

; Q1

```
(define (fac n)
  (if (= n 0) 1
      (* n (fac (- n 1)))))

(define (bind k v al)
  (cons (list k v) al))

(define (lookup k al)
  (cond ((null? al) #f)
        ((equal? k (caar al)) (cadar al))
        (else (lookup k (cdr al)))))

(define al '())

(define (fac_mem n)
  (let ((v (lookup n al)))
    (if (equal? v #f)
        (let ((res (fac n)))
          (begin
            (set! al (bind n res al))
            res))
        (begin
          (display "memoization hit\n")
          v))))
```

; Q2

```
(define (build_mem f)
  (let ((al '()))
    (lambda (x)
      (let ((v (lookup x al)))
        (if (equal? v #f)
            (let ((res (f x)))
              (begin
                (set! al (bind x res al))
                res))
            (begin
              (display ``memoization hit\n'')
              v))))))
```

; some tests for Q2

```
(define facm (build_mem fac))
(facm 3)
(facm 5)
(facm 3)

(define (fib n)
  (cond ((= n 0) 0)
        ((= n 1) 1)
        (else (+ (fib (- n 1)) (fib (- n 2))))))
(define fibm (build_mem fib))
(fibm 3)
(fibm 4)
(fibm 3)
```