```racket
#lang racket
; Natalia Alonso
; COMP 333 Spring 14
; March 25, 2014

;Problem 1
;(1)
( define cdig
   ( lambda (x)
      ( cond
         ((null? x) #f)
         ((char-numeric? (car x)) #t)
         (else (cdig (cdr x)))
      )
    )
 )

(define contains-digit?
  (lambda (s)
    (cdig (string->list s))))

;Problem 1
;(2)
(define getStringsWithDigits
  (lambda (x)
      (filter contains-digit? x)
      ))

;Problem 2
;(1)
(define prime?
  (lambda (n)
    (cond
      ((any-divisors 2 n) #f)
      (else #t)
  )))

(define any-divisors
  (lambda (x n)
    (cond
      ((= x n) #f)
      ((= (modulo n x) 0) #t)
      (else (any-divisors (+ x 1) n))
)))

;Problem 2
;(2)
(define range
  (lambda (n m)
    (cond
      ((> n m) '())
      (else (cons n (range (+ 1 n) m)))
       )))

;Problem 2
;(3)
'(length (filter prime? (range 2 10000)))
```

```
;Problem 2
;(4)
'(foldl + 0 (map * (filter prime? (range 10 1000)) (filter prime? (range 10 1000))))

;Problem 3
;(1)
(define rowsum2D
  (lambda (k x)
    (cond
      ((> k (length x)) #f)
      (else (rowsum (list-ref x (- k 1))))
      )))

(define rowsum
  (lambda (x)
    (cond
      ((null? x) 0)
      (else (+ (car x) (rowsum (cdr x))))
)))

;Problem 3
;(2)
(define colsum2D
  (lambda (k x)
    (cond
      ((> k (length (car x))) #f)
      (else (colsum (- k 1) x))
      )))

(define colsum
  (lambda (k x)
    (cond
      ((null? x) 0)
      (else (+ (list-ref (car x) k) (colsum k (cdr x))))
)))

;Problem 3
;(3)
(define diagsum
  (lambda (x)
    (cond
      ((not (= (length (car x)) (length x))) "ERROR")
      (else (diag 0 x))
      )))

(define diag
  (lambda (k x)
    (cond
      ((null? x) 0)
      (else (+ (list-ref (car x) k) (diag (+ 1 k) (cdr x))))
      )))

;Test Matrices
(define arrayList1 '((1 2 3 4) (1 2 3 4) (1 2 3 4) (1 2 3 4)))
(define arrayList2 '((2 4 6 8) (1 3 5 7) ( 1 2 3 4)))
```