2. GAIKO ARIKETAK
PROGRAMEN ESPEZIFIKAZIOA ETA DOKUMENTAZIOA

AURKIBIDEA

a) Formulak, predikatuak eta aldagai aske eta lotuak

Enuntziatu bakoitzeko honako hau egin behar da:

- Enuntziatuko esaldiak dioena adierazten duen formula idatzi lehen mailako logikaren lengoaia erabiliz.
- Idatzitako formulan aldagai askeak eta lotuak zein diren zehaztu.
- Formulari dagokion predikatua eman.

Ariketa bakoitzean definitu beharreko predikatuaren izena zein den eta aurretik definitutako predikaturik erabili behar al den ala ez adierazten da.

1. **x zenbaki bikoitia da (bikoitia)**
    - ➢ Predikatuaren izena: **bikoitia**
    - ➢ Aurretik definitutako predikaturik erabiltzeke egin.

2. **x zenbaki bakoitia da (bakoitia)**
    - ➢ Predikatuaren izena: **bakoitia**
    - ➢ Bi eratara egin:
        - (i) 1. ariketako predikatua erabiliz.
        - (ii) 1. ariketako predikatua erabiltzeke.

3. **x zenbaki osoa w zenbakiaren berredura oso bat da (berredura)**
   x zenbakia w zenbakiaren berredura bezala idaztea ahalbidetzen duen zenbaki positibo bat existitzen da. (zenbaki positiboren batentzat x balioa w balioa ber zenbaki hori da)
    - ➢ Predikatuaren izena: **berredura**
    - ➢ Aurretik definitutako predikaturik erabiltzeke egin.

4. **x zenbaki osoa 2 zenbakiaren berredura bat da (biber)**
    - ➢ Predikatuaren izena: **biber**
    - ➢ Bi eratara egin:
        - (i) 3. ariketako predikatua erabiliz.
        - (ii) 3. ariketako predikatua erabiltzeke. (x zenbakia w zenbakiaren berredura bezala idaztea ahalbidetzen duen zenbaki ez negatibo bat existitzen da) (zenbaki ez negatiboren batentzat x balioa 2 balioa ber zenbaki hori da)

5. **A(1..n) bektoreko elementu denak x baino handiagoak dira (denakhand)**
   A(1..n) bektoreko posizio denetan x baino handiagoa den zenbaki bat dago.
    - ➢ Predikatuaren izena: **denakhand**
    - ➢ Aurretik definitutako predikaturik erabiltzeke egin.

6. A(1..n) bektoreko elementuren bat positiboa da (> 0) (positiborenbat)

Positiboa den balioa duen posizio bat gutxienez existitzen da A(1..n) bektorean.

- Predikatuaren izena: **positiborenbat**
- Aurretik definitutako predikaturik erabiltzeke egin.

7. A(1..n) bektoreko elementu denak positiboak dira (> 0) (denakpos)

- Predikatuaren izena: **denakpos**
- Bi eratara egin:
  (i) 5. ariketako predikatua erabiliz.
  (ii) 5. ariketako predikatua erabiltzeke. (A(1..n) bektoreko posizio denetan balio positiboa dago)

8. A(1..n) bektorean digituak bakarrik daude (0 eta 9ren arteko zenbakiak) (digituak)

A(1..n) bektoreko posizio denetan 0 eta 9ren arteko zenbaki bat dago.

- Predikatuaren izena: **digituak**
- Aurretik definitutako predikaturik erabiltzeke egin.

9. A(1..n) bektorean x elementu bikoiti daude (bikop)

A(1..n) bektorean balio bikoitia duten posizio-kopurua x da.

- Predikatuaren izena: **bikop**
- 1. ariketako predikatua erabiliz egin.

10. A(1..n) bektoreko elementu denak bikoitiak dira (denakbik)

- Predikatuaren izena: **denakbik**
- Bi eratara egin:
  (i) 9. ariketako predikatua erabiliz.
  (ii) 1. ariketako predikatua erabiliz. (A(1..n) bektoreko posizio denetan balio bikoitia dago)

11. p aldagai boolearrak A(1..n) bektoreko elementu denak bikoitiak al diren ala ez adierazten du (denakbikdira)

- Predikatuaren izena: **denakbikdira**
- Bi eratara egin:
  (i) 10. ariketako predikatua erabiliz.
  (ii) 1. ariketako predikatua erabiliz. (p-ren balioa True izando da baldin eta soilik baldin A(1..n) bektoreko posizio denetan balio bikoiti bat badago)

12. A(1..n) bektoreko pos1 eta pos2 posizioen artean x balioa v aldiz agertzen da (aldiztartean)

A(1..n) bektorean pos1 eta pos2 balioek hutsa izan daitekeen sekzio bat definitzen dute (hau da, pos1 balioa 1 eta n + 1-en artean dago eta pos2 0 eta n-ren artean dago) eta x balioa v aldiz agertzen da pos1 eta pos2-ren artean dauden A(1..n) bektoreko posizioetan (pos1 eta pos2 ere kontuan hartuz).

- Predikatuaren izena: aldiztartean
- Aurretik definitutako predikaturik erabiltzeke egin. (pos1 balioa 1 eta n + 1-en arteko balio bat da, pos2 balioa 0 eta n-ren arteko balio bat da eta A(1..n) bektorean x balioa duten pos1 eta pos2-ren arteko posizio-kopurua v da)

13. x balioa v aldiz agertzen da A(1..n) bektorean (aldiz)

- Predikatuaren izena: **aldiz**
- Bi eratara egin:
  - (i) 12. ariketako predikatua erabiliz.
  - (ii) Aurretik definitutako predikaturik erabiltzeke. (A(1..n) bektorean x balioa duten posizioen kopurua v da)

14. A(1..n) bektorean pos1 eta pos2 posizioen arteko elementu denak desberdinak dira (denakdesb)

- Predikatuaren izena: **denakdesb**
- Bi eratara egin:
  - (i) 12. ariketako predikatua erabiliz. (pos1 1 baino handiagoa edo berdina eta n + 1 baino txikiagoa edo berdina da, pos2 0 baino handiagoa edo berdina eta n baino txikiagoa edo berdina da eta A(1..n) bektorean pos1 eta pos2-ren arteko posizio denentzat posizio horretako elementua behin bakarrik agertzen da tarte horretan) (A(1..n) bektorean pos1 eta pos2-ren arteko edozein posizio hartuta, posizio horretako elementua behin bakarrik agertzen da tarte horretan)
  - (ii) Aurretik definitutako predikaturik erabiltzeke. (pos1 1 baino handiagoa edo berdina eta n + 1 baino txikiagoa edo berdina da, pos2 0 baino handiagoa edo berdina eta n baino txikiagoa edo berdina da eta A(1..n) bektorean pos1 eta pos2-ren arteko posizio denentzat, tarte horretako edozein posizio desberdin hartuz edukia ere desberdina izando da)

15. x balioa w balioa baino gehiagotan agertzen da A(1..n) bektorean (gehiagotan)

- Predikatuaren izena: **gehiagotan**
- Aurretik definitutako predikaturik erabiltzeke egin (A(1..n) bektorean x balioa duten posizioen kopurua w balioa duten posizioen kopurua baino handiagoa da)

16. x eta w desberdinak dira eta A(1..n) bektorean biak kopuru berean agertzen dira (kopurubera)

- Predikatuaren izena: **kopurubera**
- Aurretik definitutako predikaturik erabiltzeke egin. (x eta w desberdinak dira eta A(1..n) bektorean x balioa duten posizioen kopurua eta w balioa duten posizioen kopururen berdina da)

17. A(1..n) bektorean gutxienez bi posizioetan balioak berdinak dira (gutxienezbi)

- Predikatuaren izena: **gutxienezbi**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko posizioren batentzat balio bera duen beste posizioren bat existitzen da)

18. A(1..n) bektorean justu balio bera duten bi posizio daude(justubi)

- Predikatuaren izena: **justubi**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektorean justu bi posiziorentzat balio bera duen posizio desberdin bat existitzen da)

19. A(1..n) bektorean balio denak gutxienez bi aldiz agertzen dira (denakbigehi)

- Predikatuaren izena: **denakbigehi**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko posizio denentzat balio bera duen posizio desberdin bat existitzen da) (A(1..n) bektoreko edozein posizio hartuta, balio bera duen posizio desberdinen bat existitzen da)

20. A(1..n) bektorean balio denak justu bi aldiz agertzen dira (denakjustubi)

- Predikatuaren izena: **denakjustubi**
- Bi eratara egin:
  (i) 13. ariketako predikatua erabiliz. (A(1..n) bektoreko posizio denentzat, posizio horretako balioa 2 aldiz agertzen da A(1..n) bektorean) (A(1..n) bektoreko edozein posizio hartuta, posizio horretako balioa 2 aldiz agertzen da A(1..n) bektorean)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (A(1..n) bektoreko posizio denentzat, posizio hori kontatu gabe posizio horretako balio bera duten posizio-kopurua 1 da) (A(1..n) bektoreko edozein posizio hartuta, posizio hori kontatu gabe posizio horretako balio bera duten posizio-kopurua 1 da)

21. x zenbakia lehena da (lehena)

x zenbaki bat lehena izango da 1 baino handiagoa edo berdina bada eta justu bi zatitzaile baditu (1 ez da lehena).

- Predikatuaren izena: **lehena**
- Aurretik definitutako predikaturik erabiltzeke egin. (x balioa 1 baino handiagoa edo berdina da eta x hondarrik sortu gabe zatitzen duten 1 eta x balioen arteko zenbaki-kopurua 2 da (1 eta x barne))

22. z balioa 1etik hasi eta jarraian doazen zenbakiz osatutako segida bateko elementuen batura da (batujarraian)

Adibideak: 10 = 1 + 2 + 3 + 4
9 zenbakia ezin da idatzi era horretako batura bat bezala

- Predikatuaren izena: **batujarraian**
- Aurretik definitutako predikaturik erabiltzeke egin. (1 edo handiagoa den zenbakiren batentzat 1etik zenbaki horretarainoko zenbakien batura z da)

23. (tartebatura)

x eta w balioek hutsa ez den azpitarte bat definitzen edo mugatzen dute [1..n] tartean eta s balioa A(1..n) bektorean x eta w posizioen artean dauden elementuen batura da (biak barne).

Adibidea:

x = 3 eta w = 6 direla suposatuz, s letra lodiz dauden elementuen batura izango litzateke.

A

90	22	**1**	**1**	**1**	**-70**	62	9
1	2	3	4	5	6	7	8

s = 1 + 1 + 1 + (-70)

- Predikatuaren izena: **tartebatura**
- Aurretik definitutako predikaturik erabiltzeke egin. (x 1 baino handiagoa edo berdina eta w baino txikiagoa edo berdina da, w x baino handiagoa edo berdina eta n baino txikiagoa edo berdina da eta A(1..n) bektorean x eta w posizioen artean dauden posizioetako elementuen batura s-ren berdina da)

24. (tartebatura2)

x eta w balioek hutsa ez den azpitarte bat definitzen edo mugatzen dute [1..n] tartean eta n balioa A(1..n) bektorean x eta w posizioen artean dauden elementuen batura da (biak barne).

Adibidea:

x = 3 eta w = 6 direla suposatuz, n (hau da 8) letra lodiz dauden elementuen batura izango litzateke.

A

90	22	**6**	**-4**	**5**	**1**	62	9
1	2	3	4	5	6	7	8

n = 6 + (-4) + 5 + 1

- Predikatuaren izena: tartebatura2
- 23. ariketako predikatua erabiliz.

25. s balioa A(1..n) bektoreko posizio bikoitietako elementuen batura da (batuposbik)

Adibidea:

A

90	**22**	1	**1**	1	**-70**	62	**9**
1	2	3	4	5	6	7	8

s = 22 + 1 + (-70) + 9

- Predikatuaren izena: **batuposbik**
- 1. ariketako predikatua erabiliz.

26. s balioa A(1..n) bektoreko elementu bikoitien batura da (batubik)

Adibidea:

A

90	**22**	1	1	1	**-70**	**62**	9
1	2	3	4	5	6	7	8

s = 90 + 22 + (-70) + 62

- Predikatuaren izena: **batubik**
- 1. ariketako predikatua erabiliz.

27. s balioa x baino handiagoak diren A(1..n) bektoreko elementuen batura da (batuhand)

Adibidea:

x = 20 dela suposatuz, s letra lodiz dauden elementuen batura izango litzateke.

A

90	**22**	1	1	1	-70	**62**	9
1	2	3	4	5	6	7	8

s = 90 + 22 + 62

- Predikatuaren izena: **batuhand**
- Aurretik definitutako predikaturik erabiltzeke egin.

28. A(1..n) bektoreko elementu denak positiboak dira (≥ 1) eta sp A(1..n) bektoreko elementu lehenen biderkadura da (lehenakbider)

Adibidea:

A

17	22	1	1	**2**	**7**	62	9
1	2	3	4	5	6	7	8

sp = 17 * 2 * 7

- Predikatuaren izena: **lehenakbider**
- 7. eta 21. ariketetako predikatuak erabiliz egin.

29. s balioa A(1..n) bektoreko hurrengo posizioan hurrengo zenbakia duten elementuen batura da (batuhur)

Adibidea:

A

20	22	**1**	2	1	**7**	**8**	9
1	2	**3**	4	5	**6**	**7**	8

s = 1 + 7 + 8

- Predikatuaren izena: **batuhur**
- Aurretik definitutako predikaturik erabiltzeke egin.

30. A(1..n) bektoreko pos1 eta pos2 posizioen artean x balioa gutxienez behin agertzen da (agertzenda)

- Predikatuaren izena: **agertzenda**
- Aurretik definitutako predikaturik erabiltzeke egin. Bi aukera:
  (i) pos1 balioa 1 baino handiagoa edo berdina eta pos2 baino txikiagoa edo berdina da, pos2 balioa pos1 baino handiagoa edo berdina eta n baino txikiagoa edo berdina da eta A(1..n) bektorean pos1 baino handiagoa edo berdina eta pos2 baino txikiagoa edo berdina den posizioren batean x agertzen da.
  (ii) pos1 balioa 1 baino handiagoa edo berdina eta pos2 baino txikiagoa edo berdina da, pos2 balioa pos1 baino handiagoa edo berdina eta n baino txikiagoa edo berdina da eta A(1..n) bektorean pos1 baino handiagoak edo berdinak eta pos2 baino txikiagoak edo berdinak diren eta x balioa duten posizioen kopurua 1 edo handiagoa da.

31. x eta y desberdinak dira eta A(1..n) bektorean agertzen dira (biakagertzen)

- Predikatuaren izena: **biakagertzen**
- 30. ariketako predikatua erabiliz.

32. x eta w desberdinak dira eta A(1..n) bektoreko elementu denak x baino handiagoak dira eta w-ren desberdinak (hand_desb)

- Predikatuaren izena: **hand_desb**
- Bi eratara egin:
  (i) 5. eta 30. ariketetako predikatuak erabiliz. (x eta w desberdinak dira eta A(1..n) bektoreko elementu denak x baino handiagoak dira eta w ez da agertzen A(1..n) bektorean)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (x eta w desberdinak dira eta A(1..n) bektoreko posizio denetan x baino handiagoa eta w-ren desberdina den balio bat dago)

33. A(1..n) bektorean x-ren anizkoitzak diren bi elementu eta w-ren anizkoitzak diren hiru elementu daude (bihiruaniz)

- Predikatuaren izena: **bihiruaniz**
- Aurretik definitutako predikaturik erabiltzeke egin. (x zenbakiaz zatitzean hodar bezala 0 ematen duen balioa duten A(1..n) bektoreko posizioen kopurua 2 da eta w zenbakiaz zatitzean hodar bezala 0 ematen duen balioa duten A(1..n) bektoreko posizioen kopurua 3 da)

34. A(1..n) taulako elementu bakoitza positiboa (> 0) da eta bere posizioaren anizkoitza da (posaniz)

Adibidea:

8	4	12	12	10	24	14	8
1	2	3	4	5	6	7	8

- Predikatuaren izena: **posaniz**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko posizio denetan 0 baino handiagoa den eta posizioaz zatitzean hondar bezala 0 ematen duen balioa dago)

35. A(1..n) bektoreko posizio bikoiti denetan zenbaki positibo bat dago (> 0) (posbikpos)

Adibidea:

8	6	-12	2	10	14	-1	8
1	2	3	4	5	6	7	8

- Predikatuaren izena: **posbikpos**
- 1. ariketako predikatua erabiliz egin.

