

```

1 #lang racket
2
3 (define (tail-reverse str)
4   (letrec ((helper (lambda (str acc)
5                     (if (equal? str
6 ""))
7
8 (list->string acc)
9
10 (helper
11 (substring str 1)
12 (cons (string-ref str 0)
13 acc))))))
14 (helper str null)))
15
16 (tail-reverse "mango")
17
18 (define (tail-reverse-2 str)
19   (letrec ((helper (lambda (str acc)
20                     (if (equal? str
21 ""))
22
23 acc
24 (helper
25 (substring str 1)
26 (string-append (string (string-ref

```

```
19 | str 0))
20 |
20 |
20 |
20 | acc))))))
21 |     (helper str "")))
22 |
23 | (tail-reverse-2 "mango")
24 |
```