Welcome to DrRacket, version 7.4 [3m].
Language: racket, with debugging; memory limit: 128 MB.

```
Even Tests -> (iseven? atm)
#t
#f
#t
#f

Odd Tests -> (isodd? atm)
#f
#t
#f
#t

Product Tests -> (prod lst)
120
6000
720
16
256
0

List Multiplication Tests -> (lstmul lst)
'(2 3 4)
'(2 4 6 8)
'(1 2 (3 4))
'(10 20 30 40)
'(100 200 (300 400))
'(1 2 (3 4 (5 6) (7 8) 9) 10 (11) 12)
'()
'(10 20 (10 20) 30 40)
'(2 4 (2 4) 2 4 (2 2 (2 2)) 2 4)

Sum List Tests -> (sumlist lst)
45
12
1010
105
0

Length Tests -> (len lst)
8
1
5
5
12
0

Average Tests -> (average lst)
6
5 1/2
43 4/5
222
3
0
```

```
6
3

Flatten Lists Tests -> (flatten lst)
'(1 2 3 4 5 6)
'(1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16)
'(a c d f d)
'(a d f g e h)
'()
'(1 1 1 1 1 1 1 1 1 1 1 1)

Reverse List Tests -> (revlst lst)
'(1 2 3 4 5 6)
'(1 2 (3 4) (5 6) 7 8)
'(9 87 6 5 4 3 2 1)
'(10 20 30 40)
'()
'(9 8 (7 6 (5 4) 3) 2 1)

Remove First Occurance of Item from List Tests -> (rmFirstOcc lst)
'(2 3 4 3)
'(12 14 87 12 18)
'(1 2 (3 4 (5) 6))
'(2 3 4 3)
'(1 3 5 7 9 11 13)
'(4 7 (5 2) (8 1) (9 2 (3 1)))
'(0 2 (4 6 (8 10) 12) 14 16)
'()
'(10 (20 (30 (40 (50 60) 70) 80) 90) 100)

Minimum Item in List Tests -> (minimum lst)
5
'(2 3)
5
4121
'(9 8)
26
12
21
0

Insertion Sort Tests -> (insertion-sort lst)
'(5 9 8 4 2)
'(5 9 2)
'(5 9 1 2)
'(12 7 3 5 9 11 1 8 10 4 2 6 12 1)

---------------------------------------------------
Square and Cube Program.
Give me a number, and I'll compute its square and cube.

number: 10
the square of 10 is 100
the cube of 10 is 1000
(1 2 3 4 5 6 7 8 9)

list: (1 2 3 4 5 6 7 8 9)
------------------------
```

```
List Stats Program.
length: 9
average 5
minimum 1
sum 45
product 362880
Unsorted List:
(1 2 3 4 5 6 7 8 9)
Sorted List:
(1 2 3 4 5 6 7 8 9)
>
```