36. A(1..n) bektorean gutxienez elementu bat x baino handiagoa da eta gutxienez elementu bat x baino txikiagoa da (handtxik)

- Predikatuaren izena: **handtxik**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko posizioren bateko balioa x baino handiagoa da eta A(1..n) bektoreko posizioren bateko balioa x baino txikiagoa da)

37. A(1..n) bektoreko pos posiziotik aurrera balio negatibo bakarra dago eta ez dago zerorik (negbat_zeroez)

- Predikatuaren izena: **negbat_zeroez**
- Bi eratara egin:
  (i) 12. ariketako predikatua erabiliz. (pos 1 eta n-ren arteko balio bat da, A(1..n) taulan balio negatiboa duten pos eta n posizioen arteko posizio-kopurua 1 da eta A(1..n) taulan 0 balioa duten pos eta n posizioen arteko posizio-kopurua 0 da)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (pos 1 eta n-ren arteko balio bat da, A(1..n) taulan balio negatiboa duten pos eta n posizioen arteko posizio-kopurua 1 da eta A(1..n) taulan 0 balioa duten pos eta n posizioen arteko posizio-kopurua 0 da)

38. x balioa pos posizioan agertzen da lehenengo aldiz A(1..n) bektorean (lehenager)

- Predikatuaren izena: **lehenager**
- Bi eratara egin:
  (i) 30. ariketako predikatua erabiliz. (pos balioa 1 eta n-ren artean dago, A(1..n) bektoreko pos posizioan x balioa dago eta A(1..n) bektoreko 1 eta pos – 1 posizioen artean x ez da agertzen)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (pos balioa 1 eta n-ren artean dago, A(1..n) bektoreko pos posizioan x balioa dago eta A(1..n) bektoreko 1 eta pos – 1 posizioen arteko posizio denetan x-ren desberdina den balio bat daukagu)

39. *badago* izeneko aldagai boolearrak x balioa A(1..n) bektorean agertzen al den adierazten du (agertzen_al_da)

- Predikatuaren izena: **agertzen_al_da**
- Bi eratara egin:
  (i) 30. ariketako predikatua erabiliz. (*badago* aldagaiaren balioa True da baldin eta soilik baldin x balioa A(1..n) bektorean agertzen bada)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (*badago* aldagaiaren balioa True da baldin eta soilik baldin A(1..n) bektorean x balioa duen posizioren bat existitzen bada)

40. *aurre* izeneko aldagai boolearrak x balioa A(1..n) bektorean pos posizioa baino lehenago agertzen al den adierazten du (lehenago)

- Predikatuaren izena: **lehenago**
- 30. ariketako predikatua erabiliz. (pos balioa 1 eta n-ren artean dago eta aurre izeneko aldagai boolearra True da baldin eta soilik baldin x balioa A(1..n) bektoreko 1 eta pos – 1 posizioen artean agertzen bada)

41. z balioa A(1..n) bektorean jarraian dauden zeroz osatutako bikote-kopurua da (zerobikote)

1. adibidea:

Honako bektore hau hartuta:

8	23	0	0	0	-57	14	0	94	35
1	2	3	4	5	6	7	8	9	10

Bektore honetan jarraian dauden zeroz osatutako bi bikote daudela kontsideratu behar da. Lehenengo bikotea 3 eta 4 posizioetan daudenek osatzen dute eta bigarrena 4 eta 5 posizioetan daudenek.

2. adibidea:

Honako bektore hau hartuta:

74	0	0	0	0	6	-68	0	0	-40
1	2	3	4	5	6	7	8	9	10

Bektore honetan jarraian dauden zeroz osatutako lau bikote daudela kontsideratu behar da. Lehenengo bikotea 2 eta 3 posizioetan daudenek osatzen dute, bigarrena 3 eta 4 posizioetan daudenek, hirugarrena 4 eta 5 posizioetan daudenek eta laugarrena 8 eta 9 posizioetan daudenek.

- Predikatuaren izena: **zerobikote**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektorean zero balio edukita, hurrengo posizioan ere 0 balioa duten 1 eta n – 1 tarteko posizioen kopurua z da)

42. A(1..n) bektorea 2 zenbakiaren berredura osoz osatuta dago (denakbiber)

Adibidea:

Honako bektore hau 2 zenbakiaren berredura osoz osatuta dago:

8	1	2	32	8	16	16	16	128	1
1	2	3	4	5	6	7	8	9	10

Berredura osoak 0 baino handiagoak edo berdinak diren zenbaki osoak berretzaile bezala erabiliz lortzen direnak dira. Jarraian goiko taulako elementu bakoitzaren deskonposaketa ikus daiteke:

2^3	2^0	2^1	2^5	2^3	2^4	2^4	2^4	2^7	2^0
1	2	3	4	5	6	7	8	9	10

- Predikatuaren izena: **denakbiber**
- 4. ariketako predikatua erabiliz. (A(1..n) bektoreko posizio bakoitzean 2ren berredura bat daukagu)

43. x balioa A(1..n) taulako balio txikiena da (txikiena)
- Predikatuaren izena: **txikiena**
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko posizioren batean x balioa dago eta A(1..n) bektoreko elementu bakoitza x baino handiagoa edo berdina da)

44. i eta j aldagaiek hutsa ez den sekzio edo tarte bat mugatzen dute A(1..n) bektorean eta x balioa A(1..n) taulako A(i..j) sekzioko balio handiena da (sekziokohand)

Adibidea:

Demagun A 10 elementuko bektore bat dela eta i eta j aldagaien balioak 4 eta 8 direla hurrenez hurren.

A

7	58	-60	**0**	**33**	**28**	**16**	**7**	549	95
1	2	3	**4**	**5**	**6**	**7**	**8**	9	10

A(i..j) sekzioa

- Predikatuaren izena: **sekziokohand**
- Aurretik definitutako predikaturik erabiltzeke egin. (i balioa 1 eta n-ren arteko balio bat da, j balioa i eta n-ren arteko balio bat da, A(1..n) bektorean i eta j-ren arteko posizioren batean x balioa dago eta i eta j-ren arteko posizio bakoitzean x baino handiagoa edo berdina den balio bat dago)

45. i eta j aldagaiek hutsa ez den sekzio edo tarte bat mugatzen dute A(1..n) bektorean eta A(1..n) bektoreko elementu denak positiboak dira (≥ 1) eta A(1..n) taulako A(i..j) sekzioan x elementu lehen daude (sekziolehenkop)

Adibidea:

Demagun A 10 elementuko bektore bat dela eta i eta j aldagaien balioak 2 eta 6 direla hurrenez hurren. A(i..j) sekzioak bi zenbaki lehen ditu (17 eta 23):

A

7	**58**	**17**	**21**	**23**	**64**	16	7	549	95
1	**2**	**3**	**4**	**5**	**6**	7	8	9	10

A(i..j) sekzioa

- Predikatuaren izena: **sekziolehenkop**
- 7. eta 21. ariketetako predikatuak erabiliz egin. (i balioa 1 eta n-ren arteko balio bat da, j balioa i eta n-ren arteko balio bat da, A(1..n) bektorean elementu denak 1 edo handiagoak dira eta i eta j-ren arteko posizioetan zenbaki lehenen kopurua x da.)

46. pos eta pos + 1 A(1..n) taulan balio bera duten eta jarraian dauden azkeneko bi posizioen indizeak dira (azkenberdinak)

- Predikatuaren izena: **azkenberdinak**
- Aurretik definitutako predikaturik erabiltzeke egin. (pos balioa 1 eta n – 1 balioen artean dago, A(1..n) bektorean pos eta pos + 1 posizioetan balio bera dago pos + 1 eta n – 1 tarteko posizio bakoitzeko balioa eta hurrengo posizioko balioa desberdinak dira)

47. pos balioa A(1..n) bektoreko posizio bat da eta A(1..n) bektorean pos posiziora arte (bera barne) ez dago errepikatutako elementurik (desberdinposizioraino)

- Predikatuaren izena: **desberdinposizioraino**
- 14. ariketako predikatua erabiliz egin. (pos balioa 1 eta n-ren artean dago eta A(1..n) bektorean 1 eta pos posizioen arteko elementu denak desberdinak dira)

48. x balioa A(1..n) bektorean agertzen da eta bere agerpen denak jarraian daude ezkerreko ertzetik hasita (ezkerretikjarraian)

- Predikatuaren izena: ezkerretikjarraian
- Bi eratara egin:
  (i) 12. ariketako predikatua erabiliz. (x balioa A(1..n) bektorean 1 eta 1 baino handiagoa edo berdina den beste posizioren baten arteko posizio denetan agertzen da eta x balioa 0 aldiz agertzen da posizio horretatik n-ra arteko tartean)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (x balioa A(1..n) bektorean 1 eta 1 baino handiagoa edo berdina den beste posizioren baten arteko posizio denetan agertzen da posizio horretatik n-ra arteko posizio denetan x-ren desberdina den balio bat agertzen da)

49. x balioa A(1..n) bektorean agertzen baldin bada, bere agerpen denak jarraian daude ezkerreko ertzetik hasita (ezkerretikjarraian2)

- Predikatuaren izena: ezkerretikjarraian2
- Bi eratara egin:
  (i) 12. ariketako predikatua erabiliz. (x balioa A(1..n) bektorean 1 eta 0 baino handiagoa edo berdina den beste posizioren baten arteko posizio denetan agertzen da eta x balioa 0 aldiz agertzen da posizio horretatik n-ra arteko tartean)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (x balioa A(1..n) bektorean 1 eta 0 baino handiagoa edo berdina den beste posizioren baten arteko posizio denetan agertzen da posizio horretatik n-ra arteko posizio denetan x-ren desberdina den balio bat agertzen da)

50. A(1..n) = (a_n, a_1, …, a_{n-1}) (eskuinbiraketa)

Kasu honetan a_i elementuen azpiindizearearen eta bektorean duten posizioaren arteako erlazioa zein den adierazi behar da.

A(1..n) = (a_n, a_1, …, a_{n-1}) dela esatea A(1..n) bektorea jarraian zehazten den erakoa dela esatearen berdina da:

A

a_n	a_1	a_2	…	a_{n-3}	a_{n-2}	a_{n-1}
1	2	3	…	n – 2	n – 1	n

- Predikatuaren izena: **eskuinbiraketa**
- Aurretik definitutako predikaturik erabiltzeke egin. (1 posizioan a_n balioa dago eta 2 eta n-ren arteko posizio denetan a balioaren azpiindizea posizioa ken 1 da)

51. A(1..n) = (a_2 , …, a_n , a_1) (ezkerbiraketa)

Kasu honetan a_i elementuen azpiindizearearen eta bektorean duten posizioaren arteako erlazioa zein den adierazi behar da.

A(1..n) = (a_2 , …, a_n, a_1) dela esatea A(1..n) bektorea jarraian zehazten den erakoa dela esatearen berdina da:

A

a_2	a_3	a_4	…	a_{n-1}	a_n	a_1
1	2	3	…	n – 2	n – 1	n

- Predikatuaren izena: **ezkerbiraketa**
- Aurretik definitutako predikaturik erabiltzeke egin. (n posizioan a_1 balioa dago eta 1 eta n – 1 posizioen arteko posizio denetan a balioaren azpiindizea posizioa gehi 1 da)

52. A(1..n) bektoreko elementuak ez dira B(1..m) bektorean agertzen (disjuntuak)

- Predikatuaren izena: **disjuntuak**
- Bi eratara egin:
  - (i) 30. ariketako predikatua erabiliz. (A(1..n) bektoreko edozein posiziotako elementua hartuta, elementu hori ez da B(1..m) bektorean agertzen)
  - (ii) Aurretik definitutako predikaturik erabiltzeke. (A(1..n) bektoreko edozein posiziotako elementua hartuta, B(1..m) bektorean ez da existitzen balio bera duen posiziorik)

53. A(1..n) bektorean elementu negatiboak positiboak baino gutxiago dira eta zero ez da agertzen (posgehiago)

- Predikatuaren izena: **posgehiago**
- 30. ariketako predikatua erabiliz. (Balio negatiboa duten A(1..n) bektoreko posizioen kopurua balio positiboa duten posizioen kopurua baino txikiagoa da eta 0 balioa ez da agertzen A(1..n) bektorean)

54. A(1..n) bektorea B(1..n) bektorearen permutazio bat da (permutazioa)
A(1..n) bektore bat beste B(1..n) bektore baten permutazioa izango da bektore bietan elementu berdinak eta kopuru berean agertzen badira nahiz eta orden desberdinean egon (ordenak ez du garrantzirik).

Adibidea:

A

7	80	-60	3	0	0	0	7	50	30
1	2	3	4	5	6	7	8	9	10

B

0	0	30	3	-60	7	80	7	50	0
1	2	3	4	5	6	7	8	9	10

A(1..10) bektorea B(1..10) bektorearen permutazio bat da.

- Predikatuaren izena: permutazioa
- Aurretik definitutako predikaturik erabiltzeke egin. (1 eta n-ren arteko edozein k posizio hartuta, A(k) balioa duten A(1..n) bektoreko posizioen kopurua eta A(k) balioa duten B(1..n) bektoreko posizioen kopurua berdina da)

55. A(1..n) bektorea kapikua edo palindromoa da (palindromoa)

1. adibidea:

A

7	10	-60	3	0	0	3	-60	10	7
1	2	3	4	5	6	7	8	9	10

2. adibidea:

A

'a'	'b'	'c'	'd'	'c'	'b'	'a'
1	2	3	4	5	6	7

- Predikatuaren izena: palindromoa
- Aurretik definitutako predikaturik erabiltzeke egin. (A(1..n) bektoreko edozein posizio hartuta, posizio horretako eta posizio simetrikoko balioak berdinak dira)

A(1..n) bektoreko i posizio batentzat bere posizio simetrikoa n – i + 1 da.

56. A(1..n) bektorean <u>gutxienez</u> jarraian dauden eta desberdinak diren <u>bi</u> elementu daude (gutxienezbidesb)

- Predikatuaren izena: **gutxienezbidesb**
- Aurretik definitutako predikaturik erabiltzeke egin. (1 eta n – 1 posizioen arteko posizioren batentzat posizio horretako eta hurrengo posizioko elementuak desberdinak dira)

57. A(1..n) bektorean jarraian dauden eta desberdinak diren bi elementu bakarrik daude (justubidesb)

- Predikatuaren izena: **justubidesb**
- Aurretik definitutako predikaturik erabiltzeke egin. (Hurrengo posizioan balio desberdina duten 1 eta n – 1 posizioen arteko posizio-kopurua 1)

58. np balioa A(1..n) eta B(1..m) bektoreetan balio bera duten posizioen kopurua da (balioberakop)

Adibidea:

A eta B bektoreek balio bereko 3 posizio dituzte (2, 5 eta 6):

A

26	**38**	-54	7	**1**	**-70**	44	9
1	**2**	3	4	**5**	**6**	7	8

B

0	**38**	1	6	**1**	**-70**	80	310	44	1
1	**2**	3	4	**5**	**6**	7	8	9	10

- Predikatuaren izena: **balioberakop**
- Aurretik definitutako predikaturik erabiltzeke egin. (1 eta n eta 1 eta m-ren artean dauden eta A(1..n) eta B(1..m) bektoreetan balio bera duten posizioen kopurua np da)

59. x balioaren agerpen-kopurua berdina da A(1..n) bektorean eta B(1..p) bektorean (agerpenkopbera)

- Predikatuaren izena: **agerpenkopbera**
- Aurretik definitutako predikaturik erabiltzeke egin. (x balioa duten A(1..n) bektoreko posizioen kopurua eta x balioa duten B(1..p) bektoreko posizioen kopurua berdina da)

60. A(1..n) bektoreak digituak bakarrik ditu eta x zenbaki naturala adierazten du (ezkerrean zeroak egon daitezke) (zenbakia)

Adibidea:

Zenbaki naturalak negatiboak ez diren zenbaki osoak dira (0, 1, 2, …).

Jarraian datozen bi bektore horiek 1280 zenbakiaren bi errepresentazio dira.

A(1..7)

0	0	0	1	2	8	0
1	2	3	4	5	6	7

$1280 = 0 \times 10^6 + 0 \times 10^5 + 0 \times 10^4 + 1 \times 10^3 + 2 \times 10^2 + 8 \times 10^1 + 0 \times 10^0$

A(1..4)

1	2	8	0
1	2	3	4

$1280 = 1 \times 10^3 + 2 \times 10^2 + 8 \times 10^1 + 0 \times 10^0$

- Predikatuaren izena: **zenbakia**
- 8. ariketako predikatua erabiliz. (A(1..n) bektoreak bakarrik digituak ditu eta digitu bakoitzeko digitua bider 10 ber n ken digituaren posizioa batuz x balioaren berdina den batura lortzen da)

61. A(1..n) bektorean digituak bakarrik daude eta kapikua den zenbaki bat osatzen dute digitu horiek. Bektoreak elementu bakarra duenean bakarrik onartuko da bektoreko 1 posizioan 0 balioa agertzea (kasu horretan bektore osoak 0 zenbakia adieraziko luke) (kapikua)

1. adibidea:

Adibide honetako A(1..10) bektoreak digituak bakarrik ditu eta digitu horiek zenbaki kapikua bat osatzen dute.

A

8	3	9	7	1	1	7	9	3	8
1	2	3	4	5	6	7	8	9	10

2. adibidea:

Adibide honetako A(1..7) bektoreak digituak bakarrik ditu eta digitu horiek zenbaki kapikua bat osatzen dute.

