

```
1 ; Definition of "if" :
2
3 ; (if <test> <1> <0> )
4
5 (defun my-if (n)
6   (if (evenp n)
7       (/ n 2)
8       (+ 1 (* 3 n)))
9   )
10 )
11
12
13 ; Definition of "cond" :
14
15 ; (cond <test> <1> )
16
17 (defun my-func-2 (n)
18   (cond ( (evenp n) (/ n 2) ) )
19 )
20
21
22
23
24 (defun katlar (n)
25   (cond
26     ( (zerop (rem n 2)) (/ n 2) )
27     ( (zerop (rem n 3)) (/ n 3) )
28     ( (zerop (rem n 5)) (/ n 5) )
29     ( (zerop (rem n 7)) (/ n 7) )
30     ( (zerop (rem n 11)) (/ n 11) )
31   )
32 )
33
34
35
36
37
38 (defun dont-change (n)
39   (cond
40     ( (zerop (rem n 2) ) n )
41     ( t 1 )
42   )
43 )
44
45
46
47
48
49 (defun change (n)
50   (and
51     (numberp n)
52     (integerp n)
53     (or
54       (evenp n)
55       (oddp n)
56     )
57     nil
58   )
59 )
60
```

---

```
61    )  
62  
63  
64  
65  
66  
67  
68  
69  
70
```