Oinarrizko Prog. - 6. laborategia. Zerrendak eta matrizeak

Izena:_____**Data**:_____

OHARRA:

- Proposatzen den ariketa bateko txantiloiren bat faltaz gero, zuek sortu eta eraiki beharko duzue hutsetik. Gainera, txantiloietan agertzen diren proba kasuez gain, agian gehiago ere burutu beharko duzue.
- Datu moten ezaugarriak, **zerrendak.ads** eta **bektoreak_eta_matrizeak.ads** fitxategietan topatuko dituzue. Gogoratu, taulek/arrayek/bektoreek ezin dutela luzera aldakorra izan. Baina taula bat erregistro baten barruan sartzen badugu nun erregistroak 2 eremu izango dituen, alegia bektorea, eta luzera markatzeko erabiliko dugun *integer* eremua, orduan *"luzera aldakorreko"* taula bat simulatu dezakegu. Honela, elementu berri bat sartu nahi dugun bakoitzean luzerari bat gehituko diogu eta elementu bat kendu nahi dugun bakoitzean luzerari bat kenduko diogu.
- Litekeena da proba prometan proba kasu gehiago gehitu behar izatea.
- Agertzen ez den azpiprograma edo programa nagusi bat beharrez gero, zuek sortu beharko duzue. Ariketa batzuk **ADAz egin behar dira (alegia, 1., 2., 3., 4., 5., 6., 7.,** eta **9.) gainontzekoak PYTHON** lengoaiaz egin behar dira.

1. **Zenbakia bilatu osoko-bektore desordenatuan:** N elementu dituen osoko-bektore **desordenatu** batean zenbaki bat bilatzeko algoritmoa espezifikatu eta egin. Irteera, zenbakia zerrendan dagoen ala ez izango da. Horretarako erabili proba_bektore_desordenatuan_bai_ez.adb eta bektore_desordenatuan_bai_ez.adb.

2. **Zenbakia bilatu osoko-bektore ordenatuan:** N elementu dituen osoko-bektore **ordenatu** batean (handitik txikira) zenbaki bat bilatzeko algoritmoa espezifikatu eta egin. Irteera, zenbakia, bektorean dagoen ala ez izango da, TRUE baiezkoan, FALSE bestela. Horretarako erabili proba_bektore_ordenatuan_bai_ez.adb eta bektore_ordenatuan_bai_ez.adb.

3. **Zenbakia bilatu bektore batean eta bere posizioa itzuli:** N elementu dituen osoko-bektore bat eta zenbaki bat emanda, zenbaki hori zerrendan balego, zein posiziotan dagoen itzuliko duen algoritmoa espezifikatu eta egin**.** Zenbakia hainbat

aldiz agertzen bada, nahi duzuen posizioa itzuli baina ez balego, itzuli -1 posizioa**.** Horretarako erabili posizioan_dago.adb eta proba_posizioan_dago.adb

4. **Posizio bat eskuinera mugitu** N elementuz osatutako osoko-bektore bat emanda, osagai guztiak posizio bat eskuinera mugitzen dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu. Erabili eskuinera_mugitu.adb, eta proba_eskuinera_mugitu.adb

Sarrera:

1	2	3	4	5	6	7	8	9	10
34	67	45	23	78	12	40	55	24	89

Irteera:

1	2	3	4	5	6	7	8	9	10
89	34	67	45	23	78	12	40	55	24

5. **Eliminatu elementu bat zerrenda desordenatu batean.** N elementuz osatutako osoko-bektore bat emanda, non behar bada bektorea ez den guztiz beteta egongo (beraz erregistro baten barruan egonda non erregistroak bektorez gain kopuru eremu bat ere izango duen). Eliminatu lehenengo elementua. Bektorearen elementuak ez daude ordenatuta. Horretarako, erabili eliminatu_lehenengo_elementua.adb, idatzi_lista.adb eta proba_eliminatu_lehenengo_elementua.adb

6. **Eliminatu elementu bat zerrenda ordenatu batean.** N elementuz osatutako osoko-bektore bat emanda, non behar bada bektorea ez den guztiz beteta egongo (beraz erregistro baten barruan egonda non erregistroak bektorez gain kopuru eremu bat ere izango duen). Eliminatu lehenengo elementua. Bektorearen elementuak ordenatuta daude txikitik handira. Horretarako, erabili eliminatu_lehenengo_elementua_ordenatuta.adb, idatzi_lista.adb eta proba_eliminatu_lehenengo_elementua_ordenatuta.adb