A

6	2	0	5	0	2	6
1	2	3	4	5	6	7

3. adibidea:

Adibide honetako A(1..1) bektoreak 0 zenbakia errepresentatzen du (kapikua da).

A

0
1

4. adibidea:

Jarraian datorren bektorea ez da egokia ezkerrean zeroak dituelako:

0	0	0	5	8	2	8	5
1	2	3	4	5	6	7	8

5. adibidea:

Beste bektore hau ere ez da egokia ezkerrean zeroak dituelako:

0	0	0	6	2	6	0	0	0
1	2	3	4	5	6	7	8	9

- Predikatuaren izena: **kapikua**
- 8. eta 55. ariketetako predikatuak erabiliz egin. (A(1..n) bektoreak digituak bakarrik ditu, palindromoa da eta 1 posizioan 0 balioa baldin badago, orduan n-ren balioa 1 da.)

62. B(1..n) bektoreko posizio bakoitzean A(1..n) bektoreko posizio berean dagoen elementua A(1..n) bektorean guztira zenbat aldiz agertzen den adierazten duen balioa dago (kopurua)

Adibidea:

Adibide honetako B(1..10) bektoreko posizio bakoitzean A(1..10) bektoreko posizio bereko elementua A(1..10) bektorean zenbat aldiz agertzen den adierazten da:

A

8	3	8	8	1	1	-70	59	8	20
1	2	3	4	5	6	7	8	9	10

B

4	1	4	4	2	2	1	1	4	1
1	2	3	4	5	6	7	8	9	10

8 zenbakia 4 aldiz agertzen da A(1..10) bektorean, 3 zenbakia behin eta abar.

- Predikatuaren izena: **kopurua**
- Bi eratara egin:
  (i) 13. ariketako predikatua erabiliz. (B(1..n) bektoreko edozein posizio hartuta, posizio horretako balioak posizio bereko A(1..n) bektoreko balioa A(1..n) bektorean zenbat aldiz agertzen den adierazten du)
  (ii) Aurretik definitutako predikaturik erabiltzeke. (B(1..n) bektoreko edozein posizio hartuta, posizio horretako balioak posizio bereko A(1..n) bektoreko balioa A(1..n) bektoreko zenbat posiziotan agertzen den adierazten du)

63. (sekzioazpibek)

A(1..n) bektoreak gutxienez elementu bat du. B(1..p) bektoreko elementu-kopurua A(1..n) bektorekoa baino handiagoa edo berdina da. Gainera i eta j balioek n elementuko azpitarte bat mugatzen dute [1..p] tartearen barruan. A(1..n) bektorea B(1..p) bektorearen azpibektore bat da eta A-ko elementu denak B-n jarraian eta orden berean agertzen dira i posiziotik hasita eta j posizioraino.

Adibidea:

Adibide honetako A(1..4) bektorea B(1..10) bektorearen azpibektore bat da eta 4 posiziotik 7 posizioraino doa:

A

40	333	6	10
1	2	3	4

B

11	78	0	**40**	**333**	**6**	**10**	-68	4	1
1	2	3	**4**	**5**	**6**	**7**	8	9	10
			i			j			

$\underbrace{\qquad\qquad\qquad\qquad}_{A}$

- Predikatuaren izena: **sekzioazpibek**
- Aurretik definitutako predikaturik erabiltzeke egin. (n balioa 1 baino handiagoa edo berdina da, p balioa n baino handiagoa edo berdina da, i balioa 1 baino handiagoa edo berdina eta n baino txikiagoa edo berdina da, j balioa i baino handiagoa edo berdina eta n baino txikiagoa edo berdina da, j ken i-ren balioa n – 1 da, A(1..n) bektoreko edozein k posizio hartuta, B(i + k – 1) eta A(k) berdinak dira.

64. (azpibek)

A(1..n) bektoreak gutxienez elementu bat du. B(1..p) bektoreko elementu-kopurua A(1..n) bektorekoa baino handiagoa edo berdina da. A(1..n) bektorea B(1..p) bektorearen azpibektore bat da, hau da, A-ko elementu denak B-n jarraian eta orden berean agertzen dira

Adibidea:

Adibide honetako A(1..4) bektorea B(1..10) bektorearen azpibektorea da:

A

40	333	6	10
1	2	3	4

B

11	78	0	**40**	**333**	**6**	**10**	-68	4	1
1	2	3	**4**	**5**	**6**	**7**	8	9	10

A

- Predikatuaren izena: **azpibek**
- Bi eratara egin:
  - (i) 63. ariketako predikatua erabiliz. (1 baino handiagoa edo berdina eta p baino txikiagoa edo berdina den h posizioren batentzat eta 1 baino handiagoa edo berdina eta p baino txikiagoa edo berdina den g posizioren batentzat A(1..n) bektorea justu h eta g balioek B(1..p) bektorean definitzen duten sekzioan agertzen da azpibektore bezala)
  - (ii) Aurretik definitutako predikaturik erabiltzeke. (n balioa 1 baino handiagoa edo berdina da, p balioa n baino handiagoa edo berdina da, 1 baino handiagoa edo berdina eta n baino txikiagoa edo berdina den h posizioren batentzat g – h = n – 1 betetzen duen eta h baino handiagoa edo berdina eta n baino txikiagoa edo berdina den g posizioren bat existitzen da eta A(1..n) bektoreko edozein k posizio hartuta, B(h + k – 1) eta A(k) berdinak dira.

65. (gutxienezbikotebat)

mp aldagai boolearrak A(1..n) eta B(1..m) bektoreek gutxienez jarraian dauden elementuz osatutako bikote berdinen bat ba al duten adierazten du (bektore bietan posizio berdinetan egonez).

Adibidea:

A

44	**38**	**1**	1	1	**-70**	**62**	9
1	**2**	**3**	4	5	**6**	**7**	8

B

44	**38**	**1**	6	22	**-70**	**62**	1	-70	1
1	**2**	**3**	4	5	**6**	**7**	8	9	10

Adibide honetako A eta B bektoreek jarraian dauden elementuz osatutako hiru bikote berdin dituzte posizio berdinetan:

- 1 eta 2 posizioak.
- 2 eta 3 posizioak.
- 6 eta 7 posizioak.

➢ Predikatuaren izena: **gutxienezbikotebat**
➢ Aurretik definitutako predikaturik erabiltzeke egin.

66. (justubikotebat)

b aldagai boolearrak A(1..n) eta B(1..m) bektoreek justu jarraian dauden elementuz osatutako bikote berdin bakar bat al duten adierazten du (bektore bietan posizio berdinetan egonez).

Adibidea:

Adibide honetako A eta B bektoreek jarraian dauden elementuz osatutako bikote berdin bakar bat dute posizio berdinetan: 6 eta 7 posizioak.

A

90	22	1	1	1	**-70**	**62**	9
1	2	3	4	5	**6**	**7**	8

B

44	38	1	6	22	**-70**	**62**	1	-70	1
1	2	3	4	5	**6**	**7**	8	9	10

➢ Predikatuaren izena: **justubikotebat**
➢ Aurretik definitutako predikaturik erabiltzeke egin.

67. (gutxienezbikotebat2)

dp aldagai boolearrak A(1..n) eta B(1..m) bektoreek <u>gutxienez</u> jarraian dauden elementuz osatutako bikote berdinen <u>bat</u> ba al duten adierazten du (bektore bietan posizio desberdinetan egon daitezkeelarik).

Adibidea:

A

44	**38**	1	1	**6**	**-70**	**62**	9
1	**2**	3	4	**5**	**6**	**7**	8

B

20	**44**	**38**	**6**	**-70**	**62**	5	1	-70	1
1	**2**	**3**	**4**	**5**	**6**	7	8	9	10

Adibide honetako A eta B bektoreek jarraian dauden elementuz osatutako hiru bikote berdin dituzte nahiz eta posizio desberdinetan egon:

- A(1), A(2) eta B(2), B(3) posizioak.
- A(5), A(6) eta B(4), B(5) posizioak.
- A(6), A(7) eta B(5), B(6) posizioak.

➢ Predikatuaren izena: **gutxienezbikotebat2**
➢ Aurretik definitutako predikaturik erabiltzeke egin.

68. (justubikotebat2)

c aldagai boolearrak A(1..n) eta B(1..m) bektoreek <u>justu</u> jarraian dauden elementuz osatutako bikote berdin <u>bakar bat</u> al duten adierazten du (bektore bietan posizio desberdinetan egon daitezkeelarik)

Adibidea:

A

90	22	1	1	1	**-70**	**62**	9
1	2	3	4	5	**6**	**7**	8

B

44	38	**-70**	**62**	22	18	5	2	-70	1
1	2	**3**	**4**	5	6	7	8	9	10

Adibide honetako A eta B bektoreek jarraian dauden elementuz osatutako bikote berdin bakar bat dute eta gainera posizio desberdinetan: A(6), A(7) eta B(3), B(4) posizioak.

➢ Predikatuaren izena: **justubikotebat2**
➢ Aurretik definitutako predikaturik erabiltzeke egin.

69. u eta v zenbaki lehenak dira eta beraien artean ez dago zenbaki lehenik (lehenakjarraian)

- Predikatuaren izena: l**ehenakjarraian**
- 21. ariketako predikatua erabiliz egin (u balioa v baino txikiagoa da, u lehena, v lehena da eta u eta v-ren artean ez dago lehena den zenbakirik).

70. i balioa C(1..m) bektoreko elementu handienaren indizea da (indizehand)

- Predikatuaren izena: **indizehand**
- 44. ariketako predikatua erabiliz egin.

71. i balioa C(1..m) bektoreko elementu txikienaren indizea da (indizetxik)

- Predikatuaren izena: **indizetxik**
- 43. ariketako predikatua erabiliz egin.

72. B(1..m) bektorean C(1..m) bektoreko elementuak daude baina goranzko ordenean (gorantz)

- Predikatuaren izena: **gorantz**
- 54. ariketako predikatua erabiliz egin.

73. C(1..m) bektorea B(1..m) bektorearen alderantzizkoa da. (alder)

- Predikatuaren izena: **alder**
- Aurretik definitutako predikaturik erabiltzeke egin.

74. C(1..m) bektorean ez dago errepikatutako elementurik (ezerrepikatuta)

- Predikatuaren izena: **ezerrepikatuta**
- 14. ariketako predikatua erabiliz egin.

75. C(1..m) betorean ez dago batura bezala zero ematen duen sekziorik. (ezbaturazero)

- Predikatuaren izena: **ezbaturazero**
- 23. ariketako predikatua erabiliz egin.

76. C(1..m) bektorean ez dago jarraian dauden zeroz osatutako bikoterik (ezzerobik)

- Predikatuaren izena: **ezzerobik**
- 41. ariketako predikatua erabiliz egin.

77. C(1..m) bektoreko elementu bakoitza justu bi aldiz agertzen da bektore horretan (denakbialdiz)

- Predikatuaren izena: **denakbialdiz**
- 20. ariketako predikatua erabiliz egin.

78. C(1..n), D(1..m) eta E(1..p) bektoreek ez dute elementu berdinik beraien artean (hirudisjuntu)

- Predikatuaren izena: **hirudisjuntu**
- Bi eratara egin:
  - (i) 13. ariketako predikatua erabiliz.
  - (ii) 52. ariketako predikatua erabiliz.

79. (positibohautaketa)

C(1..m) bektoreak gutxienez elementu positibo bat du (> 0), B(1..p) bektorean elementu denak positiboak (> 0) eta desberdinak dira eta B-ko elementu denak C-n ere agertzen dira.

- Predikatuaren izena: **positibohautaketa**
- 6, 7, 14 eta 30. ariketetako predikatuak erabiliz egin.

80. (partiketa)

C(1..m) eta D(1..p) bektoreek gutxienez elementu bana dute eta E(1..q) bektorearen partiketa bat osatzen dute, hau da, C eta D elkartuz E lortzen da.

Adibidea:

C	90	22	1
	1	2	3

D	44	38	1	6	22
	1	2	3	4	5

E	90	22	1	44	38	1	6	22
	1	2	3	4	5	6	7	8

- Predikatuaren izena: **partiketa**
- 63. ariketako predikatua erabiliz egin.

81. (lehenakgorantz)

A(1..n) bektorean lehenengo n zenbaki lehenak daude goranzko ordenean.

- Predikatuaren izena: **lehenakgorantz**
- 69. ariketako predikatua erabiliz. (1 posizioan 2 balioa dago (lehenengo zenbaki lehena) eta 1 eta n – 1 posizioen arteko edozein posizio hartuta, posizio horretak elementua eta hurrengo posizioko elementua elkarren ondorengo zenbaki lehenak dira.

82. (handtxikbehin)

max balioa B(1..m) bektoreko balio handiena da, *min* balioa txikiena da bai handiena eta bai txikiena behin bakarrik agertzen dira bektorean.

- Predikatuaren izena: **handtxikbehin**
- 13, 43 eta 44. ariketetako predikatuak erabiliz egin.

b) Programen aurre-ondoetako espezifikazioa

1. A(1..n) eta B(1..n) bektoreetan balio bera duten posizio-kopurua c aldagaian zenbatu

A(1..n) eta B(1..n) bektoreak emanda ($n \geq 1$), c aldagaian zenbat posiziotan koinziditzen duten kontatzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 1;
c : = 0;
while i ≤ n loop
        if A(i) = B(i) then c : = c + 1; end if;
        i : = i + 1;
end loop;
{ψ} ≡ ???
```

2. A(1..n) eta B(1..n) bektoreek balio bera duen posiziorik ba al duten erabaki w aldagaian

A(1..n) eta B(1..n) bektoreak emanda ($n \geq 1$), w aldagai boolearrean bektore biek posizioren batean koinziditzen al duten erabakitzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 1;
w : = false;
while i ≠ n + 1 and not w loop
        w : = (A(i) = B(i));
        i : = i + 1;
end loop;
{ψ} ≡ ???
```

3. A(1..n) bektoreko elementu denak berdinak al diren erabaki berdinak aldagaian

A(1..n) bektorea emanda ($n \geq 1$), berdinak aldagai boolearrean bektoreko elementu denak berdinak al diren erabakitzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 1;
berdinak : = true;
while i ≠ n and berdinak loop
        berdinak : = (A(i) = A(i + 1));
        i : = i + 1;
end loop;
{ψ} ≡ ???
```

4. A(1..n) eta B(1..n) bektoreetan balio bera duten posizioen kopurua balio desberdina dutenen kopurua baino handiagoa al den erabaki d aldagai boolearrean

A(1..n) eta B(1..n) bektoreak emanda (n ≥ 1), A(1..n) eta B(1..n) bektoreetan balio bera duten posizioen kopurua balio desberdina dutenen kopurua baino handiagoa al den d aldagai boolearrean erabakitzen duen programaren aurre-ondoetako espezifikazioa eman::

```
{φ} ≡ ???
i : = 1;
c : = 0;
while i ≠ n + 1 loop
        if A(i) = B(i) then c : = c + 1; end if;
        i : = i + 1;
end loop;
d : = (c > (n div 2));
{ψ} ≡ ???
```

div zatiketa osoa da.

5. A(1..n) bektorean batekoen kopurua zero kopurua baino handiagoa al den erabaki b aldagai boolearrean

Batekoak eta zeroak bakarrik dituen A(1..n) bektorea emanda (n ≥ 1), b aldagai boolearrean bateko kopurua zero kopurua baino handiagoa al den erabakitzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 0;
c : = 0;
while i ≠ n loop
        if A(i + 1) = 1 then c : = c + 1; end if;
        i : = i + 1;
end loop;
b : = (c > (n div 2));
{ψ} ≡ ???
```

div zatiketa osoa da.

