

## AZPIPROGRAMAK ETA BEKTOREAK

### Oharrak

- **Ariketen algoritmoak eginda ekarri behar dira laborategira**, paperean edo bestelako euskarriren batean. Oraingoan, bai espezifikazioak eta bai proba-kasuak eginda ematen zaizkizu.
- **Ariketak egiteko zenbait programa lagungarri eGela-n dituzue eskuragarri**. Lehen ariketarako proba-programak aukera ematen du azpiprograma guztiak probatzeko, proba-kasu desberdinekin. Bigarren ariketarako proba-programa zuk zeuk gara dezakezu.
  - o Ematen zaizkizun programak hutsunez beteta daude. Hiru izartxo (**\*\*\***) adierazita daude zuk zerbait idatzi beharreko guneak.
- Lan egiteko modua. Azpiprograma bakoitzeko, egin:
  - o diseinatu algoritmoa;
  - o kodetu prozedura edo funtzio gisa, proba-programa orokorrean bertan, dagokion lekuan txertatuz;
  - o idatzi azpiprograma horri dagozkion deiak proba-programan (zenbait kasutan, aldagaien bat ere erazagutu beharko duzu);
  - o konpilatu proba-programa, eta probatu exekutagarria (kontuan izan parentesi artean dauden balioak, horiek baitira lortu behar diren emaitzak).
- **Egindako lana eGela plataformaren bidez entregatuko da laborategi-saioan bertan**, aldez aurretik behar bezala kodetuta eta probatuta. **Zehazki, bi fitxategi hauek igo behar dira eGela-ra:** *Oso\_Bektore\_Proba\_Orokorra.adb* eta *Irakasgaien\_Estatistikak\_Lortu.adb*.

### ARIKETAK

1. Honako azpiprograma hauek inplementatu eta probatu, eginda ematen zaizun *Oso\_Bektore\_Proba\_Orokorra* proba-programaz baliatuz (programa hori izango da nagusia, eta, beraz, horren exekutagarria lortu eta probatu behar da) :

*Oso\_Bektore* datu-mota eta *N* konstantea emanda:

```
N : constant Positive := 10;  
type Oso_Bektore is array (1 .. N) of Integer;
```

eta *Lehena\_Da* azpiprograma:

```
function Lehena_Da (X : in Positive) return Boolean is  
  --Aurrebaldintza: X > 0, aztertu beharreko zenbakia  
  --Postbaldintza: emaitza = True, X zenbaki lehena bada;  
  --  
  emaitza = False, bestela.
```

- A) **procedure Lehen\_Zenbaki\_Lehenaren\_Pos\_Eta\_Balioa\_Lortu**  
(B : in Oso\_Bektore; Pos : out Natural; Balioa : out Integer);  
-- Aurrebaldintza:  
-- B (oso-bektorea), non edozein I-rentzat, 1 <= I <= 10, B (I) > 0.  
-- Postbaldintza:  
-- B-n zenbaki lehenik baldin badago:

```
--      Pos da B-ko lehen zenbaki lehenaren posizioa;
--      Balioa da B-ko lehen zenbaki lehena;
--      bestela:
--      Pos = 0.

B) function Denak_Zenbaki_Lehenak (B : in Oso_Bektore) return Boolean;
-- Aurrebaldintza:
--      B (oso-bektorea), non edozein I-rentzat, 1 <= I <= 10, B (I) > 0.
-- Postbaldintza:
--      emaitza = True, B-ko zenbaki guztiak lehenak baldin badira;
--      emaitza = False, bestela.

C) procedure Atzekoz_Aurrera_Jarri (B : in out Oso_Bektore);
-- Aurrebaldintza:
--      B, alderantzikatu beharreko bektorea.
-- Postbaldintza:
--      B, non osagaiak tokiz aldatu baitira, eta orain atzekoz
--      aurrera daude: lehena azkenarekin trukatu da, bigarrena
--      azkenaurrekoarekin, eta horrela hurrenez hurren.

D) procedure Igarrenean_Txertatu_Desplazatz
   (B: in out Oso_Bektore; Pos : in Positive; X: in Integer);
-- Aurrebaldintza:
--      B bektorea, non txertatu nahi den X Pos posizioan;
--      Pos posizioa, non B'First <= Pos <= B'Last;
--      X zenbakia, B-n txertatu beharrekoa.
-- Postbaldintza:
--      B, non B (Pos) = X, eta Pos-etik aurrerako guztiak posizio
--      bat eskuinera desplazatu diren; B-ko azken posizioan
--      zegoen elementua galdu egin da.
```

