

AZPIPROGRAMAK ETA SEKUENTZIAK

Behean zehaztutako ariketetatik **gutxienez derrigorrezko ariketak** egin itzazu.

Oharrak:

- **Ariketen espezifikazioa, algoritmoa eta proba-kasuen diseinua eginda ekarri behar dira laborategira**, paperean edo bestelako euskarriren batean. Adaz inplementatuta ekartzea ere ez legoke gaizki, laborategian lana errazago egiteko.
- **Lehen 3 ariketako azpiprogramen proba-programak eGela-n dituzue eskuragarri.** Proba-programa horiek aukera ematen dute kasu desberdinak probatzeko. Proba-kasu egokiak diseinatzea garrantzitsua da egindako azpiprogramen zuzentasunaz ziur egoteko.
- **Egindako lana eGela plataformaren bidez entregatuko da laborategi-saioan bertan**, aldezturik behar bezala kodetuta, probatuta eta dokumentatuta. Ariketa bakoitzeko fitxategi bat igo behar da, iturburu-kodearekin (*.adb* fitxategia). Fitxategi horretan espezifikazioa eta proba-kasuen zerrendaketa ere sartu beharko da, enuntziatu honen bukaeran zehazten den adibidean bezala.

DERRIGORREZKO ARIKETAK

1. Honako azpiprograma inplementatu eta probatu (proba-programak eGela-tik hartu; horiek nagusiak dira, eta, beraz, horien exekutagarriak lortu behar dira):

```
function Telefonoa_Da (Zbkia : in Positive) return Boolean is
-- Aurrebaldintza: Zbkia (zenb. oso positiboa)
-- Postbaldintza: Eraitza True da, b.s.b. 999999999 >= Zbkia >= 600000000
-- eta Zbkia 9, 7 edo 6 digituarekin hasten bada
```

2. Honako azpiprograma inplementatu eta probatu (proba-programak eGela-tik hartu; horiek nagusiak dira, eta, beraz, horien exekutagarriak lortu behar dira):

```
function Igarren_Digitua (Zbkia, I : in Positive) return Natural is
-- Aurrebaldintza: Zbkia telefono-zenbakia da
-- 1 <= I <= 9
-- Postbaldintza: Eraitza Zbkia-ren I. digitua da (eskuinetik kontatuta)
```

3. Honako azpiprograma inplementatu eta probatu (proba-programak eGela-tik hartu; horiek nagusiak dira, eta, beraz, horien exekutagarriak lortu behar dira):

```
function Zenbat_Digitu (N : in Positive) return Natural is
-- Aurrebaldintza: N > 0
-- Postbaldintza: Eraitza N-ren digitu kopurua da
```

4. Zero zenbakiarekin amaitzen den zenbaki-sekuentzia bat teklatutik emanda, jakin nahi da zenbaki horietatik zenbat diren telefono finkoei dagozkienak (9 digituarekin hasten diren telefono-zenbakiak), zenbat telefono mugikorrei dagozkienak (6 edo 7 digituarekin hasten diren telefono-zenbakiak) eta zenbat ez diren telefono-zenbakiak. Espezifikatu, inplementatu eta probatu programa bat, hiru eraitza horiek idazteko: Telefonoak_Zenbatu (*telefonoak_zenbatu.adb*).

Egindako programak Telefonoa_Da eta Igarren_Digitua funtzioak erabiliko ditu.

5. Zero zenbakiarekin amaitzen den zenbaki-sekuentzia bat teklatutik emanda, jakin nahi da zenbaki horietatik telefono-zenbakia den aurrenekoa zein posiziotan dagoen sekuentzian. Sekuentzian telefono-zenbakirik ez baldin badago, eraitza 0 izango da. Espezifikatu, inplementatu eta probatu programa bat, eraitza hori lortzeko: Lehen_Telefonoaren_Pos_Kalkulatu (*lehen_telefonoaren_pos_kalkulatu.adb*).

6. Puntu karaktereaz amaitzen den karaktere-sekuentzia emanda 'w' karakterearen lehenengo bi agerpenen posizioak jakin nahi dira. Karaktere hori agertzen ez bada, edo behin bakarrik, orduan 0 0 lortuko da. Espezifikatu, inplementatu eta probatu programa bat, emaitza hori lortzeko: *Lehen_bi_w (lehen_bi_w.adb)*.
7. Puntu karaktereaz amaitzen den karaktere-sekuentzia emanda 'w' karakterearen lehenengo bi agerpenen posizioak jakin nahi dira, eta gainera karaktere guztien kopurua ere. Karaktere hori agertzen ez bada, edo behin bakarrik, orduan 0 0 eta karaktere guztien kopurua lortuko da. Espezifikatu, inplementatu eta probatu programa bat, emaitza hori lortzeko: *Lehen_bi_w_eta_kontatu (lehen_bi_w_eta_kontatu.adb)*.

HAUTAZKO ARIKETAK

8. Inplementa ezazu zenbaki baten i-garren digitua ordeztzeko prozedura bat (*igarren_digitua_ordeztu.adb*), honako zehaztapen honen arabera:

```
procedure Igarren_Digitua_Ordeztu (Zbkia : in out Positive;  
                                   I : in Positive;  
                                   Bal : in Natural) is  
  -- Aurrebaldintza: Zbkia telefono-zenbakia da  
  -- I da ordeztu nahi den digituaren posizioa (1 <= I <= 9)  
  -- Bal da digitu berria (0 <= Bal <= 9)  
  -- Postbaldintza: Zbkia-aren I. digitua (eskuinetik kontatuta) Bal da;  
  -- Zbkia-ren beste digitu guztiak ez dira aldatu.
```

Egizu, halaber, azpiprograma hori probatzeko proba-programa bat: *Proba_Igarren_Digitua_Ordeztu (proba_igarren_digitua_ordeztu.adb)*.

9. Zero zenbakiarekin amaitzen den zenbaki-sekuentzia bat emanda, espezifikatu eta inplementatu programa bat sekuentziako zenbaki bakoitzaren digitu kopurua idazteko: *Zenbakien_Digitu_Kopuruak_Idatzi (zenbakien_digitu_kopuruak_idatzi.adb)*. Zenbakien digitu kopurua sekuentzia ordena hertsiki gorakorrean dagoen bitartean soilik idatziko da; hala ez denean, programa gelditu egingo da.

Adibidez:

- Sarrera-sekuentzia: <3 16 333333333 411111111 93 0>
- Emaitza: <1 2 9 9> (93 zenbakia irakurtzean, programa geratu egiten da)

Erabil itzazu beharrezkoak diren azpiprogramak.

Ariketetako programek honako eredua jarraitu beharko dute, hau da, espezifikazioa txertatu egin beharko da, eta gainera proba-kasuen zerrenda ere hasieran gehitu beharko da

```
-- Proba-kasuak
-- sarrera: ... --> irteera:
-- sarrera: ... --> irteera:
-- sarrera: ... --> irteera:
-- ...:
-- *****

WITH idatzi_katea, irakurri_oso;
PROCEDURE prozedura IS
    -- Espezifikazioa
    -- Aurrebaldintza
        -- BETE
    -- Postbaldintza
        -- BETE

    -- aldagaien erazagupena
    ...
BEGIN

    -- Kodea idatzi

END prozedura;
```