a sequence with elements $(1, [2,1,2])$, $(2, [1,1,2])$, $(1, [1,2,2])$, $(2, [1,2,1])$.

[4 marks]

(c) Define a function `explodeseq` with type

```
explodeseq : 'a list seq -> 'a seq
```

which creates an element in the output sequence from each element in each list of the input sequence. [6 marks]

(d) Define a function `implodeseq` with type

```
implodeseq : int -> 'a seq -> 'a list seq
```

which transforms a sequence of elements into a sequence of lists whose length is specified in the `int` argument. The last list in the output sequence may contain fewer elements. [7 marks]