2. Honako azpiprograma hauek eta Irakasgaien\_Estatistikak\_Lortu programa inplementatu eta probatu. Azken horren eskeletoa ematen zaizu (programa hori izango da nagusia, eta, beraz, horren exekutagarria lortu eta probatu behar da (horretarako, nahi izanez gero, lehen ariketarako emandako proba programaren antzeko bat garatuz)) :

Irakasgaia\_Bektore eta Portzentajea\_Bektore datu-motak emanda (irakasgaiak datu-mota enumeratua da):

```
type irakasgaiak is (AM, KTO, MD, OP, SDDO);
type irakasgaia_Bektore is array (irakasgaiak) of Integer;
type portzentajea_Bektore is array (irakasgaiak) of Float;
```

```
A) procedure Datuak_Kargatu (B : in out Irakasgaia_Bektore);
-- Aurrebaldintza:
--      B=(b1,b2,b3,b4,b5) eta b_i balio osoa
--      F=<x1,x2,x3,x4,x5> eta x_i balio osoa
-- Postbaldintza:
--      B-ren osagaietan sarrera estandarretik lortutako balioak
--      esleitu dira, hots, B=(x1,x2,x3,x4,x5)

B) procedure Portzentajea_Kalkulatu
   (A, B : in Irakasgaia_Bektore, C: out Portzentajea_Bektore);
-- Aurrebaldintza:
--      A, B (irakasgaia-bektorea), beren osagai guztiak > 0.
-- Postbaldintza:
--      C-ren i osagaian (non i in (AM, KTO, MD, OP, SDDO)) balioa honakoa
--      da: (A(i)/B(i))*100
```

```
C) procedure Datuak_Pantailaratu (B : in Portzentajea_Bektore);  
-- Aurrebaldintza:  
-- B (portzentajea-bektorea).  
-- Postbaldintza:  
-- B-ren osagaietako balioak pantailan idazten dira, osagaiaren  
-- indizea eta bere balioa, era honetara: i: balioa (non i in (AM,  
-- KTO, MD, OP, SDDO))  
  
D) procedure Irakasgaien_Estatistikak_Lortu;  
-- Aurrebaldintza:  
-- F=<m1, m2, m3, m4, m5>, <r1, r2, r3, r4, r5>, <a1, a2, a3, a4, a5>,  
-- hiru zenbaki sekuentzia, non guztiak positiboak (/=0) eta  
-- (AM, KTO, MD, OP, SDDO)) bost irakasgaiei dagozkien datuak diren  
-- m, r eta a zenbakiek matrikula kopurua, azterketara aurkeztutako  
-- kopurua, eta gaingiditutakoen kopurua adierazten duten, hurrenez  
-- hurren  
-- Postbaldintza:  
-- G=<i1, b1,..., i5, b5, i1, c1,..., i5, c5, i1, d1,..., d5>  
-- non i irakasgaiaren kodea den (hauetako bat: AM, KTO, MD, OP, SDDO),  
-- b, c eta d honako balioak diren, hurrenez hurren:  
-- Ebaluazio tasa: matrikulatuen artetik aurkeztutakoen portzentajea  
-- Arrakasta tasa: aurkeztutakoen artetik gaingiditutakoen portzentajea  
-- Errendimendu tasa: matrikulatuen artetik gaingiditutakoen  
-- portzentajea
```