6. A(1..n) bektorean batekoen kopurua zero-kopurua baino handiagoa baldin bada, *geh* aldagaian batekoen kopurua itzuli eta bestela zero kopurua

Batekoak eta zeroak bakarrik dituen A(1..n) bektorea emanda ($n \geq 1$), bateko kopurua zero kopurua baino handiagoa bada, *geh* aldagaiean bateko kopurua eta bestela zero kopurua itzultzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 0;
c : = 0;
u : = 0
while i ≠ n loop
          if A(i + 1) = 0 then c : = c + 1;
          else u : = u + 1; end if;
          i : = i + 1;
end loop;
if u > c then geh : = u;
           else geh : = c; end if;
{ψ} ≡ ???
```

7. A(1..n) bektorean 1 dagoen leku bakoitzean B(1..n) bektorean 0 balioa eta A(1..n) bektorean 0 dagoen leku bakoitzean B(1..n) bektorean 1 balioa gorde

Batekoak eta zeroak bakarrik dituen A(1..n) bektorea emanda ($n \geq 1$), A-n batekoak dauden lekuetan zeroak ipiniz eta A-n zeroak dauden lekuetan batekoak ipiniz B(1..n) bektorea eratzen duen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 1;
while i ≠ n + 1 loop
          if A(i) = 0 then B(i) : = 1;
          else B(i) : = 0; end if;
          i : = i + 1;
end loop;
{ψ} ≡ ???
```

8. Batekoen ordez zeroak eta zeroen ordez batekoak ipini A(1..n) bektorean

Batekoak eta zeroak bakarrik dituen A(1..n) bektorea emanda ($n \geq 1$), A-ko batekoak eta zeroak zeroz eta batekoz trukatzen dituen programari dagokion aurre-ondoetako espezifikazioa eman:

```
{φ} ≡ ???
i : = 1;
while i ≠ n + 1 loop
          if A(i) = 0 then A(i) : = 1;
          else A(i) : = 0; end if;
          i : = i + 1;
end loop;
{ψ} ≡ ???
```

c) Programen dokumentazioa

1. C(1..n) bektorean A(1..n) eta B(1..n) bektoreen batura gordetzen duen programa

C(1..n) bektorean A(1..n) eta B(1..n) bektoreen batura gordetzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
      (4) {Tarteko asertzioa}
      C(i) : =A(i) + B(i);
      (5) {Tarteko asertzioa}
      i : = i + 1;
end loop;
(6) {Bukaerako baldintza}
(7) {E espresioa}
```

2. A(1..n) eta B(1..n) bektoreetako elementuak trukatzen dituen programa

A(1..n) eta B(1..n) bektoreetako elementuak trukatzen dituen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
      (4) {Tarteko asertzioa}
      lag : =A(i);
      (5) {Tarteko asertzioa}
      A(i) : =B(i);
      (6) {Tarteko asertzioa}
      B(i) : =lag;
      (7) {Tarteko asertzioa}
      i : = i + 1;
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```

3. v aldagaian A(1..n) bektorean x balioa zenbat aldiz agertzen den zenbatzen duen programa

v aldagaian A(1..n) bektorean x balioa zenbat aldiz agertzen den zenbatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
v : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≤ n loop
    (5) {Tarteko asertzioa}
    if A(i) = x then (6) {Tarteko asertzioa}
                v : = v + 1;
                (7) {Tarteko asertzioa}
    end if;
    (8) {Tarteko asertzioa}
    i : = i + 1;
    (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

4. v aldagaian A(1..n) bektorean x balioa zenbat aldiz agertzen den zenbatzen duen programa

v aldagaian A(1..n) bektorean x balioa zenbat aldiz agertzen den zenbatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
v : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i < n loop
    (5) {Tarteko asertzioa}
    i : = i + 1;
    (6) {Tarteko asertzioa}
    if A(i) = x then (7) {Tarteko asertzioa}
                v : = v + 1;
                (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

5. "berdinak" izeneko aldagai boolearrean A(1..n) eta B(1..n) bektoreak berdinak al diren ala ez erabakitzen duen programa

"berdinak" izeneko aldagai boolearrean A(1..n) eta B(1..n) bektoreak berdinak al diren ala ez erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
berdinak : = True;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≤ n and berdinak loop
      (5) {Tarteko asertzioa}
      if A(i) ≠ B(i) then (6) {Tarteko asertzioa}
                  berdinak : = False;
                  (7) {Tarteko asertzioa}
      end if;
      (8) {Tarteko asertzioa}
      i : = i + 1;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

6. "badago" aldagaian x balioa A(1..n) bektorean agertzen al den erabakitzen duen programa

"badago" aldagaian x balioa A(1..n) bektorean agertzen al den erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
badago : = False;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≠ n and not badago loop
      (5) {Tarteko asertzioa}
      i : = i + 1;
      (6) {Tarteko asertzioa}
      if A(i) = x then
                  (7) {Tarteko asertzioa}
                  badago : = True;
                  (8) {Tarteko asertzioa}
      end if;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

7. z aldagaian x eta y zenbaki oso eta ez negatiboen arteko biderkadura kalkulatzen duen programa

z aldagaian x eta y zenbaki oso eta ez negatiboen arteko biderkadura kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
z : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} x ≠ 0 loop
      (4) {Tarteko asertzioa}
      z : = z + y;
      (5) {Tarteko asertzioa}
      x : = x – 1;
      (6) {Tarteko asertzioa}
end loop;
(7) {Bukaerako baldintza}
(8) {E espresioa}
```

8. ≥ 2 den x zenbakia emanda, "lehena" izeneko aldagai boolearrean x lehena al den ala ez erabakitzen duen programa

≥ 2 den x zenbakia emanda, "lehena" izeneko aldagai boolearrean x lehena al den ala ez erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
d : = 2;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} x mod d ≠ 0 loop
      (4) {Tarteko asertzioa}
      d : = d + 1;
      (5) {Tarteko asertzioa}
end loop;
(6) {Tarteko asertzioa}
lehena : = (d = x);
(7) {Bukaerako baldintza}
(8) {E espresioa}
```

9. A(1..n) bektorea emanda, zenb aldagaian A-n bosten anizkoitzak diren zenbat elementu dauden kalkulatzen duen programa

A(1..n) bektorea emanda, zenb aldagaian A-n bosten anizkoitzak diren zenbat elementu dauden kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
zenb : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i < n loop
      (5) {Tarteko asertzioa}
      i : = i + 1;
     (6) {Tarteko asertzioa
      if A(i) mod 5 = 0 then (7) {Tarteko asertzioa}
                      zenb : = zenb + 1;
                      (8) {Tarteko asertzioa}
      end if;
     (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

10. A(1..n) bektoreko balio txikiena m aldagaian gordetzen duen programa

A(1..n) bektoreko balio txikiena m aldagaian gordetzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
m : = A(1);
(2) {Tarteko asertzioa}
k : = 1;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} k < n loop
    (5) {Tarteko asertzioa}
    k : = k + 1;
    (6) {Tarteko asertzioa}
    if A(k) < m then (7) {Tarteko asertzioa}
               m : = A(k);
                (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

11. ≥ 0 den n zenbakia emanda, x aldagaian Fibonacci-ren segidako s_n elementua kalkulatzen duen programa

≥ 0 den n zenbakia emanda, x aldagaian Fibonacci-ren segidako s_n elementua kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez ($s_0 = 0$, $s_1 = 1$ eta $s_k = s_{k-1} + s_{k-2}$, $k \geq 2$ denerako):

```
(1) {Hasierako baldintza}
x : = 0;
(2) {Tarteko asertzioa}
y : = 1;
(3) {Tarteko asertzioa}
k : = 1;
(4) {Tarteko asertzioa}
while (5) {Inbariantea} k ≤ n loop
      (6) {Tarteko asertzioa}
      y : = y + x;
      (7) {Tarteko asertzioa}
      x : = y – x;
      (8) {Tarteko asertzioa}
      k : = k + 1;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

12. Negatiboa ez den x zenbaki bat emanda, bere faktoriala f aldagaian kalkulatzen duen programa

Negatiboa ez den x zenbaki bat emanda, bere faktoriala f aldagaian kalkulatzen duten programa hauek dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
f : = 1;
(2) {Tarteko asertzioa}
t : = 1;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} t ≤ x loop
      (5) {Tarteko asertzioa}
      f : = f ∗ t;
      (6) {Tarteko asertzioa}
      t : = t + 1;
      (7) {Tarteko asertzioa}
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```

```
(1) {Hasierako baldintza}
f : = 1;
(2) {Tarteko asertzioa}
t : = x;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} t ≥ 1 loop
      (5) {Tarteko asertzioa}
      f : = f ∗ t;
      (6) {Tarteko asertzioa}
      t : = t – 1;
      (7) {Tarteko asertzioa}
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```

13. neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa

neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
neg : = 0;
(2) {Tarteko asertzioa}
i : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≠ n loop
      if A(i + 1) < 0 then
                  (5) {Tarteko asertzioa}
                  neg : = neg + 1;
                  (6) {Tarteko asertzioa}
      end if;
      (7) {Tarteko asertzioa}
      i : = i + 1;
      (8) {Tarteko asertzioa}
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

14. pos aldagaian A(1..n) bektoreak zenbat elementu positibo dituen eta neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa

pos aldagaian A(1..n) bektoreak zenbat elementu positibo dituen eta neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
pos : = 0; neg : = 0; i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
      if A(i + 1) > 0 then
                  (4) {Tarteko asertzioa}
                   pos : = pos + 1;
                  (5) {Tarteko asertzioa}
      elsif A(i + 1) < 0 then
                  (6) {Tarteko asertzioa}
                   neg : = neg + 1;
                  (7) {Tarteko asertzioa}
      end if;
      (8) {Tarteko asertzioa}
      i : = i + 1;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

15. neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa

neg aldagaian A(1..n) bektoreak zenbat elementu negatibo dituen kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez

```
(1) {Hasierako baldintza}
neg : = 0;
(2) {Tarteko asertzioa}
i : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≠ n loop
      (5) {Tarteko asertzioa}
      i : = i + 1;
      (6) {Tarteko asertzioa}
      if A(i) < 0 then (7) {Tarteko asertzioa}
                  neg : = neg + 1;
                  (8) {Tarteko asertzioa}
      end if;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

16. "alder" aldagai boolearrean A(1..n) bektorea B(1..n) bektorearen alderantzizkoa al den erabakitzen duen programa

"alder" aldagai boolearrean A(1..n) bektorea B(1..n) bektorearen alderantzizkoa al den erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez:

```
(1) {Hasierako baldintza}
alder : = True;
(2) {Tarteko asertzioa}
i : = 1;
(3) {Tarteko asertzioa}
j : = n;
(4) {Tarteko asertzioa}
while (5) {Inbariantea} i ≠ n + 1 and alder loop
      (6) {Tarteko asertzioa}
      if A(i) ≠ B(j) then (7) {Tarteko asertzioa}
                  alder : = False;
                  (8) {Tarteko asertzioa}
      end if;
      (9) {Tarteko asertzioa}
      i : = i + 1;
      (10) {Tarteko asertzioa}
      j : = j – 1;
      (11) {Tarteko asertzioa}
end loop;
(12) {Bukaerako baldintza}
(13) {E espresioa}
```

17. A(1..n) bektorea emanda, hurrengo posizioan balio txikiagoa duten elementuen kopurua dekr aldagaian kalkulatzen duen programa

A(1..n) bektorea emanda, hurrengo posizioan balio txikiagoa duten elementuen kopurua *dekr* aldagaian kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez.

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
dekr : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≠ n – 1 loop
      (5) {Tarteko asertzioa}
      i : = i + 1;
      (6) {Tarteko asertzioa}
      if A(i) > A(i + 1) then (7) {Tarteko asertzioa}
                           dekr : = dekr + 1;
                           (8) {Tarteko asertzioa}
      end if;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

18. A(1..n) bektorea emanda, p aldagai boolearrean A(1..n) bektoreko elementu bikoitiak bakoitiak baino gehiago al diren erabakitzen duen programa

A(1..n) bektorea emanda, p aldagai boolearrean A(1..n) bektoreko elementu bikoitiak bakoitiak baino gehiago al diren erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez.

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
bik : = 0;
(3) {Tarteko asertzioa}
while (4) {Inbariantea} i ≠ n loop
      if A(i + 1) mod 2 = 0 then (5) {Tarteko asertzioa}
                                bik : = bik + 1;
                                (6) {Tarteko asertzioa}
      end if;
      (7) {Tarteko asertzioa}
      i : = i + 1;
      (8) {Tarteko asertzioa}
end loop;
(9) {Tarteko asertzioa}
p : = (bik > (n / 2));
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

19. Alderantziz(C(1..h), (c_1, c_2, …, c_h), p) predikatua eta A(1..n) bektorea alderantziz ipintzen duen programa

a) C bektoreak (c_1, c_2, …, c_h) bektoreak dituen elementu berdinak dituela baina p posiziora artekoak alderantziz ipinita daudela adierazten duen **Alderantziz(C(1..h), (c_1, c_2, …, c_h), p)** izeneko predikatua definitu:

C(1..h)

c_h	c_{h-1}	…	c_{h-p+1}	c_{p+1}	…	c_{h-p}	c_p	…	c_2	c_1
1	2	…	p	p + 1	…	h – p	h – p + 1	…	h – 1	h
Alderantziz ipinita				Alderantziz ipintzeke			Alderantziz ipinita			

(c_1, c_2, …, c_h) bektoreak honako bektore hau adierazten du:

c_1	c_2		…	c_{h-1}	c_h
1	2	3	…	h – 1	h

b) Gutxienez elementu bat duen A(1..n) bektorea alderantziz ipintzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataleko predikatua erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n / 2 loop
      lag : = A(i);
      (4) {Tarteko asertzioa}
      A(i) : = A(n – i + 1);
      (5) {Tarteko asertzioa}
      A(n – i + 1) : = lag;
      (6) {Tarteko asertzioa}
      i : = i + 1;
      (7) {Tarteko asertzioa}
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```

20. Handiagoak(C(1..p), D(1..p), x) predikatua eta b aldagai boolearrean A(1..n) bektoreko elementuen erdia baino gehiago B(1..n) bektoreko posizio bereko elementuak baino handiagoak al diren erabakitzen duen programa

a) C-ko elementua D-ko elementua baino handiagoa izatea x posizio desberdinetan gertatzen dela adierazten duen **Handiagoak(C(1..p), D(1..p), x)** izeneko predikatua definitu.

b) b aldagai boolearrean A(1..n) bektoreko elementuen erdia baino gehiago B(1..n) bektoreko posizio bereko elementuak baino handiagoak al diren erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataleko predikatua erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
z : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
      if A(i) > B(i) then
                    (4) {Tarteko asertzioa}
                     z : = z + 1;
                    (5) {Tarteko asertzioa}
      end if;
      (6) {Tarteko asertzioa}
      i : = i + 1;
      (7) {Tarteko asertzioa}
end loop;
(8) {Tarteko asertzioa}
b : = z > (n / 2);
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

21. Gorantz(C(1..m)) eta Agertzen_da(z, C(1..m)) predikatuak eta B(1..n) taulako B(1..j) azpitaulan A(1..n) bektorean x agertzen den posizioetako indizeak gordetzen dituen programa

a) C(1..m) bektoreko elementu bakoitza hurrengo posizioko elementua baino txikiagoa dela adierazten duen **Gorantz**(C(1..m)) izeneko predikatua definitu.

b) z elementua C(1..m) bektorean agertzen dela adierazten duen **Agertzen_da(z, C(1..m))** izeneko predikatua definitu.

c) B(1..n) taulako B(1..j) azpitaulan A(1..n) bektorean x agertzen den posizioetako indizeak gordetzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
j : = 0;
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
      (4) {Tarteko asertzioa}
       if A(i) = x then
                  (5) {Tarteko asertzioa}
                   j : = j + 1;
                   (6) {Tarteko asertzioa}
                   B(j) : = i;
                   (7) {Tarteko asertzioa}
      end if;
      (8) {Tarteko asertzioa}
       i : = i + 1;
      (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}

(11) {E espresioa}
```

22. Tartebatu(C(1..p), h, i, s) eta Baturabera(C(1..p), i) predikatuak eta n ≥ 2 betetzen duen A(1..n) bektorea emanda, batb aldagai boolearean A bektorea batura bereko bi zatitan bana al daitekeen ala ez erabakitzen duen programa

a) C taulan h eta i posizioen arteko elementuen batura s dela adierazten duen **Tartebatu(C(1..p), h, i, s)** izeneko predikatua definitu.

b) C taulan 1 eta i posizioen arteko elementuen batura i + 1 eta n posizioen arteko elementuen baturaren berdina dela adierazten duen **Baturabera(C(1..p), i)** izeneko predikatua definitu.

c) n ≥ 2 betetzen duen A(1..n) bektorea emanda, batb aldagai boolearean A bektorea batura bereko bi zatitan bana al daitekeen ala ez erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
x : = A(1); y : = 0; k : = 2;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} k ≤ n loop
      y : = y + A(k);
      (4) {Tarteko asertzioa}
      k : = k + 1;
      (5) {Tarteko asertzioa}
end loop;
(6) {Tarteko asertzioa}
j : = 1;
(7) {Tarteko asertzioa}
while (8) {Inbariantea} x ≠ y and j < n – 1 loop
    (9) {Tarteko asertzioa}
     j : = j + 1;
    (10) {Tarteko asertzioa}
      x : = x + A(j);
    (11) {Tarteko asertzioa}
      y : = y – A(j);
    (12) {Tarteko asertzioa}
end loop;
(13) {Tarteko asertzioa}
batb : = (x = y);
(14) {Bukaerako baldintza}
```

While bakoitzari dagokion E espresioa ere eman.

