

#### **4. GAIKO ARIKETAK** **PROGRAMEN ERATORPENA**

|  |    |
|--|----|
| 1) Bektore bateko elementuen batura kalkulatu (1. bertsioa) .....  | 3  |
| 2) Bektore bateko elementuen batura kalkulatu (2. bertsioa) .....  | 3  |
| 3) Bektore bateko elementuen batura kalkulatu (3. bertsioa) .....  | 4  |
| 4) Bektore bateko elementuen batura kalkulatu (4. bertsioa) .....  | 4  |
| 5) Bektore bateko elementuen batura kalkulatu (5. bertsioa) .....  | 5  |
| 6) c aldagaian 0 itzuli A(1..n) taulako posizio denetan 0 badago (1. bertsioa).....  | 5  |
| 7) c aldagaian 0 itzuli A(1..n) taulako posizio denetan 0 badago (2. bertsioa).....  | 6  |
| 8) c aldagaian 0 itzuli A(1..n) taulako posizio denetan 0 badago (3. bertsioa).....  | 6  |
| 9) c aldagaian 0 itzuli A(1..n) taulan gutxienez posizio batean 0 badago (1. bertsioa)   | 7  |
| 10) c aldagaian 0 itzuli A(1..n) taulan gutxienez posizio batean 0 badago (2. bertsioa) .....  | 7  |
| 11) c aldagaian 0 itzuli A(1..n) taulan gutxienez posizio batean 0 badago (3. bertsioa) .....  | 8  |
| 12) Bi bektore berdinak al diren erabaki (1. bertsioa) .....   | 9  |
| 13) Bi bektore berdinak al diren erabaki (2. bertsioa) .....   | 9  |
| 14) Bi bektore berdinak al diren erabaki (3. bertsioa) .....   | 10 |
| 15) Bi bektore berdinak al diren erabaki (4. bertsioa) .....   | 10 |
| 16) Bi bektore berdinak al diren erabaki (5. bertsioa) .....   | 11 |
| 17) x elementua A(1..n) bektorean agertzen al den erabaki (1. bertsioa) .....  | 11 |
| 18) x elementua A(1..n) bektorean agertzen al den erabaki (2. bertsioa) .....  | 12 |
| 19) x elementua A(1..n) bektorean agertzen al den erabaki (3. bertsioa) .....  | 12 |
| 20) x elementua A(1..n) bektorean agertzen al den erabaki (4. bertsioa) .....  | 13 |
| 21) x elementua A(1..n) bektorean agertzen al den erabaki (5. bertsioa) .....  | 13 |
| 22) A(1..n) bektoreko elementu denak bikoitiak al diren erabaki (2008ko ekaina) .  | 14 |
| 23) A(1..n) bektorean gutxienez elementu bikoiti bat ba al dagoen erabaki (2008ko iraila)  | 14 |
| 24) A(1..n) bektorean batekorik ez badago 0 itzuli c aldagaian eta bestela ez itzuli 0 c aldagaian (2009ko ekaina) .....   | 15 |
| 25) A(1..n) bektorean zerorik ez badago n itzuli c aldagaian eta bestela ez itzuli n c aldagaian (2009ko iraila) .....   | 15 |
| 26) 1 eta n-ren arteko posizioen batean A(1..n) eta H(1..n) bektoreek balio bera al duten erabaki z aldagai boolearrean (2010eko ekaina) .....                               | 16 |
| 27) 1 eta n-ren arteko posizioen batean C(1..n) bektoreko balioa A(1..n) eta B(1..n) bektoreetako balioen batura al den erabaki w aldagai boolearrean (2010eko iraila) ..... | 16 |



## Iterazioen eratorpen formala:

### 1) Bektore bateko elementuen batura kalkulatu (1. bertsioa)

$A(1..n)$  bektorea emanda,  $A(1..n)$  bektoreko elementuen batura  $s$  aldagaian kalkulatzeko duen programa eratorri.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><u><b>while</b></u> $\{INB\} \{E\} B?$ <u><b>loop</b></u><br>Aginduak?<br><u><b>end loop;</b></u><br>$\{\psi\} \equiv \{s = \sum_{k=1}^n A(k)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge s = \sum_{k=1}^{i-1} A(k)\}$<br>$E = n + 1 - i$  |

