Bektore/Arraiak, eta matrizeak

Sarrera

- Algoritmikan sekuentziak ikusi genituen, arraiak sekuentzien baliokideak dira baina ADAz (edo hobeto esanda programazio lengoiaetan)

- Gogoratu dezagun zergatik ziren baliagarriak arraiak.

Motibazioa

- Ataza batzuen ebazpena burutzeko orain arte ADAz ikusi ditugun datu motak ez dira lagungarriak (integer, float, boolean). Adibidez

  1) Ordenatu txikitik handira hainbat zenbaki

  2) Edo, puntuz bukatzen den karaktere sekuentzia bat atzetik aurrera inprimatu

Abantailak

- Arraien bitartez, izen baten menpe gorde eta manipula ditzakegu mota berdineko hainbat elementu

- Adibidez, ikasleen NAN-ak gorde beharko bagenitu, arraiak erabiliko genituzke.100 NAN gordetzeko arrairik gabe 100 aldagai beharko genituzke:

nan_ikas1, nan_ikas2, nan_ikas3, nan_ikas4, e.a.: integer;

- Arraiekin, *T_nan_ikasle* arraia definituko genuke

- Honela T_nota_ikasle definituko genuke jarraian ikus dezakegun bezala:

type T_nan_ikasle is array (1..100) of integer;

- Modu honetan, gure programetan bakarrik deklaratu beharko dugu mota berri honetako aldagai bat:

Ikas_nan: T_nan_ikasle;

Honela, lerro sinple honekin memoria alokatzen da 100 elementuentzat

Abantaila gehiago

- Azpiprogrameei 100 datu horiek pasatzeko, arraia baino ez diot pasatu behar, adibidez ikasleen nan-en arteko handiena topatzeko azpiprograma.

function nan_handiena (nan_ikas1, nan_ikas2, nan_ikas3
e.a. 100 arte: in integer) return integer is

function nan_handiena (ikasleak: in T_nan_ikasleak)
return integer is

Abantaila gehiago

- Arrai bat hasieratzea errazagoa da 100 aldagai hasieratzea baino.

- For ind in 1..100 loop (edo 'last edo 'first erabiliaz)

```
        ikas_nan(ind):=22222222;
   end loop;
Edo
     ikas_nan:=(others =>22222222);
```

Algoritmika vs. ADA

```
package datuak is
    type T_sekuentzia is array (1..12) of integer;
end_datuak;
```

Algoritmika	ADA

seku: 12 integer;
hasieran_kokatu(seku);
errepikatu
a:=egungo_elementua(seku);
aurrera_egin(seku);
Irten denean sekuentziatik_kanpo(seku);
	seku: T_sekuentzia;
	ind:=1;
	loop
	a:=seku(ind);
	ind:=ind+1;
	exit when ind > secu'last;

- Orain 4 mota berri erabil ditzazkegu:
  - Osokoen_bektorea
  - Errealen_bektorea
  - Boolearren_bektorea
  - Karakteren_bektorea
- Eta euren definizioa hurrengoa litzateke
  - Type Osokoen_bektorea is array (integer range <>) of integer;
  - Type Errealen_bektorea is array (integer range <>) of float;
  - Type Boolearren_bektorea is array (integer range <>) of boolean;
  - Type Karakteren_bektorea is array (integer range <>) of character;

Definizioa vs Deklarazioa

- 2 modu sekuentzia datu mota berri bat sortzeko

Definiziotik tamainan mugatua

type T_NAN_ikasleak is array (1..100) of integer;

Definiziotik tamainan mugatugabea

type T_DNIs_alumnos is array (integer range <>) of integer;

Baina orduan deklaratzeko momentuan mugatu behar da

NANak: T_NAN_ikasleak(1..100);

Hasieraketa

- Orokorra

  For ind in 1..100 loop *(edo ‘last eta ‘first erabiliaz)*

  NANak(ind):=22222222;

  end loop;

- Bakarrik Adaz

  NANak(others =>22222222);

Sekuentzia_irakurri eta NAN_handiena

- Hurrengo datu motarekin
  Type T_nan_ikasleak is array (1..100) of integer;

- Idatzi hurrengo azpiprograma *('last)*
  Procedure/function sekuentzia_irakurri*(T: ... T_nan_ikasleak) ...*

- Ikasleen nan-en arteko handiena kalkulatuko duen programa idatzi

..... nan_handiena (T: ... T_nan_ikasleak) .....

