Relación 2

sábado, 30 de septiembre de 2023 13:16

① a) Pre → $Q \equiv \{ V.length > 0 \}$

Impl → boolean comprobar (int[] v) // Devuelve b

Post → $R \equiv \{ b = \exists i \in \mathbb{N}, 0 \le i \le V.length - 1 : V[i] = 0 \}$

b) $Q \equiv \{ V.length > 0 \}$

boolean allPositive(int[] V) // Devuelve b

$R \equiv \{ b = \forall i \in \mathbb{N}, 0 \le i \le V.length - 1 : V[i] > 0 \}$

c) $Q \equiv \{ V.length > 0 \}$

boolean comproba(int[] V, int x) // Devuelve b

$R \equiv \{ b = (N_{i=0}^{V.length-1} V[i] = x) \ge 1 \}$

d) $Q \equiv \{ V.length > 0 \}$

boolean comprobar (int[] V, int x) // Devuelve b

$R \equiv \{ b = (N_{i=0}^{V.length-1} V[i] = x) == 1 \}$

e) $Q \equiv \{ V.length > 0 \}$

boolean comprobar(int[] V, int x) // Devuelve b

$R \equiv \{ b = (N_{i=0}^{V.length-1} V[i] = V[0]) == x \}$

f) $Q \equiv \{ V.length > 0 \}$

boolean comprobar (int[] V, int x) // Devuelve b

$R \equiv \{ b = (\sum_{i=0}^{V.length} V[i]) == x \}$

g) $Q \equiv \{ V.length > 0 \}$

boolean comprobar (int[] V) // Devuelve b

$R \equiv \{ b = (N_{i=0}^{V.length-1} V[i] = i \cdot 2) == V.length \}$

h) $Q \equiv \{ V.length == W.length \}$

boolean comprobar(int[] V, int[] W) // Devuelve b

$R \equiv \{ b = \forall i \in \mathbb{N}, 0 \le i < V.length : V[i] = W[W.length - 1 - i] \}$

i) $Q \equiv \{ V.length > 0 \}$

boolean comprobar(int[] V) // Devuelve b

$R \equiv \{ b = \forall i, j \in \mathbb{N}, 0 \le i, j < V.length, i \neq j : V[i] \neq V[j] \}$

j) $Q \equiv \{ V.length > 0 \}$

boolean comprobar (int[] V)

$R \equiv \{ b = (\exists i \in \mathbb{N}, 0 \le i < V.length : V[i] = 0) \wedge (\exists j \in \mathbb{N}, 0 \le j < V.length : V[j] = 1) \vee (\nexists k, 0 \le k < V.length : V[k] \neq 0 \}$

② a) $Q \equiv \{ True \}$

int todos (int[] V) // Devuelve z

$R \equiv \{ z = (N_{i=0}^{V.length-1} V[i] = 0) \}$

b) $Q \equiv \{ True \}$

int comprobar (int[] V) // Devuelve z

$R \equiv \{ z = (N_{i=0}^{V.length-1} V[i] = V[0]) \}$

c) $Q \equiv \{ True \}$

int suma (int[] V) // Devuelve sum

$R \equiv \{ z = (\sum_{i=0}^{V.length-1} V[i]) \}$

d) $Q \equiv \{ V.length > 0 \}$

int minimo(int[] V) // Devuelve min

$R \equiv \{ (\forall i \in \mathbb{N}, 0 \le i < V.length : min \le V[i]) \wedge (\exists j, 0 \le j < V.length : V[j] = min \}$

e) $Q \equiv \{ V.length > 0 \}$

int comprobar(int[] V, int x) // Devuelve i

$R \equiv \{ (V[i] = x) \wedge (0 \le i < V.length) \wedge ((N_{j=0}^{i} V[j] = x) == 0) \}$

③ a) $Q \equiv \{ V.length > 0 \}$

int frec(int[] V, int x) // frecuencia

$R \equiv \{ frecuencia = (N_{i=0}^{V.length-1} V[i] = x) \}$

b) $Q \equiv \{ V.length > 0 \wedge W.length > 0 \}$

boolean perm(int[] V, int[] W) // c

$R \equiv \{ c = (\forall i \in \mathbb{N}, 0 \le i < V.length : (N_{j=0}^{V.length-1} V[j] = V[i]) == (N_{z=0}^{W.length-1} W[z] = V[i])) \}$

c) $Q \equiv \{ V.length > f \}$

boolean ord(int[] V, int c, int f) // ord_dec

Q ≡ { v.length > 0 }

boolean ord(int[] V, int c, int f) // ord_dec

R ≡ { ord_dec : (∀ i,j ∈ ℕ, c ≤ i,j ≤ f, i<j : v[i] ≥ v[j]) }

④

Q ≡ { num > 1 }

→ boolean esPrimo (int num) // primo

Pos → R ≡ { primo = ($\sum_{i=1}^{num}$ num % i = ø) == 2 }

{ ($\sum_{i=2}^{num-1}$ num % i = ø) == ø }

⑤ → n = m! = m · (m-1) · (m-2) · ... · 1

Q ≡ { n > 0 }

boolean check (int n) // f

R ≡ { f = (∃ m ∈ ℕ : $\left|\prod_{i=0}^{m-1}(m-i)\right|$ == n) }

⑥ [Array ordenado]

Q ≡ { True }

boolean sorted (int[] V) // ok

R ≡ { ok = (∀ i,j ∈ ℕ, 0 ≤ i,j < V.length, i<j : (V[i] ≤ V[j]) ∨ (V[i] ≥ V[j])) }