42. (2009ko iraila) ezzerobat(C(1..r)), bitak(C(1..r)) eta ordezkatuta(D(1..q), (d_1, d_2, …, d_q), E(1..q), (e_1, e_2, …, e_q), pos) predikatuak eta zerorik eta batekorik ez duen A(1..n) bektorea eta bakarrik zeroak eta batekoak dituen B(1..n) bektorea emanda, B(1..n) bektoreko zero bakoitza eta A(1..n) bektoreko posizio bereko elementua trukatzen dituen programa. -- #

a) **ezzerobat(C(1..r))** $\equiv \{\forall k\ (1 \le k \le r \rightarrow (C(k) \ne 0 \wedge C(k) \ne 1))\}$

b) **bitak(C(1..r))** $\equiv \{\forall k\ (1 \le k \le r \rightarrow (C(k) = 0 \vee C(k) = 1))\}$

c) **ordezkatuta(D(1..q), (d_1, d_2, …, d_q), E(1..q), (e_1, e_2, …, e_q), pos)** $\equiv$
$\{(0 \le pos \le q) \wedge$
$\forall k\ (1 \le k \le pos \rightarrow ((e_k = 0 \rightarrow (D(k) = 0 \wedge E(k) = d_k)) \wedge$
$\wedge\ (e_k = 1 \rightarrow (D(k) = d_k \wedge E(k) = e_k))))\}$

d) Asertzioak ematerakoan egokiena den ordena jarraituko da:

(1) {Hasierako baldintza} $\equiv \{n \ge 1 \wedge \text{ezzerobat}(A(1..n)) \wedge \text{bitak}(B(1..n)) \wedge$
$\forall k\ (1 \le k \le n \rightarrow (A(k) = a_k \wedge B(k) = b_k))\}$

Hasierako baldintzaren bidez A eta B bektoreek gutxienez elementu bana izango dutela, A(1..n) bektorean zerorik eta batekorik ez dagoela, B(1..n) bektorean zeroak eta batekoak bakarrik daudela eta A(1..n) eta B(1..n) bektoreetako hasierako balioak *a* eta b minuskulen bidez eta dagozkien azpiindezeak erabiliz adieraziko ditugula esaten da.

(2) {Tarteko asertzioa} $\equiv \{(1) \wedge i = 0\}$

(9) {Bukaerako baldintza} $\equiv$
{ordezkatuta(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), n)}

Bukaerako baldintzaren bidez bektore osoan, hau da, n posizioraino egin beharreko ordezkaketa denak eginda daudela esaten da.

(3) {Inbariantea} $\equiv \{(0 \le i \le n) \wedge$
ordezkatuta(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), i)}

Inbariantearen bidez i posizioraino (posizio hori ere barne) egin beharreko aldaketa denak eginda daudela adierazten da. Berriz while-an sartzen bagara, orain i aldagaiak erakusten duen posizioaren hurrengo posizioko elementua aztertuko da orduan.

(4) {Tarteko asertzioa} $\equiv \{(0 \le i \le n - 1) \wedge$
ordezkatuta(A(1..n), (a_1, a_2, …, a_n), B(1..n), (b_1, b_2, …, b_n), i)}

while-ean sartu garenez badakigu while-aren baldintza bete egin dela eta i ez dela n.

(5) {Tarteko asertzioa} ≡ $\{(1 \le i \le n) \wedge$
$\text{ordezkatuta}(A(1..n), (a_1, a_2, \ldots, a_n), B(1..n), (b_1, b_2, \ldots, b_n), i-1)\}$

Aurreko asertzioan, hau da, (4) puntuan, i posizioraino (i posizioa ere barne) egin beharreko aldaketa denak egin direla esan da. Orain i aldagairen balioa handitu egin da, bat gehiago balio du, baina i aldagaiak orain erakusten edo zehazten duen posizio berria ez da aztertu, beraz ordezkaketek ez dute aurrera egin, orain ordezkaketak i – 1 posizioraino daude eginda. Bestalde (4) puntuan i aldagaiaren balioa 0 eta n – 1 balioen artekoa dela esan da baina orain i-ren balioa 1 handitu denez, i-ren balio hori 1 eta n-ren artekoa izango dela ziurta dezakegu.