7. **Txertatu zenbaki bat zerrendako *Pos*garren posizioan.** Beteta ez dagoen zerrenda batean, txertatu ezazu zenbakia *pos* posizioan eta gainontzeko zenbakiak desplazatu itzazu eskuinera. Horretarako, erabili itzazu txertatu_zenbakia_pos.adb eta proba_txertatu_zenbakia_pos.adb

8. **Bilaketa matrize batean.** N x M osokoen matrize bat hartuta eta zenbaki bat, zenbakia matrize horretan dagoen ala ez esango digun azpiprograma idatzi. Zenbakia matrizean balego zenbakiaren posizioa ere (hau da X,Y koordenatuak) bueltatu beharko digu. Zenbakia errepikatuta balego, itzuli edozein posizio. Horretarako erabili matrizean_badago.adb eta proba_matrizean_badago.adb

9. **Matrize baten maximoa**. N x M osokoen matrize bat hartuta, matrize horretan elementurik handiena zein den eta bere posizioa kalkulatzen duen azpiprograma idatzi. Zenbakia errepikaturik balego, aurreneko posizioa itzuli. Erabili proba_maximoa_kalkulatu.adb, maximoa_kalkulatu.adb probatzeko

10. **Bektoreko ordenazioa, Txertaketa deritzon algoritmoa jarraituz**. http://www.youtube.com/watch?v=gTxFxgvZmQs&feature=related N elementu dituen osoko-lista bat emanda, bektoreko osagaiak Txertaketa deritzon algoritmoa

jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
def ordenatu_txertatuz (L1):
--pre:
--post: L1en elementuak agertuko dira baina goranzko ordenan ordenaturik
```

OHARRA: Bektorea zearkatzen hasiko gara elementuz elementu. *i*-garren iterazioaren hasieran, *i*-1 elemtuak ordenatuta egongo dira; *i+1*-garren iterazioan egungo elementua hartuko da eta aurreko elementuekin elementuz elementu konparatuko da (alegia, jada ordenatuta dauden elementuekin) topatzeko zein litzateke egungo elementuaren posizio egokia:

Sarrera:

1	2	3	4	5	6	7	8	9	10
2	4	7	14	25	12	11	55	5	6

1. gorde auxiliar batetan 12
2. bilatu 12-a txertatzeko posizio egokia
   12-a txertatzeko posizio egokia 4. litzateke.
3. desplazatu eskuinera 4. posiziotik 6.ra dauden elementuak

Sarrera:

1	2	3	4	5	6	7	8	9	10
2	4	7	14	14	25	11	55	5	6

4. txertatu 4. posizioan auxiliarrean gordetako 12-a

Erabilgarriak izan daitezkeen azpiprogramak:

* egungo elementuaren posizioa eta bektorea emanda esan ze posizioan txertatu beharko litzatekeen egungo elementua (aurreko kasuan 4. posizioan)

* Posizio bat emanda eta muga bat, posizio horretatik aurrera muga arte elementu guztiak eskumara desplazatu “hutsune” bat utziz.

Horretarako, erabili itzazu txertatzeko_posizioa_topatu, ordenatu_txertatuz, proba_ordenatu_txertatuz, eskuinera_mugitu.

11. Hautaketa deritzon algoritmoa jarraituz, bektore bat ordenatu: http://www.youtube.com/watch?v=boOwArDShLU

N elementu dituen osoko-lista bat emanda, bektoreko osagaiak hautaketa deritzon algoritmoa jarraituz ordenatuko dituen algoritmoa espezifikatu eta egin. Azpiprograma modura inplementatu.