23. Berdinak(D(1..r), x), Batura_da(D(1..r), s) eta Bider_da(D(1..r), p) predikatuak eta A(1..n) bektorea emanda, c aldagaian batura eta biderkadura berdina ematen duen 1..n tarteko azken posizioa kalkulatzen duen programa

a) D bektorean 1 eta x posizioen arteko elementuen batura eta biderkadura berdinak direla adierazten duen **Berdinak(D(1..r), x)** izeneko predikatua definitu.

b) D bektorean 1 eta r posizioen arteko elementuen batura s dela adierazten duen **Batura_da(D(1..r), s)** izeneko predikatua definitu.

c) D bektorean 1 eta r posizioen arteko elementuen biderkadura p dela adierazten duen **Bider_da(D(1..r), p)** izeneko predikatua definitu.

d) A(1..n) bektorea emanda, c aldagaian batura eta biderkadura berdina ematen duen 1..n tarteko azken posizioa kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
c : = 1;
batu : = A(1);
bider : = A(1);
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i < n loop
        i : = i + 1;
       (4) {Tarteko asertzioa}
        batu : = batu + A(i);
        (5) {Tarteko asertzioa}
        bider : = bider ∗ A(i);
        (6) {Tarteko asertzioa}
        if batu = bider then (7) {Tarteko asertzioa}
                                c : = i;
                                (8) {Tarteko asertzioa}
        end if;
        (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

24. Gora(T(a..b)) eta Behera(T(a..b)) predikatuak eta b aldagai boolearrean A(1..n) bektoreko posizio batera arte elementuak goranzko ordenean eta posizio horretatik aurrera beheranzko ordenean al dauden erabakitzen duen programa

a) T(a..b) sekzioan elementu bakoitza hurrengoa baino txikiagoa dela adierazten duen **Gora**(T(a..b)) predikatua eta T(a..b) sekzioan elementu bakoitza hurrengoa baino handiagoa dela adierazten duen **Behera**(T(a..b)) predikatua definitu.

b) b aldagai boolearrean A(1..n) bektoreko posizio batera arte elementuak goranzko ordenean eta posizio horretatik aurrera beheranzko ordenean al dauden erabakitzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataleko predikatuak erabiliz. Beraz b aldagaiak bukaeran true balioa izango du A taula bi zatitan bana badaiteke: lehenengoa goranzkorra eta bigarrena beheranzkorra eta bien artean taula osoa hartuz. Zati bakoitzaren elementu-kopurua 1 edo handiagoa izan daiteke. Goranzko ordenean dagoen zatiaren azkeneko elementua beheranzko ordenean dagoen zatiaren lehenengo elementua izango da.

```
(1) {Hasierako baldintza}
i : = 1; j : = n; jarraitu : = True;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ j and jarraitu loop
        (4) {Tarteko asertzioa}
        if A(j – 1) > A(j) then (5) {Tarteko asertzioa}
                                j : = j – 1;
                                 (6) {Tarteko asertzioa}
        elsif A(j – 1) < A(j) (7) {Tarteko asertzioa}
                                i : = i + 1;
                              (8) {Tarteko asertzioa}
        else (9) {Tarteko asertzioa}
              jarraitu : = False;
               (10) {Tarteko asertzioa}
        end if;
        (11) {Tarteko asertzioa}
end loop;
(12) {Tarteko asertzioa}
b : = (i = j);
(13) {Bukaerako baldintza}
(14) {E espresioa}
```

25. Gora(T(a..b)), Behera(T(a..b)) eta Perm(T1(a..b), T2(c..d), T3(e..f)) predikatuak eta goranzko ordenean dagoen A(1..m) bektorea eta beheranzko ordenean dagoen B(1..n) bektorea emanda, C(1..m + n) bektorean A(1..m) eta B(1..n) nahastuta baina goranzko ordenean ipintzen dituen programa

a) T(a..b) sekzioan elementu bakoitza hurrengoa baino txikiagoa dela adierazten duen **Gora**(T(a..b)) izeneko predikatua, T(a..b) sekzioan elementu bakoitza hurrengoa baino handiagoa dela adierazten duen **Behera**(T(a..b)) izeneko beste predikatua eta T3 bektoreko T3(e..f) sekzioa T1 bektoreko eta T2 bektoreko T1(a..b) eta T2(c..d) sekzioen permutazioa dela adierazten duen **Perm**(T1(a..b), T2(c..d), T3(e..f)) predikatua definitu (T3(e..f) beste bien permutazioa izateak T3(e..f) sekzioan T1(a..b) eta T2(c..d) sekzioetako elementu denak eta kopuru berean agertzen direla esan nahi du).

Adibidea: T1(1..3) = (14, 9, 6), T2(1..7) = (2, 5, 5, 1, 14, 7, 20) eta
T3(1..8) = (0, 1, 14, 9, 7, 14, 5, 3) hartuz, T3(2..6) sekzioa
T1(1..2) eta T2(4..6)-ren permutazioa da.

b) Goranzko ordenean dagoen A(1..m) bektorea eta beheranzko ordenean dagoen B(1..n) bektorea emanda, C(1..m + n) bektorean A(1..m) eta B(1..n) nahastuta baina goranzko ordenean ipintzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataleko predikatuak erabiliz

Adibidea: A(1..6) = (3, 6, 8, 8, 13, 18) eta B(1..3) = (9, 5, 3) hartuz,
emaitza C(1..9) = (3, 3, 5, 6, 8, 8, 9, 13, 18) izango litzateke.

```
(1) {Hasierako baldintza}
i : = 1; j : = n; k : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} k ≤ m + n loop
        if i = m + 1 then (4) {Tarteko asertzioa}
                              C(k) : = B(j);
                              (5) {Tarteko asertzioa}
                               j : = j – 1;
                              (6) {Tarteko asertzioa}
        elsif j = 0 then (7) {Tarteko asertzioa}
                            C(k) : = A(i);
                            (8) {Tarteko asertzioa}
                             i : = i + 1;
                            (9) {Tarteko asertzioa}
        elsif A(i) ≤ B(j) then (10) {Tarteko asertzioa} C(k) : = A(i);
                                   (11) {Tarteko asertzioa} i : = i + 1;
                                  (12) {Tarteko asertzioa}
        else (13) {Tarteko asertzioa} C(k) : = B(j);
             (14) {Tarteko asertzioa} j : = j – 1;
        end if;
        (15) {Tarteko asertzioa} k : = k + 1;
        (16) {Tarteko asertzioa}
end loop;
(17) {Bukaerako baldintza}
(18) {E espresioa}
```

26. Bikoitiak(T(1..n)), Bakoitiak(T(1..n)) eta Perm(T1(1..n), T2(1..m), T3(1..p)) predikatuak eta A(1..n) emanda, hurrenez hurren A-ko elementu bikoitiak eta bakoitiak dituzten B(1..p) eta C(1..q) bektoreak itzultzen dituen programa

a) T(1..n) bektoreko elementu denak bikoitiak direla adierazten duen **Bikoitiak(T(1..n))** izeneko predikatua definitu.
b) T(1..n) bektoreko elementu denak bakoitiak direla adierazten duen **Bakoitiak(T(1..n))** izeneko predikatua definitu.
c) T1(1..n) bektorea T2(1..m) eta T3(1..p) bektoreetako elementuen permutazioa dela adierazten duen **Perm(T1(1..n), T2(1..m), T3(1..p))** izeneko predikatua definitu. T1 beste bien permutazioa izateak T1-eko elementu kopurua T2 eta T3 bektoreetako elementu kopuruen baturaren berdina dela eta T1-eko elementu bakoitzaren T1-eko agerpen kopurua T2 eta T3-ko agerpen kopuruen baturaren berdina dela esan nahi du.
Adibidea:
T1(1..7) = (9, 2, 2, 8, 0, 5, 8) bektorea T2(1..5) = (0, 2, 8, 2, 5) eta T3(1..2) = (8, 9) bektoreen permutazioa da.

d) A(1..n) emanda, hurrenez hurren A-ko elementu bikoitiak eta bakoitiak dituzten B(1..p) eta C(1..q) bektoreak itzultzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz.
Adibidea:
A(1..6) = (8, 8, 5, 2, 0, 5) hartuz B(1..4) = (8, 8, 2, 0) eta C(1..2) = (5, 5) itzuliko lituzke

```
(1) {Hasierako baldintza}
i : = 1;   p : = 0;   q : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
        (4) {Tarteko asertzioa}
        if A(i) mod 2 = 0
        then (5) {Tarteko asertzioa}
                B(p + 1) : = A(i);
                (6) {Tarteko asertzioa}
                p : = p + 1;
                  (7) {Tarteko asertzioa}
        else  (8) {Tarteko asertzioa}
                C(q + 1) : = A(i);
                (9) {Tarteko asertzioa}
                q : = q + 1;
                (10) {Tarteko asertzioa}
        end if;
        (11) {Tarteko asertzioa}
        i : = i + 1;
       (12) {Tarteko asertzioa}
end loop;
(7) {Bukaerako baldintza}
(8) {E espresioa}
```

27. tartekoa(C(1..p), D(1..p), x, z) predikatua eta A(1..n) bektorean min baino txikiagoak direnak min balioaz eta max baino handiagoak direnak max balioaz ordezkatzen dituen programa

a) x balioa z baino txikiagoa edo berdina dela, C-n x baino txikiagoa den balio bat dagoenean D-n x dagoela, C-n z baino handiagoa den balio bat dagoenean D-n z dagoela eta beste posizio denetan C eta D berdinak direla adierazten duen **tartekoa(C(1..p), D(1..p), x, z)** izeneko predikatua definitu.

Adibidea:

x = 2
Z = 10

C(1..8)	6	**-7**	**-10**	**20**	8	**43**	5	2
	1	2	3	4	5	6	7	8

D(1..8)	6	**2**	**2**	**10**	8	**10**	5	2
	1	2	3	4	5	6	7	8

b) A(1..n) bektorea eta *min* eta *max* (min ≤ max) bi balio emanda, A taulan min baino txikiagoa den zerbait dagoenean min ipiniz, A taulan max baino handiagoa den zerbait dagoenean max ipiniz eta beste kasuetan A-ko elementu bera ipiniz B(1..n) bektorea osatzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataleko predikatua erabiliz.

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
        (4) {Tarteko asertzioa}
        if A(i) < min then (5) {Tarteko asertzioa}
                              B(i) : = min;
                              (6) {Tarteko asertzioa}
        elsif A(i) > max then (7) {Tarteko asertzioa} B(i) : = max;
                                (8) {Tarteko asertzioa}
        else (9) {Tarteko asertzioa}
             B(i) : = A(i);
             (10) {Tarteko asertzioa}
        end if;
        (11) {Tarteko asertzioa}
        i : = i + 1;
        (12) {Tarteko asertzioa}
end loop;
(13) {Bukaerako baldintza}
(14) {E espresioa}
```

28. Bikoitiak(D(1..r)), Bakoitiak(D(1..r)) eta tartekatuta(D(1..r), E(1..s), F(1..t)) predikatuak eta A(1..n) eta B(1..p) bektoreak emanda (1 ≤ n ≤ p) eta A-ko elementu denak bikoitiak direla eta B-ko elementu denak bakoitiak direla jakinda, ahal den bitartean A-ko eta B-ko elementuak txandaka ipiniz eta A bukatutakoan B-n sobratu diren beste elementu denak jarraian ipiniz C(1..n + p) bektorea osatzen duen programa

a) D(1..r) bektoreko elementu denak bikoitiak direla adierazten duen **Bikoitiak(D(1..r))** izeneko predikatua definitu.

b) D(1..r) bektoreko elementu denak bakoitiak direla adierazten duen **Bakoitiak(D(1..r))** izeneko predikatua definitu.

c) r balioa s baino txikiagoa edo berdina dela, t balioa r gehi s dela, D-ko elementuak bikoitiak direla, E-ko elementuak bakoitiak direla, D eta E-ko elementuak tartekatuta agertzen direla F bektorean tartekatzea posible den lekuraino eta gero sobratu diren E-ko elementuak egongo direla F-n adierazten duen **tartekatuta(D(1..r), E(1..s), F(1..t))** predikatua definitu.

1. adibidea:
D(1..3), E(1..5) eta F(1..8) bektoreek predikatua bete egiten dute:

D(1..3)

8	20	6
1	2	3

E(1..5)

5	7	3	9	3
1	2	3	4	5

F(1..8)

8	5	20	7	6	3	9	3
1	2	3	4	5	6	7	8

2. adibidea:
Adibide honetako D(1..2), E(1..4) eta F(1..6) bektoreek ez dute betetzen predikatua D-ko eta E-ko elementuak ez daudelako tartekatuta F bektorean:

D(1..2)

4	10
1	2

E(1..4)

9	7	5	9
1	2	3	4

F(1..6)

4	7	9	10	5	9
1	2	3	4	5	6

d) A(1..n) eta B(1..p) bektoreak emanda ($1 \leq n \leq p$) eta A-ko elementu denak bikoitiak direla eta B-ko elementu denak bakoitiak direla jakinda, ahal den bitartean A-ko eta B-ko elementuak txandaka ipiniz eta A bukatutakoan B-n sobratu diren beste elementu denak jarraian ipiniz C(1..n + p) bektorea osatzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz.

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
     (4) {Tarteko asertzioa}
     C((i * 2) – 1) : = A(i);
      (5) {Tarteko asertzioa}
     C(i * 2) : = B(i);
      (6) {Tarteko asertzioa}
     i : = i + 1;
     (7) {Tarteko asertzioa}
end loop;
(8) {Tarteko asertzioa}
while (9) {Inbariantea} i ≠ p + 1 loop
     (10) {Tarteko asertzioa}
     C(n + i) : = B(i);
     (11) {Tarteko asertzioa}
     i : = i + 1;
     (12) {Tarteko asertzioa}
end loop;
(13) {Bukaerako baldintza}
(14) {Lehenengo while-aren E espresioa}
(15) {Bigarren while-aren E espresioa}
```

29. hiru(C(1..r)) eta aurrekoak(C(1..r), D(1..r)) predikatuak eta A(1..n) bektorean 0 edo 1 balioa badago, B(1..n) bektorean gauza bera ipini eta A(1..n) 2 badago B(1..n) 0 edo 1 (gutxien agertzen dena) ipintzen duen programa

a) C(1..r) bektorean bakarrik 0, 1 eta 2 balioak agertzen direla adierazten duen **hiru(C(1..r))** predikatua definitu.

1. adibidea:
Adibide honetako C(1..5) taulak predikatua bete egiten du:

C(1..5)	1	2	2	1	0
	1	2	3	4	5

2. adibidea:
Adibide honetako C(1..5) bektoreak ere predikatua bete egiten du, ez baita 0, 1 edo 2 ez den baliorik agertzen hor:

C(1..5)	2	0	2	2	2
	1	2	3	4	5

b) Jarraian zehazten dena betetzen dela adierazten duen **aurrekoak(C(1..r), D(1..r))** izeneko predikatua definitu:

- C-ko posizio batean 1 edo 0 dagoenean, D bektorean ere balio bera egongo da.
- C-ko posizio batean 2 balioa dagoenean D bektorean honako hau edukiko da:
  - D-ko aurreko posizioetan bateko kopurua zero kopurua baino txikiagoa bada, orduan 1.
  - Bestela, 0.

Adibidea:
Adibide honetako C(1..8) eta D(1..8) bektoreek predikatua bete egiten dute:

C(1..8)	0	**2**	0	**2**	1	**2**	1	1
	1	2	3	4	5	6	7	8

D(1..8)	0	**1**	0	**1**	1	**0**	1	1
	1	2	3	4	5	6	7	8

C-ko 2ren lehenengo bi agerpenak 1 balioaz ordezkatu dira D-n, D-ko aurreko posizioetan bateko kopurua zero kopurua baino txikiagoa delako. C-ko hirugarren 2a 0 balioaz ordezkatu da D taulan D-ko aurreko posizioetan zero-kopurua bateko kopurua baino txikiagoa delako.

c) Gutxienez elementu bat duen A(1..n) bektorea emanda, B(1..n) bektorea jarraian adierazten den bezala kalkulatzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:
   - A-ko posizio batean 1 edo 0 dagoenean, B bektorean ere balio bera gordeko da.
   - A-ko posizio batean 2 balioa dagoenean B bektoreko aurreko posizioetan bateko kopurua zero kopurua baino txikiagoa bada, orduan 1 gordeko da eta bestela 0 gordeko da.

```
(1) {Hasierako baldintza}
i : = 1; zeroak : = 0; batekoak : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    if A(i) = 0 then (5) {Tarteko asertzioa}
                     B(i) : = 0;
                     (6) {Tarteko asertzioa}
                     zeroak : = zeroak + 1;
                     (7) {Tarteko asertzioa}
    elsif A(i) = 1 then (8) {Tarteko asertzioa} B(i) : = 1;
                        (9) {Tarteko asertzioa}
                        batekoak : = batekoak + 1;
                        (10) {Tarteko asertzioa}
    elsif batekoak < zeroak then (11) {Tarteko asertzioa}
                            B(i) : = 1;
                            (12) {Tarteko asertzioa}
                            batekoak : = batekoak + 1;
                            (13) {Tarteko asertzioa}
    else (14) {Tarteko asertzioa} B(i) : = 0;
         (15) {Tarteko asertzioa} zeroak : = zeroak + 1;
         (16) {Tarteko asertzioa}
    end if;
    (17) {Tarteko asertzioa}
    i : = i + 1;
    (18) {Tarteko asertzioa}
end loop;
(19) {Bukaerako baldintza}
(20) {E espresioa}
```

30. bitak(D(1..p)) eta trukatuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), F(1..r), pos) predikatuak eta A(1..n), B(1..n) eta C(1..n) bektoreak emanda eta C(1..n) bektorean bakarrik zeroak eta batekoak daudela jakinda, C bektorean 1 dagoen posizioetan A eta B bektoreetako balioak trukatzen dituen programa

a) D(1..p) bektorean zeroak eta batekoak bakarrik daudela adierazten duen **bitak(D(1..p))** izeneko predikatua definitu.

b) Jarraian zehazten dena adierazten duen **trukatuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), F(1..r), pos)** predikatua definitu:

- F(1..r) bektorean batekoak eta zeroak bakarrik daude.
- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- F bektoreko 1 eta pos posizioen arteko edozein k posiziorentzat, posizo horretan 0 baldin badago, orduan D bektorean d_k eta E bektorean e_k daukagu posizio horretan.
- F bektoreko 1 eta pos posizioen arteko edozein k posiziorentzat, posizo horretan 1 baldin badago, orduan D bektorean e_k eta E bektorean d_k daukagu posizio horretan.

