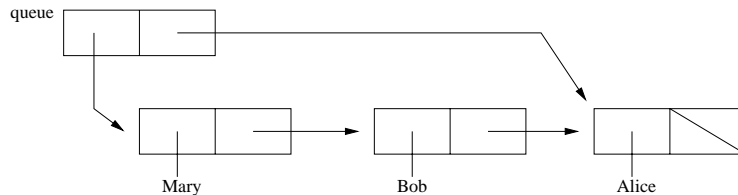
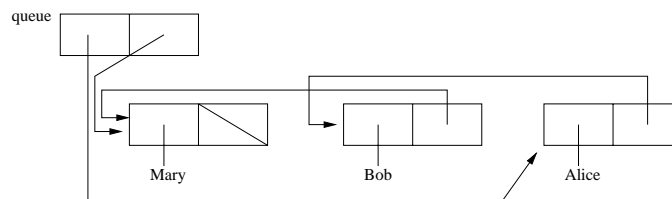


CIS-280
Homework 9:
Due Tuesday, May 17, 2005
60 points

- (10 points) Define a procedure `(substitute-queue! queue item)` that replaces the first element of `queue` with `item`.
- (35 points) Write a procedure `reverse-queue! queue` that returns `queue` with its elements in reverse order. This should be done by mutating the structure so that its links are reversed. In other words, if the original queue is



then the new structure should be



For example,

```
(define q1 (make-queue))
(insert-queue! q1 'Mary)
(insert-queue! q1 'Bob)
(insert-queue! q1 'Alice)
q1
```

The following will be returned as the value of `q1`

```
((mary bob alice) alice)
```

```
(reverse-queue! q1)
q1
```

The following will be returned as the value of `q1`

```
((alice bob mary) mary)
```

- (15 points) A calculator is an object that can handle additions, subtractions, multiplications, divisions, and display. Write a procedure **Make-Calculator** that returns a “calculator object” with a specified value, and which uses the message passing style of programming for operations, where the messages are `'add`, `'subtract`, `'multiply`, and `'look-up`.
 Thus, `(define X (make-calculator 0))` creates a “calculator object” `X` with an initial value of 0.
`((X 'add) 40)` causes 40 to be added to the value of the object `X`.
`((X 'multiply) 3)` causes the value of object `X` to be tripled.
`((X 'look-up))` causes the value of the object `X` to be returned.

Use the procedures in `~carberrry/HW9.scm` — but do not include them when you submit your code on the electronic submission system for Homework 9 since they will be part of the test code.