

## *Oinarrizko Prog. - 8. laborategia.*

### *Azterketetako ariketak*

#### **OHARRAK:**

1. Proposatzen den ariketa bateko txantiloiren bat edo probetarako programa faltaz gero, zuek sortu eta eraiki beharko duzue **hutsetik**. Gainera, txantiloietan agertzen diren proba kasuak agertzeak ez du esan nahi proba guztiak daudenik. Zuek gehitu beharko dituzue **falta direnak**.
2. Laborategi honetan ez da txostenik entregatu behar. eGelara jatorri fitxategiak (.adb eta .py) fitxategi batean konprimatuta (.zip) igo beharko dituzue. Gogoratu fitxategiaren orain arteko izendatze arauak errespetatzeaz (adibidez, Jsaez\_Petxeberria\_lab8.zip).
3. Ariketak zuzenak direla auresuposatzen da, hau da, konpilazio errorerik ez dutela eta ondo funtzionatzen dutela. Horrela, ariketa ondo egoteak ez du punturik ematen, baina akatsak egoteak penalizatzen du. Behin soluzioa zuzena dela, honakoa da ebaluatuko dena (eta puntuatuko duena):
  - a) **Proba kasuak:** Proba kasu guztiak gehitu dira, orokorretatik kritikoenetara? Batzuk ematen dira, baina beste asko falta dira.
  - b) **Eraginkortasuna:** Beharrezkoak direnean “salatariak” erabiltzen dira begizta osoa ez korritzeko? Beharrezkoak ez diren baldintzak daude? Beharrezko parametro eta aldagaiak BAKARRIK definitzen dira?
  - c) **Argitasuna:** Kodea tabulatua dago? Aldagaien izenek kodea ulertzen laguntzen dute? Funtzioaren bukaeran return BAKARRA dago? ....

# 1. Ariketa: Ordenatu bikoiti eta bakoitiak

Ariketa hau **Pythonez** bakarrik ebatzi behar da.

**Txantiloia:** ordenatu\_bikoiti\_bakoitiak.py

Osokoen\_bektore bat sarrera gisa hartuta, bektore horretan agertzen diren elementuak berrantolatzen dituen azpiprograma egitea eskatzen da. Horretarako, bikoitiak bektorearen hasieran multzokatu behar dira eta bakoitiak bektorearen bukaeran.

Adibidez, hurrengo bektorea izanda:

10	3	13	4	6	3	5	2	9	7	8	18	12	2
----	---	----	---	---	---	---	---	---	---	---	----	----	---

Emaitza izan zitekeen:

10	4	6	2	8	18	12	2	3	13	3	5	9	7
----	---	---	---	---	----	----	---	---	----	---	---	---	---

## OHARRAK:

- Emaizako bikoitien arteko ordenak ez du axola, bakoitiena ere ez. Garrantzitsua den gauza bakarra, bektorearen **HASIERAN** (edozein ordenan) bikoitiak agertzea eta bukaeran (edozein ordenan) bakoitiak.
- Gogoratu eraginkortasuna garrantzitsua dela, eta ondorioz, bektorea **BEHIN** bakarrik zeharkatu beharko da. Horretarako, bi indize erabiltzea (bat bektorearen hasieran jarritz eta bestea bukaeran) erabiltzea gomendatzen da.

# 2. Ariketa: Hauteskundeak

Ariketa hau **ADAz** bakarrik ebatzi behar da.

**Fitxategiak** (ADA): *motak.ads*, *maximoak\_esleitu.adb*, *alderdien\_kozianteak\_kalkulatu.adb*, *koziante\_guztiak\_kalkulatu.adb*, *botoak\_banatu.adb*, *aulkiak\_banatu.adb*, *aulkiak\_idatzi.adb*, *alderdien\_botoen\_lista\_idatzi.adb*, *alderdien\_kozianteen\_taula\_ida-tzi.adb*, *dhont\_taula\_idatzi.adb*, *ehuneko\_3\_baino\_gehiagoko\_adlerdiak\_lortu.adb*, *aulkiak\_banatu.adb* (ez dira moldatu behar).

