```
#lang racket

(define (next-four-h n spacing i)
  (define next-diag (+ n spacing 1))
  (cond
    [(< i 0) (cons 0 n)]
    [else
     (define next (next-four-h next-diag spacing (- i 1)))
     (cons (+ next-diag (car next)) (cdr next))]))


(define (next-four n spacing)
  (next-four-h n spacing 3))

(define (solve-h n spacing side-len)
  (cond
    [(= n (expt side-len 2)) 0]
    [else
     (define next (next-four n spacing))
     (+ (car next)
        (solve-h (cdr next) (+ 2 spacing) side-len))]))

(define (solve side-len)
  (+ 1
     (solve-h 1 1 side-len)))

(solve 1001)
```