```
procedure Ordenazioa_hautaketaz(L: in out osoko-lista);
-- pre:
-- post: Lren elementuak goranzko ordenan ordenaturik
```

Sarrerako bektorea(6 elementu)::

1	2	3	4	5	6	7	8	9	10
23	25	27	21	29	26	34	32	30	31

1. eta azkeneko posizioen artean dauden elementuetatik txikiena 21 da eta 4. posizioan dago, beraz ordezkatzen dut txikiena lehenengo posizioan dagoen elementuarekin.

1	2	3	4	5	6	7	8	9	10
21	25	27	23	29	26	34	32	30	31

Orain 2 eta 4. posizio artean dauden elementurik txikiena bilatzen dut, 23 eta 4. posizioan dago, Beraz ordezkatu beharko nuke 4. posizioan dagoen elementua (23) 2. posizioan dagoen elementuarekin (25) :

1	2	3	4	5	6	7	8	9	10
21	23	27	25	29	26	34	32	30	31

Orain 3 eta 4. posizio artean dauden elementurik txikiena bilatzen dut, 25 eta 4. posizioan dago, Beraz ordezkatu beharko nuke 4. posizioan dagoen elementua (25) 3. posizioan dagoen elementuarekin (27) :

1	2	3	4	5	6	7	8	9	10
21	23	25	27	29	26	34	32	30	31

Orain 4 eta 6. posizio artean dauden elementurik txikiena bilatzen dut, 26 eta 6. posizioan dago, Beraz ordezkatu beharko nuke 6. posizioan dagoen elementua (26) 4. posizioan dagoen elementuarekin (27) :

1	2	3	4	5	6	7	8	9	10
21	23	25	26	29	27	34	32	30	31

Orain 5 eta 6. posizio artean dauden elementurik txikiena bilatzen dut, 27 eta 6. posizioan dago, Beraz ordezkatu beharko nuke 6. posizioan dagoen elementua (27) 5. posizioan dagoen elementuarekin (27) :

1	2	3	4	5	6	7	8	9	10
21	23	25	26	27	29	34	32	30	31

Prozesu hau behin eta berriro egikarituko da bektore guztia zeharkatu arte.

Erabilgarri izan daitezkeen azpiprogramak:

* Horretarako, erabili minimoaren_posizioa_topatu, ordenatu_hautatuz, elkar_trukatu eta proba_ordenatu_hautatuz

12. Burbuila metodoa jarraituz bektore bat ordenatu. Begiratu
http://www.youtube.com/watch?v=1JvYAXT_064&feature=related

Algoritmoa honakoa izan daiteke:

Sarrerako zerrenda:

7	5	1	2	34	32	30	31	28	12

```
ADAz honela izango litzateke
num_pasatakoak:=1;
loop exit when num_pasatakoak > num_elementuak-1;
   i:=1;
   loop exit when i> num_elementuak-1;    --5. Elementua ez atzitzeko, ez baitago
        if t(i) > t(i+1) then ---aldatu
             aux:=t(i);
             t(i):=t(i+1);
             t(i+1):=aux;
        end if;
        i:=i+1;
   end loop;
num_pasatakoak:= num_pasatakoak+1;
end loop;
```

Aurreneko iterazio ostean ....

5	1	2	7	32	30	31	28	12	34

Bigarren iterazio ostean ....

1	2	5	7	30	31	28	12	32	34

Oharra: aldatu aurreko algoritmoa eraginkorrago egiteko. Adibidez, boolear bat erabili dezakezu iterazio batean ezer ez bada aldatzen algoritmoak amaitu dezan bektorea ordenatuta egongo baita, horrela, eraginkorragoa izango da.

Adibidez,

5	1	2	7	12	28	31	32	33	34

Lehenengo buelta eman ostean emaitza hau litzateke:

1	2	5	7	12	28	31	32	33	34

Eta beraz 2. bueltan ez litzateke ezer aldatuko ordenatuta dagoelako jada. Aldaketarik ez dela gertatu adierazteko boolear bat erabiliko dugu.

bektore_eta_matrizeak.ads

```
package bektore_eta_matrizeak is

type Osoko_bektorea is array (Integer range <>) of Integer;
type Errealen_bektorea is array (Integer range <>) of Float;
type Boolear_bektorea is array (Integer range <>) of Boolean;
type Karaktere_bektorea is array (Integer range <>) of Character;

type Osoko_matrizea is array (Integer range <>, Integer range <>) of Integer;
type Errealen_matrizea is array (Integer range <>, Integer range <>) of Float;
type Boolear_matrizea is array (Integer range <>, Integer range <>) of Boolean;
type Karaktere_bektorea is array (Integer range <>, Integer range <>) of
Character;
end Bektore_eta_matrizeak;
```

zerrendak.ads

```
with bektore_eta_matrizeak; use bektore_eta_matrizeak;

package zerrendak is
Max_Elem: constant Integer := 10;

type osoko-lista is record
        Zenbakiak: Osoko_bektorea (1.. Max_Elem);
        Kont: Integer;
end record;
end Defs_Lab6;
```