**Babes fitxategiak** (ADA): *sortu\_1kasuko\_alderdien\_botoen\_lista.adb*, *proba\_maximoak\_esleitu.adb*, *proba\_alderdien\_kozianteak\_kalkulatu.adb*, *proba\_koziante\_guztiak\_kalkulatu.adb*, *proba\_botoak\_banatu.adb*, *proba\_ehuneko\_3\_baino\_gehiagoko\_alderdiak\_lortu.adb* eta *proba\_aulkiak\_banatu.adb* (moldatu daitezke beharrezkoa iritziz gero)

**Txantiloia** (ADA): *aulkiak\_kalkulatu.adb* eta *proba\_aulkiak\_kalkulatu.adb*

*Motak.ads* izeneko fitxategian, autonomia-erkidegoko hauteskundearen emaitzak gordetzeko prestatua dagoen datu-egitura aurki dezakezue. Ondo aztertu, horiek baitira ariketa honetan zehar erabili beharko dituzuen egiturak.

Ezarritako konstanteak

Max_kop_alderdi_politiko: <b>constant</b> integer :=10; Kop_aulki: <b>constant</b> integer := 25;
--

## Tarteak

```
subtype T_tarte_alderdi is Integer range 0..Max_kop_alderdi_politiko;
subtype T_izena is String(1..15);
```

## Alderdi politikoen lista beraien botoekin (T\_Lista\_alderdiak\_botoak)

```
type T_info_alderdien_botoak is record
    Izena : T_izena;
    Botoak : Natural;
end record;

type T_alderdien_taula is array (1..Max_kop_alderdi_politiko) of
T_info_alderdien_botoak;

type T_Lista_alderdien_botoak is record
    Alderdi_kop: T_tarte_alderdi;
    Alderdien_taula: T_alderdien_taula;
end record;
```

## Alderdi politikoen lista berauen aulkiekin (T\_lista\_aulkiak)

```
type T_info_alderdi_aulkiak is record
    Izena : T_izena;
    Aulki_kop: Natural;
end record;

type T_taula_aulkiak is array (1..Max_kop_alderdi_politiko) of T_info_alderdi_aulkiak;

type T_lista_aulkiak is record
    Kop_alderdi : T_tarte_alderdi;
    Taula_aulkiak : T_taula_aulkiak;
end record;
```

Honakoa eskatzen da:

Hauteskundetara aurkeztu diren alderdi politiko bakoitzari dagozkion aulki kopurua kalkulatzeko. Azpiprogramaren burua honakoa da:

```
procedure aulkiak_kalkulatu (
    LPV_Bizkaia, LPV_Araba, LPV_Gipuzkoa : in T_Lista_alderdiak_botoak;
    Emaizta: out T_lista_aulkiak)
-- Aurre: sarrerako 3 aldagaiek, probintzia bakoitzeko hauteskundearen emaitzak dituzte
-- Post: Emaizta, alderdi politiko bakoitzak lortu dituen aulki kopurua izango da.
    Probintzi bakoitzeko 25 aulki esleitu dira, non 25 aulkiak alderdi
    politikoen artean banatu diren.
```

Adibidez, ondorengo hauteskundearen emaitzak kontuan izanda:

Bizkaia		Araba		Gipuzkoa	
PNV/EA	264774	PNV/EA	51601	PNV/EA	147498
PSE-PSOE	151347	PP	43765	EHAK	78088
PP	113867	PSE-PSOE	43765	PP	70577
EHAK	65431	EHAK	14180	PSE-PSOE	51163
EB-IU	36258	EB-IU	8395	EB-IU	20278
ARALAR	10187	ARALAR	2541	ARALAR	15273

Aulkien asignazioa D'Hont metodoaren bitartez eginez gero, probintzi bakoitzeko aulkien asignazioa ondorengo izango litzateke:

Bizkaia		Araba		Gipuzkoa	
PNV/EA	11	PNV/EA	8	PNV/EA	10
PSE-PSOE	6	PP	7	EHAK	5
PP	5	PSE-PSOE	7	PP	5
EHAK	2	EHAK	2	PSE-PSOE	3
EB-IU	1	EB-IU	1	EB-IU	1
				ARALAR	1

Non, guztiak kontuan izanda, alderdi politiko bakoitzak lortutako aulki kopurua honakoa izango litzateke:

Aulkiak guztira	
PNV/EA	29
PSE-PSOE	16
PP	17
EHAK	9
EB-IU	3
ARALAR	1

Oharrak:

- Emaitzak ez du zertan ordenaturik egon behar.
- Ariketa hau ebatzi ahal izateko, *aulkiak\_banatu* azpiprogramaz gain, honek beharrezkoak dituen beste azpiprograma batzuk ere banatzen zaizkizue (ez dituzue garatu behar). *Aulkiak\_banatu* azpiprogramaren betebeharra D'Hont metodoari deitzea da, non azken honen helburua, probintzi bakoitzean lortutako botoetan oinarrituz, 25 aulkien banaketa burutzea izango den.