(6) {Tarteko asertzioa} ≡ $\{(1 \le i \le n) \wedge$
$\text{ordezkatuta}(A(1..n), (a_1, a_2, \ldots, a_n), B(1..n), (b_1, b_2, \ldots, b_n), i-1)$
$\wedge B(i) = 0 \wedge B(i) = b_i \wedge A(i) = a_i\}$

if aginduaren **then** aukeratik sartu garenez, badakigu B taulako i posizioa zero dagoela, hau da, b_i balioa 0 dela. Bestalde aldaketarekin hastera goazenez A(i) posizioan hasierako balioa, hau da, a_i balioa dagoela gogoratzea komeni da.

(6) era laburrean:
(6) ≡ $\{(5) \wedge B(i) = 0 \wedge B(i) = b_i \wedge A(i) = a_i\}$

(7) {Tarteko asertzioa} ≡ $\{(1 \le i \le n) \wedge$
$\text{ordezkatuta}(A(1..n), (a_1, a_2, \ldots, a_n), B(1..n), (b_1, b_2, \ldots, b_n), i-1)$
$\wedge b_i = 0 \wedge B(i) = A(i) \wedge A(i) = a_i\}$

B(i) : = A(i); esleipena burutu ondoren badakigu b_i (B(i)-ko hasierako balioa) 0 dela eta orain B(i) eta A(i) balioak a_i direla (A(i)-ren hasierako balioa). Baina esleipen hori egin arren aldaketa osoak i – 1 posizioraino bakarrik daude eginda, i posizioan erdizka gaude eta horregatik *ordezkatuta* predikatuan i – 1 ipini behar da.

(7) era laburrean:
(7) ≡ $\{(5) \wedge b_i = 0 \wedge B(i) = A(i) \wedge A(i) = a_i\}$
(6) puntua ezin da erabili.

(8) {Tarteko asertzioa} $\equiv$ {$(1 \leq i \leq n) \wedge$
ordezkatuta(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, i – 1)
$\wedge\ b_i = 0 \wedge B(i) = a_i \wedge A(i) = 0$ }

A(i) : = 0; esleipena burutu ondoren badakigu b_i (B(i)-ren hasierako balioa) 0 dela eta orain B(i)-ren balioa a_i dela eta A(i)-rena 0 dela.

(8) era laburrean:
(8) $\equiv$ {(5) $\wedge\ b_i = 0 \wedge B(i) = a_i \wedge A(i) = 0$ }
(6) eta (7) puntuak ezin dira erabili.

Beste aukera bat ere badago. Izan ere ordezkatuta(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, i – 1) predikatuak dio 1 eta i – 1 -posizioen arteko kalkuluak eginda daudela eta $b_i = 0 \wedge B(i) = a_i \wedge A(i) = 0$ formula kontuan hartuz badakigu i posiziokoa ere eginda dagoela, beraz *ordezkatuta* predikatuan i ipiniz 1 eta i posizioen arteko kalkuluak eginda daudela adieraz dezakegu.

(8) $\equiv$ {$(1 \leq i \leq n) \wedge$
ordezkatuta(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, i) $\wedge\ b_i = 0$ }
Kasu honetan $\wedge\ B(i) = a_i \wedge A(i) = 0$ ipini beharrik ez dago, hori predikatuan i ipintzerakoan esanda gelditzen delako.

(11) {Tarteko asertzioa} $\equiv$ {$(1 \leq i \leq n) \wedge$
ordezkatuta(A(1..n), $(a_1, a_2, \ldots, a_n)$, B(1..n), $(b_1, b_2, \ldots, b_n)$, i)}

(11) puntua end if eta end loop-en artean betetzen den asertzioa da. Puntu honetan ez dakigu then bidetik joan al garen ala ez eta horregatik ezin dugu $b_i = 0 \wedge B(i) = a_i \wedge A(i) = 0$ ipini.

(10) E = n – i

Inbariantea betetzen den lekuan gauden bakoitzean E espresioak while agindua bukatzeko zenbat buelta falta diren adierazi behar du. Taula ezkerretik eskuinera zeharkatzen denean E espresioa "i aldagaiak hartuko duen azkeneko balioa" ken "i" izango da. Azken batean E espresioa n eta i-ren arteko distantzia da. Horrela i-ren balioa handitzen denean, n eta i-ren arteko distantzia txikiagoa izango da eta eman beharreko buelta-kopurua ere txikiagoa izango da.

Koloreen bidez asertzio batetik bestera dauden aldaketak nabarmendu dira.