### 2) Bektore bateko elementuen batura kalkulatu (2. bertsioa)

$A(1..n)$  bektorea emanda,  $A(1..n)$  bektoreko elementuen batura  $s$  aldagaian kalkulatzeko duen programa eratorri.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><u><b>while</b></u> $\{INB\} \{E\} B?$ <u><b>loop</b></u><br>Aginduak?<br><u><b>end loop;</b></u><br>$\{\psi\} \equiv \{s = \sum_{k=1}^n A(k)\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge s = \sum_{k=1}^i A(k)\}$<br>$E = n - i$  |

### 3) Bektore bateko elementuen batura kalkulatu (3. bertsioa)

A(1..n) bektorea emanda, A(1..n) bektoreko elementuen batura s aldagaian kalkulatzeko duen programa eratorri.

|  |
|--|
| $\{\phi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\}$ $\{E\}$ B? <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{s = \sum_{k=1}^n A(k)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n) \wedge s = \sum_{k=1}^i A(k)\}$<br>$E = n - i$   |

### 4) Bektore bateko elementuen batura kalkulatu (4. bertsioa)

A(1..n) bektorea emanda, A(1..n) bektoreko elementuen batura s aldagaian kalkulatzeko duen programa eratorri.

|  |
|--|
| $\{\phi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\}$ $\{E\}$ B? <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{s = \sum_{k=1}^n A(k)\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge s = \sum_{k=i+1}^n A(k)\}$<br>$E = i$   |

### 5) Bektore bateko elementuen batura kalkulatu (5. bertsioa)

$A(1..n)$  bektorea emanda,  $A(1..n)$  bektoreko elementuen batura  $s$  aldagaian kalkulatzeko duen programa eratorri.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B? \text{ loop }$<br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{s = \sum_{k=1}^n A(k)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n) \wedge s = \sum_{k=i}^n A(k)\}$<br>$E = i - 1$   |

### 6) $c$ aldagaian 0 itzuli $A(1..n)$ taulako posizio denetan 0 badago (1. bertsioa)

Bakarrik negatiboak ez diren zenbakiak dituen  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizio denetan 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulako posizio denetan 0 ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1 \wedge \text{ezneg}(A(1..n))\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B? \text{ loop }$<br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = 0)\}$ |
| $\{INB\} \equiv \{\text{ezneg}(A(1..n)) \wedge (1 \leq i \leq n + 1) \wedge c = \sum_{k=1}^{i-1} A(k)\}$<br>$E = n + 1 - i$<br>$\text{ezneg}(A(1..n)) \equiv \{\forall k(1 \leq k \leq n \rightarrow A(k) \geq 0)\}$  |

**7)  $c$  aldagaian 0 itzuli  $A(1..n)$  taulako posizio denetan 0 badago (2. bertsioa)**

Bakarrik negatiboak ez diren zenbakiak dituen  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizio denetan 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulako posizio denetan 0 ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1 \wedge \text{ezneg}(A(1..n))\}$<br>Hasieraketak?<br>$\{\text{INB}\}$<br><b><u>while</u></b> $\{\text{INB}\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = 0)\}$ |
| $\{\text{INB}\} \equiv \{\text{ezneg}(A(1..n)) \wedge (0 \leq i \leq n) \wedge c = \sum_{k=1}^i A(k)\}$<br>$E = n - i$<br>$\text{ezneg}(A(1..n)) \equiv \{\forall k(1 \leq k \leq n \rightarrow A(k) \geq 0)\}$  |

**8)  $c$  aldagaian 0 itzuli  $A(1..n)$  taulako posizio denetan 0 badago (3. bertsioa)**

Bakarrik negatiboak ez diren zenbakiak dituen  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizio denetan 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulako posizio denetan 0 ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1 \wedge \text{ezneg}(A(1..n))\}$<br>Hasieraketak?<br>$\{\text{INB}\}$<br><b><u>while</u></b> $\{\text{INB}\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = 0)\}$ |
| $\{\text{INB}\} \equiv \{\text{ezneg}(A(1..n)) \wedge (1 \leq i \leq n) \wedge c = \sum_{k=1}^i A(k)\}$<br>$E = n - i$<br>$\text{ezneg}(A(1..n)) \equiv \{\forall k(1 \leq k \leq n \rightarrow A(k) \geq 0)\}$  |

