

(even? n) This predicate tests the even property of a number (returns #t or #f) and preexists in DrScheme.

;; nil is not predefined in DrScheme; define as empty list; page 101

```
(define nil '())
```

```
(define (move-tower size from to extra)
```

```
  (cond ((= size 0) nil)
```

```
        (else (and (move-tower (- size 1) from extra to)
```

```
                  (print-move from to)
```

```
                  (move-tower (- size 1) extra to from))))))
```

```
(define (print-move from to)
```

```
  (and (newline)
```

```
       (display "Move top disk from ")
```

```
       (display from)
```

```
       (display " to ")
```

```
       (display to)))
```

Example: The example better shows $T(n) = \Theta(2^n)$ and space $S(n) = \Theta(n)$

```
(move-tower 2 1 3 2) ;; size != 0, so do else where from=1, to=3, extra=2
```

```
  (move-tower 1 1 2 3) ;; size != 0, so do else where from=1, to=2, extra=3
```

```
    (move-tower 0 1 3 2) ;; size == 0, nil
```

```
    (print-tower 1 2) ;; prints Move top disk from 1 to 2
```

```
  (move-tower 0 3 2 1) ;; size == 0, nil
```

```
(print-tower 1 3) ;; prints Move top disk from 1 to 3
```

```
(move-tower 1 2 3 1) ;; size != 0, so do else where from=2, to=3, extra=1
```

```
  (move-tower 0 2 3 1) ;; size == 0, nil
```

```
  (print-tower 2 3) ;; prints Move top disk from 2 to 3
```

```
(move-tower 0 1 3 2) ;; size == 0, nil
```

From the prompt:

```
> (move-tower 2 1 3 2)
```

```
Move top disk from 1 to 2
```

```
Move top disk from 1 to 3
```

```
Move top disk from 2 to 3()
```