1.

```
(define (first-n lyst num)
  (if (or (= num 0) (null? lyst))
      '()
      (append (list (car lyst)) (first-n (cdr lyst) (- num 1)))))
```

2.

```
(define (part1 lyst)
  (if (null? lyst)
      '()
      (first-n lyst (floor (/ (length lyst) 2)))))
```

3.

```
(define (rest lyst)
  (define (rest-helper count newlyst)
    (if (= count 0)
        newlyst
        (rest-helper (- count 1) (cdr newlyst))))
  (rest-helper (floor (/ (length lyst) 2)) lyst))
```

4.

```
(define (middle-datum lyst)
  (car (rest lyst)))
```

5.

```
(define (part2 lyst)
  (cdr (rest lyst)))
```

6.

```
(define (binary-tree lyst)
  (if (null? lyst)
      tet
      (make-bintree (middle-datum lyst)
                    (binary-tree (part1 lyst))
                    (binary-tree (part2 lyst)))))
```