**9)  $c$  aldagaian 0 itzuli  $A(1..n)$  taulan gutxienez posizio batean 0 badago (1. bertsioa)**

Zenbaki osozko  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizioen batean 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulan 0rik ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\}$ $\{E\}$ B? <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = 0)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge c = \prod_{k=1}^{i-1} A(k)\}$<br>$E = n + 1 - i$  |

**10)  $c$  aldagaian 0 itzuli  $A(1..n)$  taulan gutxienez posizio batean 0 badago (2. bertsioa)**

Zenbaki osozko  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizioen batean 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulan 0rik ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\}$ $\{E\}$ B? <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = 0)\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge c = \prod_{k=1}^i A(k)\}$<br>$E = n - i$  |

**11)  $c$  aldagaian 0 itzuli  $A(1..n)$  taulan gutxienez posizio batean 0 badago (3. bertsioa)**

Zenbaki osozko  $A(1..n)$  bektorea emanda,  $A(1..n)$  taulako posizioen batean 0 baldin badago, osoa den  $c$  aldagaian 0 itzuliko duen eta  $A(1..n)$  taulan 0rik ez badago,  $c$  aldagaian 0 itzuliko ez duen programa eratorri.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\}$ $\{E\}$ B? <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop</u></b> ;<br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = 0)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n) \wedge c = \prod_{k=1}^i A(k)\}$<br>$E = n - i$   |



## 12) Bi bektore berdinak al diren erabaki (1. bertsioa)

$A(1..n)$  eta  $B(1..n)$  bektoreak emanda  $c$  aldagai boolearrean  $A(1..n)$  eta  $B(1..n)$  berdinak al diren ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeko.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k))\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge (c \leftrightarrow \forall k(1 \leq k \leq i - 1 \rightarrow A(k) = B(k)))\}$<br>$E = n + 1 - i$  |

## 13) Bi bektore berdinak al diren erabaki (2. bertsioa)

$A(1..n)$  eta  $B(1..n)$  bektoreak emanda  $c$  aldagai boolearrean  $A(1..n)$  eta  $B(1..n)$  berdinak al diren ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeko.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k))\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge (c \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) = B(k)))\}$<br>$E = n - i$  |

#### 14) Bi bektore berdinak al diren erabaki (3. bertsioa)

$A(1..n)$  eta  $B(1..n)$  bektoreak emanda  $c$  aldagai boolearrean  $A(1..n)$  eta  $B(1..n)$  berdinak al diren ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k))\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n) \wedge (c \leftrightarrow \forall k(1 \leq k \leq i \rightarrow A(k) = B(k)))\}$<br>$E = n - i$  |

#### 15) Bi bektore berdinak al diren erabaki (4. bertsioa)

$A(1..n)$  eta  $B(1..n)$  bektoreak emanda  $c$  aldagai boolearrean  $A(1..n)$  eta  $B(1..n)$  berdinak al diren ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k))\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge (c \leftrightarrow \forall k(i \leq k \leq n \rightarrow A(k) = B(k)))\}$<br>$E = i - 1$  |

### 16) Bi bektore berdinak al diren erabaki (5. bertsioa)

$A(1..n)$  eta  $B(1..n)$  bektoreak emanda  $c$  aldagai boolearrean  $A(1..n)$  eta  $B(1..n)$  berdinak al diren ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \forall k(1 \leq k \leq n \rightarrow A(k) = B(k))\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge (c \leftrightarrow \forall k(i + 1 \leq k \leq n \rightarrow A(k) = B(k)))\}$<br>$E = i$  |

### 17) x elementua $A(1..n)$ bektorean agertzen al den erabaki (1. bertsioa)

$x$  zenbaki bat eta  $A(1..n)$  bektorea emanda  $c$  aldagai boolearrean  $x$  elementua  $A(1..n)$  bektorean agertzen al den ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = x)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge (c \leftrightarrow \exists k(1 \leq k \leq i - 1 \wedge A(k) = x))\}$<br>$E = n + 1 - i$  |

### 18) x elementua A(1..n) bektorean agertzen al den erabaki (2. bertsioa)

x zenbaki bat eta A(1..n) bektorea emanda  $c$  aldagai boolearrean x elementua A(1..n) bektorean agertzen al den ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = x)\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge (c \leftrightarrow \exists k(1 \leq k \leq i \wedge A(k) = x))\}$<br>$E = n - i$  |

