```
; Definition of "if" :

; (if <test> <1> <0> )

(defun my-if (n)

  (if (evenp n)
    (/ n 2)
    (+ 1 (* 3 n))
    )
  )

; Definition of "cond" :

; (cond <test> <1> )

(defun my-func-2 (n)

  (cond ( (evenp n) (/ n 2) ) )

  )


(defun katlar (n)

  (cond

     ( (zerop (rem n 2))  (/ n 2) )
     ( (zerop (rem n 3))  (/ n 3) )
     ( (zerop (rem n 5))  (/ n 5) )
     ( (zerop (rem n 7))  (/ n 7) )
     ( (zerop (rem n 11)) (/ n 11) )

     )
  )


(defun dont-change (n)

  (cond

   (    (zerop (rem n 2) )    n   )
   (    t                     1   )

   )

)

(defun change (n)

  (and
      (numberp n)
      (integerp n)
      (or
           (evenp  n)
           (oddp   n)
           )
      nil

      )
```

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