Predikatu hau definitzerakoan aurreko ataleko predikatua erabili behar da (bitak).

Adibidea:

Adibide honetako D(1..5), E(1..5) eta F(1..5) bektoreek honako hau betetzen dute:

trukatuta(D(1..5),(4, **5, 15**, 6, 5), E(1..5), (8, **9, 2**, 30, 7), F(1..5), 5):

D(1..7)	4	9	2	6	5	4	23
	1	2	3	4	5	6	7

E(1..7)	8	5	15	30	7	2	10
	1	2	3	4	5	6	7

F(1..7)	0	**1**	**1**	0	0	1	1
	1	2	3	4	5	6	7

Baina beste hauek ez dira betetzen:

trukatuta(D(1..7),(4, **5, 15**, 6, 5, 4, 23), E(1..7), (8, **9, 2**, 30, 7, 2, 10), F(1..7), 6)
trukatura (D(1..7),(4, **5, 15**, 6, 5, 4, 23), E(1..7), (8, **9, 2**, 30, 7, 2, 10), F(1..7), 7)

c) A(1..n), B(1..n) eta C(1..n) bektoreak emanda eta C(1..n) bektorean bakarrik zeroak eta batekoak daudela jakinda, C bektorean 1 dagoen posizioetan A eta B bektoreetako balioak trukatzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1){Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    if C(i) = 1 then (5) {Tarteko asertzioa}
                     lag : = A(i);
                     (6) {Tarteko asertzioa}
                     A(i) : = B(i);
                     (7) {Tarteko asertzioa}
                     B(i) : = lag;
                     (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
    i : = i + 1;
   (10) {Tarteko asertzioa}
end loop;
(11) {Bukaerako baldintza}
(12) {E espresioa}
```

31. gorantz(C(1..r)), permutazioa(C(1..r), (c_1, c_2, …, c_r)) eta tarteko_txik(E(1..r), g, h, pos) predikatuak eta A(1..n) bektorea emanda, elementuak goranzko ordenean ipiniz bektorea ordenatzen duen programa

a) C(1..r) taulako elementuak goranzko ordenean daudela (elementu bakoitza hurrengoa baino txikiagoa dela) adierazten duen **gorantz(C(1..r))** predikatua definitu.

b) D(1..r) bektorea (d_1, d_2, …, d_r) bektorearen permutazioa dela, hau da, D-ko elementu bakoitzaren agerpen-kopurua D(1..r) bektorean eta (d_1, d_2, …, d_r) bektorean berdina dela adierazten duen **permutazioa(D(1..r), (d_1, d_2, …, d_r))** predikatua definitu.

c) Jarraian zehazten dena adierazten duen **tarteko_txik(E(1..r), g, h, pos)** predikatua definitu:
   - g balioa 1 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da eta h balioa g baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
   - pos posizioa g eta h balioek definitutako tartean dago.
   - pos posizioko elementua g eta h balioek definitutako tarteko balio txikiena da.

<u>Adibidea</u> (tarteko_txik predikatuarentzat):
E(1..10) bektoreak eta g = 3, h = 8 eta pos = 6 balioek **tarteko_txik(E(1..10), 3, 8, 6)** predikatua bete egiten dute 3 eta 8 posizioen arteko balio txikiena 6 posizioan dagoelako:

E(1..10)	14	0	10	10	4	**2**	6	20	-6	0
	1	2	3	4	5	6	7	8	9	10

Tarte honetako txikiena 6 posizioan dago

d) A(1..n) bektorea emanda, elementuak goranzko ordenean ipiniz bektorea ordenatzen duen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    postxik : = txikienaren_posizioa(A(1..n), i);
    (5) {Tarteko asertzioa}
    lag : = A(i);
    (6) {Tarteko asertzioa}
    A(i) : = A(postxik);
    (7) {Tarteko asertzioa}
    A(postxik) : = lag;
    (8) {Tarteko asertzioa}
    i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

Programak *txikienaren_posizioa* izeneko **funtzio laguntzailea** erabiltzen du. Funtzio horrek, bektore bat eta bektoreko posizio bat emanda, posizio horretatik aurrera balio txikiena denaren posizioa itzultzen du (funtzio hori definituta dagoela suposatu behar da).

Adibidea (*txikienaren_posizioa* izeneko **funtzio laguntzailea**rentzat**):**

E(1..10) bektorea hartuz

E(1..10)	14	-20	10	10	4	2	6	20	**-6**	0
	1	2	3	4	5	6	7	8	9	10

txikienaren_posizioa(E(1..10), 3) deiak 9 itzuliko luke 3 posiziotik aurrera balio txikiena 9 posizioan dagoelako.

32. txikiagoak(C(1..r), D(1..r), pos) eta permutazioa(E(1..r), F(1..r), (e_1, e_2, …, e_r), (f_1, f_2, …, f_r)) predikatuak eta A(1..n) eta B(1..n) bi bektore emanda, A-ko posizio bateko elementua B-ko posizio bereko elementua baino handiago denean balio horiek trukatu egiten dituen programa

a) C(1..r) bektorean 1 eta pos posizioen arteko elementu bakoitza D(1..r) bektoreko posizio berean dagoena baino txikiagoa edo berdina dela adierazten duen **txikiagoak(C(1..r), D(1..r), pos)** predikatua definitu.

b) E(1..r) eta F(1..r) batera kontsideratuz daukaguna (e_1, e_2, …, e_r) eta (f_1, f_2, …, f_r) batera hartuz daukagunaren permutazioa dela adierazten duen **permutazioa(E(1..r), F(1..r), (e_1, e_2, …, e_r), (f_1, f_2, …, f_r))** predikatua definitu. Beraz predikatu horrek honako hau adierazi behar du:

- E(1..r)-ko elementu bakoitzarentzat elementu horren E(1..r) bektoreko agerpen kopurua gehi F(1..r) bektoreko agerpen kopurua, (e_1, e_2, …, e_r) bektoreko agerpen kopurua gehi (f_1, f_2, …, f_r) bektoreko agerpen kopuruaren berdina da.
- F(1..r)-ko elementu bakoitzarentzat elementu horren E(1..r) bektoreko agerpen kopurua gehi F(1..r) taulako agerpen kopurua, (e_1, e_2, …, e_r) tbektoreko agerpen kopurua gehi (f_1, f_2, …, f_r) bektoreko agerpen kopuruaren berdina da.

Adibidea (permutazioarentzat):
Adibide honetako E(1..5) eta F(1..5) bektoreek honako hau betetzen dute:

permutazioa(E(1..5), F(1..5), (4, 4, 15, 6, 8), (8, 9, 7, 8, 7)):

E(1..5)	8	4	8	6	8
	1	2	3	4	5

F(1..5)	7	4	15	9	7
	1	2	3	4	5

c) A(1..n) eta B(1..n) bi bektore emanda, A-ko posizio bateko elementua B-ko posizio bereko elementua baino handiago denean balio horiek trukatu egiten dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    if A(i) > B(i) then (5) {Tarteko asertzioa}
                        lag : = A(i);
                        (6) {Tarteko asertzioa}
                        A(i) : = B(i);
                        (7) {Tarteko asertzioa}
                        B(i) : = lag;
                        (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
    i : = i + 1;
    (10) {Tarteko asertzioa}
end loop;
(11) {Bukaerako baldintza}
(12) {E espresioa}
```

33. bikoitia(x) eta birkokatuta(C(1..r), (c_1, c_2, …, c_r), pos) predikatuak eta n bikoitia duen A(1..n) bektorea emanda, posizio bikoiti bateko balioa bere aurreko posizio bakoitiko balioa baino txikiagoa denean balio horiek lekuz trukatzen dituen programa

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) Aurreko ataleko predikatua erabiliz jarraian zehazten dena adierazten duen **birkokatuta(C(1..r), (c_1, c_2, …, c_r),** pos**)** izeneko predikatua definitu:

- r bikoitia da.
- Pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 2 eta pos-en artean dauden k posizio bikoitietan c_k balioa c_{k-1} balioa baino txikiagoa denean, C(k)-ren balioa c_{k-1} izango da eta C(k – 1)-ren balioa c_k izango da.
- 2 eta pos-en artean dauden k posizio bikoitietan c_k balioa c_{k-1} balioa baino txikiagoa ez denean, C(k)-ren balioa c_k izango da eta C(k – 1)-ren balioa c_{k-1} izango da.

Adibidea (birkokatuta predikatuarentzat):

Adibide honetako C(1..6) eta (4, 2, 4, 8, 5, 1) bektoreek honako hau betetzen dute:

birkokatuta(C(1..6), (4, 2, 4, 8, 5, 1), 4):

C(1..6)	2	7	7	8	9	1
	1	2	3	4	5	6

c) n bikoitia duen A(1..n) bektorea emanda, posizio bikoiti bateko balioa bere aurreko posizio bakoitiko balioa baino txikiagoa denean balio horiek lekuz trukatzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 2;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 2 loop
    (4) {Tarteko asertzioa}
    if A(i – 1) > A(i) then    (5) {Tarteko asertzioa}
                               lag : = A(i – 1);
                               (6) {Tarteko asertzioa}
                               A(i – 1) : = A(i);
                               (7) {Tarteko asertzioa}
                               A(i) : = lag;
                              (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
    i : = i + 2;
   (10) {Tarteko asertzioa}
end loop;
(11) {Bukaerako baldintza}
(12) {E espresioa}
```

34. perm(C(1..r), (c_1, c_2, …, c_r)) eta hasieran(C(1..r), pos) predikatuak eta A(1..n) bektorea emanda, A(1) balioaren agerpen denak bektorearen ezkerreko aldean (1 eta p posizioen artean) eta A(1)-en desberdinak diren balio denak eskuineko aldean (p + 1 eta n posizioen artean) ipintzen dituen programa

a) C(1..r) bektorea (c_1, c_2, …, c_r) bektorearen permutazio bat dela adierazten duen **perm(C(1..r), (c_1, c_2, …, c_r))** predikatua definitu. C(1..r) bektorea (c_1, c_2, …, c_r) bektorearen permutazioa izateak C-ko elementu bakoitzaren C-ko agerpen-kopurua eta (c_1, c_2, …, c_r) bektoreko agerpen-kopurua berdinak izango direla esan nahi du.

b) Jarraian zehazten dena adierazten duen **hasieran(C(1..r), pos)** predikatua definitu:

- r balioa 1 baino handiagoa edo berdina da.
- pos balioa 1 baino handiagoa edo berdina da eta r baino txikiagoa edo berdina da. Beste era batera esanda, pos balioa C(1..r) bektoreko posizio bat da.
- 1 eta pos posizioen arteko elementu denak (biak barne) C(1) elementuaren berdinak dira, hau da, bektoreko lehenengo elementuaren berdinak dira.
- pos + 1 eta r posizioen arteko elementu denak (biak barne) C(1) elementuaren desberdinak dira, hau da, bektoreko lehenengo elementuaren desberdinak dira.

Adibidea ("hasieran" predikatuarentzat):

Adibide honetako D(1..6) bektoreak eta 3 balioak predikatua bete egiten dute:

hasieran(D(1..6), 3)

D(1..6)	8	8	8	5	1	5
	1	2	3	4	5	6

c) A(1..n) bektorea emanda, A(1) balioaren agerpen denak bektorearen ezkerreko aldean (1 eta p posizioen artean) eta A(1)-en desberdinak diren balio denak eskuineko aldean (p + 1 eta n posizioen artean) ipintzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 2;   p : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    if A(i) = A(1) then (5) {Tarteko asertzioa} A(i) : = A(p + 1);
                        (6) {Tarteko asertzioa} A(p + 1) : = A(1);
                        (7) {Tarteko asertzioa} p : = p + 1;
                        (8) {Tarteko asertzioa}
    end if;
    (9) {Tarteko asertzioa}
    i : = i + 1;
    (10) {Tarteko asertzioa}
end loop;
(11) {Bukaerako baldintza}
(12) {E espresioa}
```

35. (2008ko apirila #1) ezzerobat(C(1..r)), bitak(C(1..r)) eta ordezkatuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), pos) predikatuak eta Zerorik eta batekorik ez duen A(1..n) bektorea eta bakarrik zeroak eta batekoak dituen B(1..n) bektorea emanda eta n ≥1 dela jakinda, B(1..n) bektoreko zero bakoitza eta A(1..n) bektoreko pareko elementua trukatzen dituen programa

a) C(1..r) bektorean zerorik eta batekorik ez dagoela adierazten duen **ezzerobat(C(1..r))** izeneko **predikatua definitu** 1.

b) C(1..r) bektorean bakarrik zeroak eta batekoak daudela adierazten duen **bitak(C(1..r))** izeneko **predikatua definitu**.

c) Jarraian zehazten dena adierazten duen

ordezkatuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), pos) predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - e_k balioa 0 bada, orduan D(k) balioa 0 da eta E(k) balioa d_k balioaren berdina da.
  - e_k balioa 1 bada, orduan D(k) balioa d_k balioaren berdina da eta E(k) balioa e_k balioaren berdina da.

Adibidea:

Adibide honetako D(1..8), (4, 9, 2, 6, 20, 0, 0, 1), E(1..8) eta (0, 1, 1, 0, 0, 0, 0, 1) bektoreek honako hau betetzen dute:

ordezkatuta(D(1..8), (**4, 9, 2, 6, 20**, 18, -9, 14), E(1..8), (**0, 1, 1, 0, 0**, 0, 0, 1), **5**)

D(1..8)	**0**	**9**	**2**	**0**	**0**	18	-9	14
	1	**2**	**3**	**4**	**5**	6	7	8

E(1..8)	**4**	**1**	**1**	**6**	**20**	0	0	1
	1	**2**	**3**	**4**	**5**	6	7	8

d) Zerorik eta batekorik ez duen A(1..n) bektorea eta bakarrik zeroak eta batekoak dituen B(1..n) bektorea emanda eta $n \geq 1$ dela jakinda, B(1..n) bektoreko zero bakoitza eta A(1..n) bektoreko posizio bereko elementua trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
    (4) {Tarteko asertzioa}
    if B(i + 1) = 0 then (5) {Tarteko asertzioa}
                      B(i + 1) : = A(i + 1);
                      (6) {Tarteko asertzioa}
                      A(i + 1) : = 0;
                      (7) {Tarteko asertzioa}
    end if;
    (8) {Tarteko asertzioa}
    i : = i + 1;
    (9) {Tarteko asertzioa}
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

36. (2008ko apirila #2) bikoitiak(C(1..r)), bakoitiak(C(1..r)) eta mugituta(D(1..s), (d_1, d_2, …, d_s), E(1..s), (e_1, e_2, …, e_s), F(1..s), (f_1, f_2, …, f_s), pos) predikatuak eta A(1..n) bektorean bakarrik zenbaki bikoitiak daudela eta C(1..n) bektorean bakarrik zenbaki bakoitiak daudela jakinda, B(1..n) bektoreko posizio batean elementu bikoitia dagoenean posizio horretako A(1..n) eta B(1..n) bektoreetako elementuak trukatzen dituen eta B(1..n) bektoreko posizio batean elementu bakoitia dagoenean posizio horretako A(1..n) eta C(1..n) bektoreetako elementuak trukatzen dituen programa

a) C(1..r) bektoreko elementu denak bikoitiak direla adierazten duen **bikoitiak(C(1..r))** izeneko predikatua definitu.

b) C(1..r) bektoreko elementu denak bakoitiak direla adierazten duen **bakoitiak(C(1..r))** izeneko predikatua definitu.

a) Aurreko ataletako predikatuak erabiliz jarraian zehazten dena adierazten duen

mugituta(D(1..s), (d_1, d_2, …, d_s), E(1..s), (e_1, e_2, …, e_s), F(1..s), (f_1, f_2, …, f_s), pos)

predikatua definitu:

- D(1..s) bektoreko elementu denak bikoitiak dira eta F(1..s) bektoreko elementu denak bakoitiak dira.
- **pos** balioa 0 baino handiagoa edo berdina eta s baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - e_k bikoitia bada, orduan D(k) eta e_k berdinak dira, E(k) eta d_k berdinak dira eta F(k) eta f_k berdinak dira
  - e_k bakoitia bada, orduan D(k) eta d_k berdinak dira, E(k) eta f_k berdinak dira eta F(k) eta e_k berdinak dira

Adibidea:

Adibide honetako D(1..8), (4, 6, 2, 8, 20, 18, 8, 14), E(1..8), (2, 8, 1, 2, 9, 2, 4, 9), F(1..8) eta (3, 9, 3, 7, 11, 3, 15, 7) bektoreek honako hau betetzen dute:

mugituta(D(1..8), (4, 6, 2, 8, 20, 18, 8, 14), E(1..8), (2, 8, 1, 2, 9, 2,4, 9), F(1..8), (3, 9, 3, 7, 11, 3, 15, 7), 5):

D(1..8)	**2**	**8**	**2**	**2**	**20**	18	8	14
	1	**2**	**3**	**4**	**5**	6	7	8

E(1..8)	**4**	**6**	**3**	**8**	**11**	2	4	9
	1	**2**	**3**	**4**	**5**	6	7	8

F(1..8)	**3**	**9**	**1**	**7**	**9**	3	15	7
	1	**2**	**3**	**4**	**5**	6	7	8

d) Zenbaki osozko A(1..n), B(1..n) eta C(1..n) bektoreak emanda eta n ≥1 dela eta A(1..n) bektorean bakarrik zenbaki bikoitiak daudela eta C(1..n) bektorean bakarrik zenbaki bakoitiak daudela jakinda, B(1..n) bektoreko posizio batean elementu bikoitia dagoenean posizio horretako A(1..n) eta B(1..n) bektoreetako elementuak trukatzen dituen eta B(1..n) bektoreko posizio batean elementu bakoitia dagoenean posizio horretako A(1..n) eta C(1..n) bektoreetako elementuak trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n + 1 loop
    (4) {Tarteko asertzioa}
    if B(i) mod 2 = 0 then (5) {Tarteko asertzioa}
                            lag := A(i);
                             (6) {Tarteko asertzioa}
                            A(i) : = B(i);
                             (7) {Tarteko asertzioa}
                            B(i) : = lag;
    else
        lag := C(i);
        C(i) := B(i);
        B(i) := lag;
    end if;
    (8) {Tarteko asertzioa}
    i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

37. (2008ko ekaina) bikoitia(x) eta trukabik(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) predikatuak eta 1 eta n-ren arteko posizio batean bai A(1..n) bektorean eta bai B(1..n) bektorean elementu bikoitiak badaude, bektorez trukatzen dituen programa

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.
b) Aurreko ataleko predikatua erabiliz jarraian zehazten dena adierazten duen:

trukabik(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) predikatua definitu:

- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne), c_k eta d_k bikoitiak badira, C(k) balioa d_k–ren berdina da eta D(k) balioa c_k-ren berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne), c_k edo d_k (gutxienez bietako bat) bakoitia bada, C(k) balioa c_k–ren berdina da eta D(k) balioa d_k-ren berdina da.

Adibidea (trukabik predikatuarentzat):
Adibide honetako C(1..6), (4, 9, 8, 2, 15, 20), D(1..6), (2, 10, 6, 8, 3, 12) bektoreek honako hau betetzen dute:

trukabik(C(1..6), (**4, 9, 8**, 2, 15, 20), D(1..6), (**2, 10, 6**, 8, 3, 12), **3**)

baina ez dute betetzen beste hau:

trukabik(C(1..6), (**4, 9, 8, <u>2</u>**, 15, 20), D(1..6), (**2, 10, 6, <u>8</u>**, 3, 12), **4**)

izan ere (4, 9, 8, **2**, 15, 20) eta (2, 10, 6, **8**, 3, 12) bektoreetako 4. posizioetako elementuak bikoitiak dira baina C(1..6) eta D(1..6) bektoreetan ez daude lekuz trukatuta:

C(1..6)	2	9	6	2	15	20
	1	2	3	4	5	6

D(1..6)	4	10	8	8	3	12
	1	2	3	4	5	6

c) Zenbaki osozko A(1..n) eta B(1..n) bektoreak emanda, 1 eta n-ren arteko posizio batean bai A(1..n) bektorean eta bai B(1..n) bektorean elementu bikoitiak badaude, bektorez trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
  (4) {Tarteko asertzioa}
  if A(i) mod 2 = 0 and B(i) mod 2 = 0 then
                              (5) {Tarteko asertzioa}
                              lag : = A(i);
                              (6) {Tarteko asertzioa}
                              A(i) : = B(i);
                              (7) {Tarteko asertzioa}
                              B(i) : = lag;
    end if;
    (8) {Tarteko asertzioa}
    i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

38. (2008ko iraila) bikoitia(x) eta bikmugi(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) predikatuak eta A(1..n) bektoreko posizioetan elementu bikoitia dagoen bakoitzean, posizio horretako A(1..n) eta B(1..n) bektoreetako elementuak trukatu eta A(1..n) bektoreko posizioetan elementu bakoitia dagoen bakoitzean, A(1..n) bektoreko posizio horretan -1 gorde eta B(1..n) bektoreko posizio horretan ezer egiten ez duen programa

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) Aurreko ataleko predikatua erabiliz jarraian zehazten dena adierazten duen **bikmugi(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos)** predikatua definitu:
- **r** balioa 1 baino handiagoa edo berdina da.
- **pos** balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne), c_k bikoitia bada, C(k) balioa d_k–ren berdina da eta D(k) balioa c_k-ren berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne), c_k bakoitia bada, C(k)-ren balioa -1 da eta D(k) balioa d_k-ren berdina da.

Adibidea (bikmugi predikatuarentzat):

Adibide honetako C(1..6), (10, 5, 8, 15, 2, 17), D(1..6) y (11, 0, 6, 68, 33, 80) bektoreek honako hau betetzen dute:

bikmugi(C(1..6), (10, 5, 8, 15, 2, 17), D(1..6), (11, 0, 6, 68, 33, 80), **3**)

baina beste hau ez dute betetzen

bikmugi(C(1..6), (10, 5, 8, 15, 2, 17), D(1..6), (11, 0, 6, 68, 33, 80), **4**)

C taulako 4 posizioan -1 balioa ez dagoelako:

C(1..6)	11	-1	6	15	2	17
	1	2	3	4	5	6

D(1..6)	10	0	8	68	33	80
	1	2	3	4	5	6

c) Zenbaki osozko A(1..n) eta B(1..n) bektoreak emanda, jarraian zehazten dena burutzen duen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:
   - A(1..n) bektoreko posizioetan elementu bikoitia dagoen bakoitzean, posizio horretako A(1..n) eta B(1..n) bektoreetako elementuak trukatu.
   - A(1..n) bektoreko posizioetan elementu bakoitia dagoen bakoitzean, A(1..n) bektoreko posizio horretan -1 gorde eta B(1..n) bektoreko posizio horretan ezer ez egin.

n-ren balioa 1 baino handiagoa edo berdina izango da:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i < n loop
  (4) {Tarteko asertzioa}
  i : = i + 1;
  (5) {Tarteko asertzioa}
  if A(i) mod 2 = 0   then   (6) {Tarteko asertzioa}
                             lag : = A(i);
                             (7) {Tarteko asertzioa}
                             A(i) : = B(i);
                             (8) {Tarteko asertzioa}
                             B(i) : = lag;
                             (9) {Tarteko asertzioa}
  else  A(i) : = -1;
  end if;
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

39. (2009ko apirila #1) bikoitia(x) eta batuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), pos) predikatuak eta A(1..n)-ko elementu bikoitiak B(1..n) taulan posizio berean dauden elementuei batzen dizkien eta A(1..n)-ko posizio horietan 0 balioa gordetzen duen programa

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) Jarraian zehazten dena adierazten duen **batuta(D(1..r), (d_1, d_2, …, d_r), E(1..r), (e_1, e_2, …, e_r), pos)** predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - d_k balioa bikoitia bada, orduan D(k) balioa 0 da eta E(k) balioa d_k + e_k balioaren berdina da
  - d_k ez bada bikoitia, orduan D(k) balioa d_k balioaren berdina da eta E(k) balioa e_k balioaren berdina da.

Predikatu hau definitzerakoan a) ataleko predikatua erabili behar da.

Adibidea:

Adibide honetako D(1..8), (4, 9, 2, 6, 17, 4, 9, 2), E(1..8) eta (7, 8, 2, 1, 3, 5, 15, 3) bektoreek honako hau betetzen dute:

batuta(D(1..8), (**4, 9, 2, 6, 17**, 4, 9, 2), E(1..8), (**7, 8, 2, 1, 3**, 5, 15, 3), 5)

D(1..8)	**0**	**9**	**0**	**0**	**17**	4	9	2
	1	**2**	**3**	**4**	**5**	6	7	8

E(1..8)	**11**	**8**	**4**	**7**	**3**	5	15	3
	1	**2**	**3**	**4**	**5**	6	7	8

c) Zenbaki osoz osatutako A(1..n) eta B(1..n) bektoreak emanda eta n ≥ 1 dela jakinda, A(1..n)-ko elementu bikoitiak B(1..n) taulan posizio berean dauden elementuei batzen dizkien eta A(1..n)-ko posizio horietan 0 balioa gordetzen duen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
    (4) {Tarteko asertzioa}
    i : = i + 1;
    (5) {Tarteko asertzioa}
    if A(i) mod 2 = 0 then (6) {Tarteko asertzioa}
                      B(i) : = A(i) + B(i);
                      (7) {Tarteko asertzioa}
                      A(i) : = 0;
    end if;
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```

40. (2009ko apirila #2) kontrakoak(C(1..r), D(1..r)) eta aldatutaneg(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) predikatuak eta kontrako zeinua duten A(1..n) eta B(1..n) taulak hartuz, A(1..n) taulako elementu negatibo bakoitza B(1..n) taulako posizio bereko elementuaz trukatuko duen programa

a) Jarraian zehazten dena adierazten duen **kontrakoak(C(1..r), D(1..r))** predikatua definitu:

- 1 eta r balioen artean dauden k posizioetan (1 eta pos posizioak barne), C taulako elementua negatiboa (< 0) bada, D taulakoa ez da izango negatiboa (≥ 0).
- 1 eta r balioen artean dauden k posizioetan (1 eta pos posizioak barne), C taulako elementua negatiboa ez bada (≥ 0), orduan D taulakoa negatiboa (< 0) izango da.

Adibidea:
Adibide honetako C(1..8) eta D(1..8) bektoreek honako hau betetzen dute:

kontrakoak(C(1..8), D(1..8))

C(1..8)	3	-8	-5	0	7	-4	-2	7
	1	2	3	4	5	6	7	8

D(1..8)	-25	0	4	-35	-12	10	15	-3
	1	2	3	4	5	6	7	8

b) Jarraian zehazten dena adierazten duen:

aldatutaneg(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos)

predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - c_k balioa negatiboa bada, orduan C(k)-ren balioa d_k da eta D(k)-ren balioa c_k da
  - c_k balioa negatiboa ez bada, orduan C(k)-ren balioa c_k da eta D(k)-ren balioa d_k da.

Adibidea:
Adibide honetako C(1..8), (5, -40, -5, 0, 7, -4, -24, 77), D(1..8) eta (-60, -6, 4, -80, 5, 90, 15, -6) bektoreek honako hau betetzen dute:

aldatutaneg (C(1..8), (**5, -40, -5, 0, 7**, -4, -24, 77), D(1..8),
(**-60, -6, 4, -80, 5**, 90, 15, -6), 5):

C(1..8)	**5**	**-6**	**4**	**0**	**7**	-4	-24	77
	1	**2**	**3**	**4**	**5**	6	7	8

D(1..8)	**-60**	**-40**	**-5**	**-80**	**5**	90	15	-6
	1	**2**	**3**	**4**	**5**	6	7	8

c) Jarraian erakusten den programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

- Programari sarrera bezala zenbaki osoz osatutako A(1..n) eta B(1..n) taulak emango zaizkio. Bektore horiek gutxienez elementu bat izango dute **(n ≥ 1)** eta gainera bi taula horietako elementuek kontrako zeinua izango dute, hau da, **A(1..n) taulako posizio batean zenbaki negatibo bat** (< 0) dagoenean, posizio horretan **B(1..n) taulan negatiboa ez den** (≥ 0) **zenbaki bat egongo da** eta **A(1..n) taulako posizio batean negatiboa ez den** (≥ 0) **zenbaki bat** dagoenean, posizio horretan **B(1..n) taulan negatiboa** (< 0) **den zenbaki bat egongo da**.

- Programak A(1..n) taulako elementu negatibo bakoitza B(1..n) taulako posizio bereko elementuaz trukatuko du:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
     (4) {Tarteko asertzioa}
     if A(i + 1) < 0 then (5) {Tarteko asertzioa}
                          lag : = A(i + 1);
                          (6) {Tarteko asertzioa}
                          A(i + 1) : = B(i + 1);
                          (7) {Tarteko asertzioa}
                          B(i + 1) : = lag;
     end if;
     (8) {Tarteko asertzioa}
     i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

41. (2009ko ekaina) bikoitia(x), bikobako(C(1..r), D(1..r)) eta aldatutabiko(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos) predikatuak eta posizioz posizio kontrako balioz (bikoitia/bakoitia) osatuta dauden B(1..n) eta A(1..n) taulak emanda, A(1..n) taulako elementu bakoiti bakoitzari 1 kenduko dion eta B(1..n) taulako elementu bikoiti bakoitzari 1 batuko dion programa

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) Aurreko ataleko predikatua erabiliz jarraian zehazten dena adierazten duen **bikobako(C(1..r), D(1..r))** predikatua definitu:

- 1 eta pos-en artean dagoen k posizio bakoitzean (1 eta pos posizioak barne), C taulako balioa bikoitia bada, orduan D taulako balioa bakoitia izango da.
- 1 eta pos-en artean dagoen k posizio bakoitzean (1 eta pos posizioak barne), C taulako balioa bakoitia bada, orduan D taulako balioa bikoitia izango da.

Adibidea:

Adibide honetako C(1..8) eta D(1..8) bektoreek honako hau betetzen dute:

bikobako(C(1..8), D(1..8))

C(1..8)	4	8	7	0	9	5	-2	7
	1	2	3	4	5	6	7	8

D(1..8)	-25	3	18	7	16	10	15	2
	1	2	3	4	5	6	7	8

c) Jarraian zehazten dena adierazten duen **aldatutabiko(C(1..r), (c_1, c_2, …, c_r), D(1..r), (d_1, d_2, …, d_r), pos)** predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - c_k balioa bikoitia ez bada, orduan C(k)-ren balioa c_k – 1 da eta D(k)-ren balioa d_k + 1 da
  - c_k balioa bikoitia bada, orduan C(k)-ren balioa c_k da eta D(k)-ren balioa d_k da.

Adibidea:

Adibide honetako C(1..8), (4, 8, 7, 0, 9, 5, -2, 7), D(1..8) eta (-25, 3, 18, 7, 16, 10, 15, 2) bektoreek honako hau betetzen dute:

aldatutabiko(C(1..8), (**4, 8, 7, 0, 9**, 5, -2, 7), D(1..8), (**-25, 3, 18, 7, 16**, 10, 15, 2), **5**):

C(1..8)	**4**	**8**	**6**	**0**	**8**	5	-2	7
	1	**2**	**3**	**4**	**5**	6	7	8

D(1..8)	**-25**	**3**	**19**	**7**	**17**	10	15	2
	1	**2**	**3**	**4**	**5**	6	7	8

d) Jarraian erakusten den programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

- Programari sarrera bezala zenbaki osoz osatutako A(1..n) eta B(1..n) taulak emango zaizkio. Bektore horiek gutxienez elementu bat izango dute **(n ≥ 1)** eta gainera **A(1..n) taulako posizio batean zenbaki bikoiti bat** dagoenean, posizio horretan **B(1..n) taulan bakoitia den zenbaki bat egongo da** eta **A(1..n) taulako posizio batean bakoitia den zenbaki bat** dagoenean, posizio horretan **B(1..n) taulan bikoitia den zenbaki bat egongo da**.