### 19) x elementua A(1..n) bektorean agertzen al den erabaki (3. bertsioa)

x zenbaki bat eta A(1..n) bektorea emanda  $c$  aldagai boolearrean x elementua A(1..n) bektorean agertzen al den ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = x)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n) \wedge (c \leftrightarrow \exists k(1 \leq k \leq i \wedge A(k) = x))\}$<br>$E = n - i$  |

## 20) x elementua A(1..n) bektorean agertzen al den erabaki (4. bertsioa)

x zenbaki bat eta A(1..n) bektorea emanda  $c$  aldagai boolearrean x elementua A(1..n) bektorean agertzen al den ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkua izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = x)\}$ |
| $\{INB\} \equiv \{(1 \leq i \leq n + 1) \wedge (c \leftrightarrow \exists k(i \leq k \leq n \wedge A(k) = x))\}$<br>$E = i - 1$  |

## 21) x elementua A(1..n) bektorean agertzen al den erabaki (5. bertsioa)

x zenbaki bat eta A(1..n) bektorea emanda  $c$  aldagai boolearrean x elementua A(1..n) bektorean agertzen al den ala ez erabakitzen duen programa eratorri. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkua izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

|  |
|--|
| $\{\varphi\} \equiv \{n \geq 1\}$<br>Hasieraketak?<br>$\{INB\}$<br><b><u>while</u></b> $\{INB\} \{E\} B?$ <b><u>loop</u></b><br>Aginduak?<br><b><u>end loop;</u></b><br>$\{\psi\} \equiv \{c \leftrightarrow \exists k(1 \leq k \leq n \wedge A(k) = x)\}$ |
| $\{INB\} \equiv \{(0 \leq i \leq n) \wedge (c \leftrightarrow \exists k(i + 1 \leq k \leq n \wedge A(k) = x))\}$<br>$E = i$  |

## 22) A(1..n) bektoreko elementu denak bikoitiak al diren erabaki (2008ko ekaina)

A(1..n) bektorea emanda  $p$  aldagai boolearrean A(1..n) bektoreko elementu denak bikoitiak al diren ala ez erabakitzen duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna ezezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

```
{φ} ≡ {n ≥ 1}
Hasieraketak?
while {INB} B? loop
    Aginduak?
end loop;
{ψ} ≡ {p ↔ ∀k(1 ≤ k ≤ n → bikoitia(A(k)))}
```

---

```
{INB} ≡ {(0 ≤ i ≤ n) ∧ p ↔ ∀k(1 ≤ k ≤ i → bikoitia (A(k)))}
```

```
E = n - i
```

```
bikoitia(x) ≡ {x mod 2 = 0}
```

## 23) A(1..n) bektorean gutxienez elementu bikoiti bat ba al dagoen erabaki (2008ko iraila)

A(1..n) bektorea emanda,  $p$  aldagai boolearrean A(1..n) bektorean gutxienez elementu bikoitiren bat ba al dagoen ala ez erabakitzen duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

```
{φ} ≡ {n ≥ 1}
Hasieraketak?
while {INB} B? loop
    Aginduak?
end loop;
{ψ} ≡ {p ↔ ∃k(1 ≤ k ≤ n ∧ bikoitia(A(k)))}
```

---

```
{INB} ≡ {(1 ≤ i ≤ n + 1) ∧ p ↔ ∃k(i ≤ k ≤ n ∧ bikoitia(A(k)))}
```

```
E = i - 1
```

```
bikoitia(x) ≡ {x mod 2 = 0}
```

