Q2.1

; Signature: (last-element list)

; Type: [T[] -> T]

; Purpose: return the last element of list

; Pre-conditions: list is not empty

; Tests: (last-element (list 1 3 4)) → 4

Q2.2

; Signature: (power n m)

; Type: [number \* number -> number]

; Purpose: return n^m

; Pre-conditions: m positive number

; Tests: (power 2 4) → 16

Q2.3

; Signature: (sum-lst-power list n)

; Type: [ number[] \* number -> number]

; Purpose: return the sum of all the elements of list in the power of n

; Pre-conditions: n positive

; Tests: (sum-lst-power (list 1 4 2) 3) → 1^3+ 4^3 + 2^3 = 73

Q2.4

; Signature: (num-from-digits list)

; Type: [ number[] -> number]

; Purpose: returns the number consisted from digits in list

; Pre-conditions: list of positive integers

; Tests: (num-from-digits (list 2 4 6)) → 246

Q2.5

; Signature: (is-narcissistic list)

; Type: [ number[] -> boolean]

; Purpose: return if the list of digits that represents a number is narcissistic

; Pre-conditions: list of positive integers

; Tests: (is-narcissistic (list 1 5 3)) → #t