- Programak A(1..n) taulako elementu bakoiti bakoitzari 1 kenduko dio eta B(1..n) taulako elementu bikoiti bakoitzari 1 batuko dio:

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
  (4) {Tarteko asertzioa}
  if A(i) mod 2 ≠ 0 then
                    (5) {Tarteko asertzioa}
                    A(i) : = A(i) – 1;
                    (6) {Tarteko asertzioa}
                    B(i) : = B(i) + 1;
                    (7) {Tarteko asertzioa}
  end if;
  (8) {Tarteko asertzioa}
  i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

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) C(1..r) bektorean zerorik eta batekorik ez dagoela adierazten duen **ezzerobat(C(1..r))** izeneko **predikatua definitu**.
b) C(1..r) bektorean bakarrik zeroak eta batekoak daudela adierazten duen **bitak(C(1..r))** izeneko **predikatua definitu**.
c) Jarraian zehazten dena adierazten duen **ordezkatuta(D(1..q), (d_1, d_2, …, d_q), E(1..q), (e_1, e_2, …, e_q), pos)**

predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta q baino txikiagoa edo berdina da.
- 1 eta pos-en artean (1 eta pos posizioak barne) dagoen edozein k posizio hartuta:
  - e_k balioa 0 bada, orduan D(k) balioa 0 da eta E(k) balioa d_k balioaren berdina da.
  - e_k balioa 1 bada, orduan D(k) balioa d_k balioaren berdina da eta E(k) balioa e_k balioaren berdina da.

Adibidea:
Adibide honetako D(1..8), (4, 9, 2, 6, 20, 18, -9, 14), E(1..8) eta (0, 1, 1, 0, 0, 0, 0, 1) bektoreek honako hau betetzen dute:

ordezkatuta(D(1..8), (**4, 9, 2, 6, 20**, 18, -9, 14), E(1..8), (**0, 1, 1, 0, 0**, 0, 0, 1), **5**)

D(1..8)	**0**	**9**	**2**	**0**	**0**	18	-9	14
	1	**2**	**3**	**4**	**5**	6	7	8

E(1..8)	**4**	**1**	**1**	**6**	**20**	0	0	1
	1	**2**	**3**	**4**	**5**	6	7	8

d) Zerorik eta batekorik ez duen A(1..n) bektorea eta bakarrik zeroak eta batekoak dituen B(1..n) bektorea emanda eta $n \geq 1$ dela jakinda, B(1..n) bektoreko zero bakoitza eta A(1..n) bektoreko posizio bereko elementua trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz:

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
     (4) {Tarteko asertzioa}
     i : = i + 1;
     (5) {Tarteko asertzioa}
     if B(i) = 0 then (6) {Tarteko asertzioa}
                      B(i) : = A(i);
                      (7) {Tarteko asertzioa}
                      A(i) : = 0;
                      (8) {Tarteko asertzioa}
     end if;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

43. (2010eko apirila #1) bikoitia(x), bakoitiak(D(1..r)) eta bikbik (E(1..r), (e_1, e_2, …, e_r), F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), pos) predikatuak eta C(1..n) bektoreko elementu denak bakoitiak direla jakinda, A(1..n) bektoreko eta B(1..n) bektoreko posizio berean zenbaki bikoitiak dauden bakoitzean posizio horretako B(1..n) eta C(1..n) bektoreetako elementuak trukatzen dituen programa.

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) D(1..p) bektoreko elementu denak bakoitiak direla adierazten duen **bakoitiak(D(1..r))** predikatua definitu a) ataleko predikatua erabiliz

c) Jarraian zehazten dena adierazten duen **bikbik (E(1..r), (e_1, e_2, …, e_r), F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), pos)** predikatua definitu:

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne), E(k) bikoitia baldin bada, orduan F(k) bakoitia da.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - ➢ e_k eta f_k balioak bikoitiak baldin badira, orduan E(k) balioa e_k da, F(k) balioa g_k-ren berdina da eta G(k) balioa f_k-ren berdina da.
  - ➢ e_k edo f_k (gutxienez bietako bat) bakoitia baldin bada, orduan E(k) balioa e_k da, F(k) balioa f_k-ren berdina da eta G(k) balioa g_k-ren berdina da.

Predikatu hau definitzerakoan a) ataleko predikatua erabili behar da.

Adibidea:

Adibide honetako E(1..8), (**4, 9, 10, 12, 17**, 30, 4, 2), F(1..8), (**7, 8, 6, 4, 3**, 18, 39, 3), G(1..8) eta (**19, 11, 25, 1, 3**, 5, 15, 3) bektoreek honako hau betetzen dute:

bikbik(E(1..8), (**4, 9, 10, 12, 17**, 30, 4, 2), F(1..8), (**7, 8, 6, 4, 3**, 18, 39, 3),
G(1..8), (**19, 11, 25, 1, 3**, 43, 15, 3), **5**):

E(1..8)	**4**	**9**	**10**	**12**	**17**	30	4	2
	1	**2**	**3**	**4**	**5**	6	7	8

F(1..8)	**7**	**8**	**25**	**1**	**3**	18	39	3
	1	**2**	**3**	**4**	**5**	6	7	8

G(1..8)	**19**	**11**	**6**	**4**	**3**	43	15	3
	1	**2**	**3**	**4**	**5**	6	7	8

d) Zenbaki osoz osatutako A(1..n), B(1..n) eta C(1..n) bektoreak emanda eta $n \geq 1$ dela eta hasieran C(1..n) bektoreko elementu denak bakoitiak direla jakinda, A(1..n) bektoreko eta B(1..n) bektoreko posizio berean zenbaki bikoitiak dauden bakoitzean posizio horretako B(1..n) eta C(1..n) bektoreetako elementuak trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz.

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
    (4) {Tarteko asertzioa}
    if A(i + 1) mod 2 = 0 and B(i + 1) mod 2 = 0
    then (5) {Tarteko asertzioa}
            lag : = B(i + 1);
            (6) {Tarteko asertzioa}
            B(i + 1) : = C(i + 1);
            (7) {Tarteko asertzioa}
            C(i + 1) : = lag;
    end if;
    (8) {Tarteko asertzioa}
    i : = i + 1;
end loop;
(9) {Bukaerako baldintza}
(10) {E espresioa}
```

44. (2010eko apirila #2) bitak(E(1..q)) eta biratuta(F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), H(1..r), (h_1, h_2, …, h_r), Q(1..r), pos) predikatuak eta A(1..n), B(1..n), C(1..n) eta zeroak eta batekoak bakarrik dituen D(1..n) bektoreak emanda, C(i) 0 bada A(i) → B(i) → C(i) → A(i) biraketa eta C(i) 1 bada A(i) → C(i) → B(i) → A(i) biraketa burutzen duen programa

a) E(1..q) bektoreko elementu denak 1 edo 0 direla adierazten duen **bitak(E(1..q))** predikatua definitu.

d) Jarraian zehazten dena adierazten duen **biratuta(F(1..r), (f_1, f_2, …, f_r), G(1..r), (g_1, g_2, …, g_r), H(1..r), (h_1, h_2, …, h_r), Q(1..r), pos)** predikatua definitu:

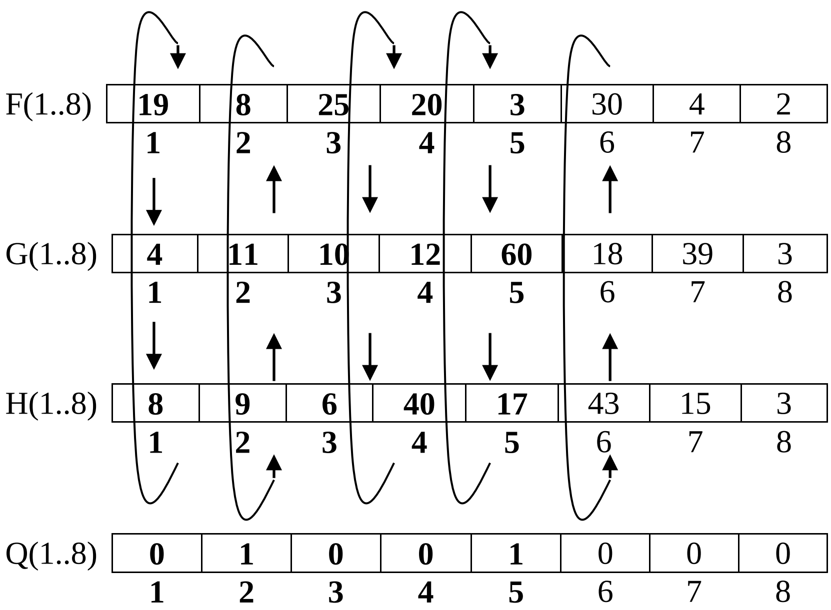
- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- Q(1..r) bektoreko elementu denak 1 edo 0 dira.
- 1 eta pos-en artean dauden k posizioetan (1 eta pos posizioak barne):
  - Q(k)-ren balioa 0 bada, orduan F(k) balioa h_k da, G(k) balioa f_k-ren berdina da eta H(k) balioa g_k-ren berdina da.
  - Q(k)-ren balioa 1 bada, orduan F(k) balioa g_k da, G(k) balioa h_k-ren berdina da eta H(k) balioa f_k-ren berdina da.

Predikatu hau definitzerakoan a) ataleko predikatua erabili behar da.

Adibidea:

Adibide honetako F(1..8), (4, 9, 10, 12, 17, 30, 4, 2), G(1..8), (7, 8, 6, 4, 3, 18, 39, 3), H(1..8), (19, 11, 25, 1, 3, 5, 15, 3) eta Q(1..8) bektoreek honako hau betetzen dute:

biratuta(F(1..8), (**4, 9, 10, 12, 17**, 30, 4, 2), G(1..8), (**7, 8, 6, 40, 3**, 18, 39, 3), H(1..8), (**19, 11, 25, 20, 60**, 43, 15, 3), Q(1..8), **5**):



c) Zenbaki osoz osatutako A(1..n), B(1..n), C(1..n) eta D(1..n) bektoreak emanda eta n ≥ 1 dela eta hasieran D(1..n) bektorean bakarrik batekoak eta zeroak daudela jakinda, D(1..n) bektoreko i posizio batean 0 dagoenean posizio horretako beste hiru bektoreetako elementuak A(i) → B(i) → C(i) → A(i) eran, hau da beherantz biratu arazten dituen eta D(1..n) bektoreko i posizio batean 1 dagoenean posizio horretako beste hiru bektoreetako elementuak A(i) → C(i) → B(i) → A(i) eran, hau da gorantz biratu arazten dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz.

```
(1) {Hasierako baldintza}
i : = 0;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≠ n loop
    (4) {Tarteko asertzioa}
    i : = i + 1;
    (5) {Tarteko asertzioa}
    if D(i) = 0
    then (6) {Tarteko asertzioa}
        lag : = C(i);
        (7) {Tarteko asertzioa}
        C(i) : = B(i);
        (8) {Tarteko asertzioa}
        B(i) : = A(i);
        (9) {Tarteko asertzioa}
        A(i) : = lag;
    then
        lag : = A(i);
        A(i) : = B(i);
        B(i) : = C(i);
        C(i) : = lag;
    end if;
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

45. (2010eko ekaina) bikoitia(x) eta mugibi(C(1..r), (c_1, c_2, …, c_r), pos) predikatuak eta A(1..n) bektorean jarraian dauden posizioak (1 eta 2, 3 eta 4, 5 eta 6, eta abar) binaka trukatzen dituen programa. A(1..n) bektoreko elementu kopurua bakoitia bada (n bakoitia), azkeneko elementua ez da lekuz mugituko.

a) x bikoitia dela adierazten duen **bikoitia(x)** predikatua definitu.

b) Aurreko ataleko predikatua erabiliz jarraian zehazten dena adierazten duen **mugibi(C(1..r), (c_1, c_2, …, c_r), pos)** predikatua definitu::

- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da.
- 1 eta pos-en artean dagoen k posizio bakoitzean (1 eta pos posizioak barne):
  - ➢ k bikoitia bada C(k) balioa c_{k-1} balioaren berdina da eta C(k – 1) balioa c_k balioaren berdina da.
  - ➢ k balioa pos balioaren berdina bada eta gainera bakoitia bada, orduan C(k) balioa c_k balioaren berdina da.

Adibidea:

Adibide honetako C(1..8) eta (4, 8, 7, 0, 9, 5, -2, 7) bektoreek honako hau betetzen dute:

mugibi(C(1..8), (**4, 8, 7, 0, 9**, 10, -2, 7), **5**)
mugibi (C(1..8), (**4, 8, 7, 0,** 9, 10, -2, 7), **4**)

C(1..8)	**8**	**4**	**0**	**7**	**9**	10	-2	7
	1	**2**	**3**	**4**	**5**	6	7	8

c) Zenbaki osoz osatutako A(1..n) taula emanda eta n ≥ 1 betetzen dela jakinda, jarraian dauden posizioak (1 eta 2, 3 eta 4, 5 eta 6, eta abar) binaka trukatzen dituen programa dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz. Kasu berezi bezala, A(1..n) bektoreko elementu kopurua bakoitia bada (n bakoitia bada), azkeneko elementua ez da lekuz mugituko.

```
(1) {Hasierako baldintza}
i : = 1;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i ≤ n loop
  (4) {Tarteko asertzioa}
  if i mod 2 = 0 then
                    (5) {Tarteko asertzioa}
                    lag : = A(i – 1) ;
                    (6) {Tarteko asertzioa}
                    A(i – 1) : = A(i);
                    (7) {Tarteko asertzioa}
                    A(i) : = lag;
                    (8) {Tarteko asertzioa}
  end if;
  (9) {Tarteko asertzioa}
  i : = i + 1;
end loop;
(10) {Bukaerako baldintza}
(11) {E espresioa}
```

46. (2010eko iraila) bikoitia(x), lauaniz(x) eta trukalau(D(1..r), (d_1, d_2, …, d_r), pos) predikatuak eta A(1..n) bektorean posizio bikoitietako elementuak (2 eta 4 posizioetakoak, 6 eta 8 posizioetakoak eta abar) trukatzen dituen programa

a) x zenbakia bikoitia dela adierazten duen **bikoitia(x)** izeneko **predikatua definitu**.
b) x zenbakia lauren anizkoitza dela adierazten duen **lauaniz(x)** izeneko **predikatua definitu.**
c) Jarraian zehazten dena adierazten duen

trukalau(D(1..r), (d_1, d_2, …, d_r), pos)

predikatua definitu:

- r balioa 1 baino handiagoa edo berdina da eta gainera 4ren anizkoitza da.
- pos balioa 0 baino handiagoa edo berdina eta r baino txikiagoa edo berdina da eta gainera lauren anizkoitza da.
- 1 eta pos-en artean (1 eta pos posizioak barne) dagoen edozein k posizio hartuta:
  - k balioa 4ren anizkoitza baldin bada, orduan D(k) balioa d_{k-2} balioaren berdina da eta D(k – 2) balioa d_k balioaren berdina da.
  - k balioa 4ren anizkoitza ez bada baina bikoitia bada, orduan D(k) balioa d_{k+2} balioaren berdina da eta D(k + 2) balioa d_k balioaren berdina da.
  - k balioa bikoitia ez bada, orduan D(k) balioa d_k balioaren berdina da.

Adibidea:

Adibide honetako D(1..12) eta (5, 9, 2, 10, 20, 18, -9, 17, 0, 3, 6, 1) bektoreek honako hau betetzen dute:

trukalau(D(1..12), (**5, <u>9</u>, 2, <u>10</u>, 20, <u>18</u>, -9, <u>17</u>**, 0, 3, 6, 1), **8**)

D(1..12)	**5**	**<u>10</u>**	**2**	**<u>9</u>**	**20**	**<u>17</u>**	**-9**	**<u>18</u>**	0	3	6	1
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	9	10	11	12

Honako hau ere egia da:

trukalau(D(1..12), (**5, <u>9</u>, 2, <u>10</u>,** 20, 18, -9, 17, 0, 3, 6, 1), **4**)

Baina <u>beste hiru hauek</u> ez dira betetzen, <u>false dira</u>:

trukalau(D(1..12), (5, 9, 2, 10, 20, 18, -9, 17, 0, 3, 6, 1), **9**)
Hau False da 9 ez delako 4ren anizkoitza.

trukalau(D(1..12), (5, 9, 2, 10, 20, 18, -9, 17, 0, 3, 6, 1), **6**)
Hau False da 6 ez delako 4ren anizkoitza.

trukalau(D(1..12), (5, 9, 2, 10, 20, 18, -9, 17, 0, 3, 6, 1), **12**)
Hau False da 10 eta 12 posizioetako elementuek ez dutelako betetzen predikatuaren definizioan zehaztutakoa.

d) A(1..n) bektorea emanda eta $n \geq 1$ dela eta gainera 4ren anizkoitza dela jakinda, posizio bikoitietako elementuak (2 eta 4 posizioetakoak, 6 eta 8 posizioetakoak eta abar) trukatzen dituen programa hau dokumentatu hor zehazten diren formulak emanez eta aurreko ataletan definitutako predikatuak erabiliz.

```
(1) {Hasierako baldintza}
i : = 4;
(2) {Tarteko asertzioa}
while (3) {Inbariantea} i <= n loop
    (4) {Tarteko asertzioa}
    lag : = A(i – 2);
    (5) {Tarteko asertzioa}
    A(i – 2) : = A(i);
    (6) {Tarteko asertzioa}
     A(i) : = lag;
    (7) {Tarteko asertzioa}
     i : = i + 4;
end loop;
(8) {Bukaerako baldintza}
(9) {E espresioa}
```