**24) A(1..n) bektorean batekorik ez badago 0 itzuli c aldagaian eta bestela ez itzuli 0 c aldagaian (2009ko ekaina)**

Zeroak eta batekoak bakarrik dituen A(1..n) bektorea emanda, A(1..n) taulan bateko kopurua 0 baldin bada, osoa den c aldagaian 0 itzuliko duen eta A(1..n) taulako bateko kopurua 0 ez bada, c aldagaian 0 itzuliko ez duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1 \wedge \text{bitak}(A(1..n))\}$<br>Hasieraketak?<br>$\{\text{INB}\}$<br><b>while</b> {INB} B? <b>loop</b><br>Aginduak?<br><b>end loop;</b><br>$\{\psi\} \equiv \{(c = 0) \leftrightarrow (\text{Nk}(1 \leq k \leq n \wedge A(k) = 1) = 0)\}$<br><hr/> $\{\text{INB}\} \equiv \{\text{bitak}(A(1..n)) \wedge (0 \leq i \leq n) \wedge c = \sum_{k=1}^i A(k)\}$<br><br>$E = n - i$<br><br>$\text{bitak}(A(1..n)) \equiv \{\forall k(1 \leq k \leq n \rightarrow (A(k) = 0 \vee A(k) = 1))\}$ |
|---|

**25) A(1..n) bektorean zerorik ez badago n itzuli c aldagaian eta bestela ez itzuli n c aldagaian (2009ko iraila)**

Zeroak eta batekoak bakarrik dituen A(1..n) bektorea emanda, A(1..n) taulan zero-kopurua 0 baldin bada, osoa den c aldagaian n balioa itzuliko duen eta A(1..n) taulako zero-kopurua 0 ez bada, c aldagaian n itzuliko ez duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz. Eratorritako programak  $\varphi$ ,  $\psi$ , INB eta E-rekiko zuzena izan behar du.

|   |
|---|
| $\{\varphi\} \equiv \{n \geq 1 \wedge \text{bitak}(A(1..n))\}$<br>Hasieraketak?<br>$\{\text{INB}\}$<br><b>while</b> {INB} B? <b>loop</b><br>Aginduak?<br><b>end loop;</b><br>$\{\psi\} \equiv \{(c = n) \leftrightarrow (\text{Nk}(1 \leq k \leq n \wedge A(k) = 0) = 0)\}$<br><hr/> $\{\text{INB}\} \equiv \{\text{bitak}(A(1..n)) \wedge (1 \leq i \leq n) \wedge c = \sum_{k=1}^i A(k)\}$<br><br>$E = n - i$<br><br>$\text{bitak}(A(1..n)) \equiv \{\forall k(1 \leq k \leq n \rightarrow (A(k) = 0 \vee A(k) = 1))\}$ |
|---|

**26) 1 eta n-ren arteko posizioen batean A(1..n) eta H(1..n) bektoreek balio bera al duten erabaki z aldagai boolearrean (2010eko ekaina)**

Zenbaki osozko A(1..n) eta H(1..n) bektoreak emanda, z aldagai boolearrean 1 eta n-ren arteko posizioen batean A(1..n) eta H(1..n) bektoreek balio bera ba al duten erabakitzen duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

```

{φ} ≡ {n ≥ 1}
Hasieraketak?
{INB}
while {INB} B? loop
    Aginduak?
end loop;
{ψ} ≡ {z ↔ ∃k(1 ≤ k ≤ n ∧ A(k) = H(k))}
-----
{INB} ≡ {(0 ≤ i ≤ n) ∧ z ↔ ∃k(1 ≤ k ≤ i ∧ A(k) = H(k))}

E = n - i

```

**27) 1 eta n-ren arteko posizioen batean C(1..n) bektoreko balioa A(1..n) eta B(1..n) bektoreetako balioen batura al den erabaki w aldagai boolearrean (2010eko iraila)**

Zenbaki osozko A(1..n), B(1..n) eta C(1..n) bektoreak emanda, w aldagai boolearrean C(1..n) bektoreko posizioen bateko elementua A(1..n) eta B(1..n) bektoreetako posizio horretako elementuen batura al den erabakitzen duen programa eratorri Hoare-ren kalkuluko While-aren Erregela eta Esleipenaren Axioma erabiliz. Lortutako programak eraginkorra izan beharko du, hau da, une batean erantzuna baiezkoa izango dela konturatuz gero, programak bukatu egin beharko du gainontzeko posizioak aztertzeke.

```

{φ} ≡ {n ≥ 1}
Hasieraketak?
{INB}
while {INB} B? loop
    Aginduak?
end loop;
{ψ} ≡ {w ↔ ∃k(1 ≤ k ≤ n ∧ C(k) = A(k) + B(k))}
-----
{INB} ≡ {(1 ≤ i ≤ n) ∧ w ↔ ∃k(1 ≤ k ≤ i ∧ C(k) = A(k) + B(k))}

E = n - i

```