p13q6 p6q5
MR

Comparative Programming Languages 2001

createco(f, size), creats a coroutine with given stack size and main function. On creation it is left suspended in the cowait call of

```
c : f(cowait()) REPEAT
```

ie it wait for a value.

deleteco(cptr) deletes a given coroutine.

callco(cptr, val) suspends the current coroutine and transfers control to the specified one passing it a value. The current coroutine becomes the parent of the called coroutine.

cowait(val) suspends the currents corourine transferring control to the parent and passing a value.

(b)

```
LET merge(f, g) BE
{ LET p = createco(f, 200)
  LET q = createco(g, 200)
  LET x = callco(p)
  LET y = callco(q)

  { TEST x<y THEN { cowait(x); x := callco(p) }
             THEN { cowait(y); x := callco(q) }
  } REPEAT
}
```

There is room for comments on the exact meaning of the question. eg is the merge meant to be like set union and are the given inifite lists strictly increasing.

(c) Possibly use Java threads but be careful to arrange that only one thread runs at any time.
or
Use classes to define the infinite streams each having a next method.