### 3. Ariketa: Tutoretzetara joatea

Ariketa hau **ADAz** bakarrik ebatzi behar da.

**Fitxategiak** (ADA): *datuak.ads* (ez da berridatzi behar)

**Txantiloiak** (ADA): *nan\_posizioa.adb*, *ezabatu\_errepikatuak.adb*, *ordenatu.adb*, *ikasle\_eta\_tutoretza\_totalak\_lortu.adb* eta *ikasle\_eta\_tutoretza\_totalak\_lortu\_proba.adb*

Ikasleen tutoretzetara joateak kudeatzeko honako datu egiturak dituzue. Egutegiak, lauhilekoaren lanegun bakoitzeko tutoretzetara joateak gordetzen ditu. Lanegun bakoitzeko, data eta tutoretzetara joandako ikasleen NAN zenbakiak gordetzen dira.

```
IK: constant Integer := 60; -- Ikasle kopuru maximoa

subtype Ikasle_Tartea is Integer range 1..IK;
type T_NANak is array (Ikasle_Tartea) of Positive;
type Info_eguna is record
    Data: String(1..10); -- "uuuu/hh/ee" formatuan
    Ikasle_kop: Natural;
    NANak: T_NANak;
end record;

LEK: constant Integer := 200; -- Lanegun kopuru maximoa
subtype Lanegun_Tartea is Integer range 1..LEK;
type T_egunak is array (Lanegun_Tartea) of Info_eguna;
type Info_egutegia is record
    Zenbat_egun: Natural;
    Egunak: T_egunak;
end record;
```

Horrez gain, emaitzak gordetzen joateko, honako datu egiturak eskaintzen zaizkizue:

```
type Info_tutoretzak is record
    NAN: Positive;
    Tutoretzak_Guztira: Natural;
end record;

type T_tutoretzak is array(Ikasle_Tartea) of Info_tutoretzak;

type Ikasleen_tutoretza_zerrenda is record
    Zenbat: Natural;
    Tutoretzak: T_tutoretzak;
end record;
```

Azpiprograma bat idatzi behar da, non, ikasle bat zenbat aldiz joan den tutoretzatarara kalkulatzen duena. Zerrenda hau, ikasleak joandako tutoretza kopuru totalaren arabera ordenatuta agertuko da, handitik txikienera. Azpiprogramaren burukoa honakoa da:

```
procedure ikasle_eta_tutoretza_totalak_lortu (egutegia: in Info_egutegia;
                                              Zerrenda: out Ikasleen_tutoretza_zerrenda)
---Aurre: Egutegia-k tutoretzetara joateen informazioa dauka (lauhilabeteko lanegun
---        bakoitzeko, data eta tutoretzetara joandakoen NAN zenbakiak)
---Post: Zerrenda ORDENATUTA dago, ikasleak joandako tutoretza kopuru totalaren arabera
---       Ikasle bat egun berdinean behin baino gehiagotan joan bada tutoretzetara, behin
---       bakarrik kontatuko zaio tutoretzetara joatea egun horretako guztien partez.
```

Adibidez, suposatu egutegiak honako informazioa daukala:

- \* 2018/06/18 egunean honako ikasleak joan dira tutoretzetara: 11, 22, 33, 11, 22, 33, 44 eta 55.
- \* 2018/06/19 egunean honako ikasleak joan dira tutoretzetara: 55, 22 eta 55 .
- \* 2018/06/20 ez da ikaslerik tutoretzetara etorri.
- \* 2018/06/21 egunean honako ikasleak joan dira tutoretzetara: 22 eta 33.
- \* 2018/06/22 egunean honako ikasleak joan dira tutoretzetara: 44, 33, 22, 44 eta 11.

Irteera: Kasu honetarako, irteeran jasoko den informazioa (Ikasleen\_tutoretza\_zerrenda motakoa) honakoa beharko luke (azken hiru NAN zenbakiak kopuru berdina dutenez, euren arteko ordenak ez du garrantzirik):

NAN 22-ren joateak: 4 aldiz.

NAN 33-ren joateak: 3 aldiz.

NAN 55-ren joateak: 2 aldiz.

NAN 44-ren joateak: 2 aldiz.

NAN 11-ren joateak: 2 aldiz.