- Idatzi hurrengo azpiprograma *('last)*
  Procedure/function inprimatu_sekuentzia*(T: ... T_nan_ikasleak) ... is*

Hilabetearen_izena_lortu

```
package datuak is
----subtype T_hilabete is string(1..10); ---Hilabetean
izen bat gordetzeko
----type t_hilabeteak is array (1..12) of T_hilabete;
type t_hilabeteak is array (1..12) of string(1..10);
end datuak;
```

Idatzi hilabetearen_izena_lortu azpiprograma

Idatzi azpiprograma bat hilabete bati dagokion zenbakia pasata hilabetearen izena idatziko duena (1 Urtarrila, 2 Otsaia e.a.)

Desplazatu ezkerretara

```
Package datuak is
    type osokoen_bektorea is array (1..10) of integer;
end datuak;
```

Gure ariketan adibidez hurrengo aldagaia izango dugu

```
taula: osokoen_bektorea;
```

Demagun sekuentzia_irakurri(taula)
Eta erabiltzaileak sartu duela hurrengoa

1	2	3	4	5	6	7	8	9	10
3	7	1	3	9	12	23	5	8	3

Idatzi azpiprograma bat arraiaren elementuak ezkerretara rotatzeko

Bektoreen bektoreak: matrizeak

- Bektore baten elementuak atzitzeko indize bat erabiliko dugu
- Matrizeetan 2 indize erabiliko dira
- Orokorrean, erabili daitezke behar diren bezainbeste indize (matrizearen dimentsioaren arabera)

Hemendik aurrera matrizeak ezagutzen ditugula 4 matrize erabili ditzazkegu:

- Osokoen_matrizea
- Errealen_matrizea
- Boolearren_matrizea
- Karaktereen_matrizea

matrize motatako aldagiaiak deklaratzeko:

- Matrizearen izena
- Elementu kopurua
- Mota

Type boolearren_matrizea is array (integer range<>,integer range <>) of boolean;
Type karaktere_matrizea is array (integer range<>,integer range <>) of character;

- Adibideak:

  Itsazontziak: boolearren_matrizea(1 .. 10,1 .. 10);
  Pantaila: karaktere_matrizea(1 .. 24,1 .. 80);

Definizioa vs deklarazioa

Definizio limitatua:

```
type osokoen_matrizea is array (1..10, 1..10) of integer;
type boolearren_matrizea is array (1..10,1..10) of boolean;
```

Deklarazioa:

```
Barkuak: osokoen_matrizea;
```

Limitatu gabeko definizioa:

```
type osokoen_matrizea2 is array (integer range <>, integer range <>) of integer;
```

Deklarazioa:

```
Barkuak: osokoen_matrizea2(1..10,1..10);
```

Definizioa vs deklarazioa

Definizio limitatua:

```
type osokoen_bektorea is array (1..10) of integer;
type osokoen_matrizea is array (1..10) of osokoen_bektorea;

type karaktereen_bektorea is array (1..80) of character
Type karaktereen_matrizea is array (1..24) of
karaktereen_bektorea
```

Deklarazioa:

```
Barkuak: osokoen_matrizea;
```

- Elementu bat atzitzeko segun eta nola dagoen definituta:
  - Pantaila(6, 25) := 'F';

Ala

  - Pantaila(6)(25) := 'F';

```
Package datuak is
            type osokoen_bektorea is array (1..10) of integer;
            type matrizeen_bektorea is array (1..10) of osokoen_bektorea;
 end datuak;
```

Demagun Matrizea_pantailaratu (M: in osokoen_matrizea) idatzi nahi dugula

```
Procedure Matrizea_pantailaratu (M: in osokoen_matrizea) is
      begin
            for lerroa in 1..M'last loop
                  for elem_zutabe in 1..M(lerroa)'last loop
                        put(M(lerroa)(elem_zutabe));
                  end loop;
            end loop;
End Matrizea_pantailaratu;
```

Ariketa: Matrizea bete eta idatzi matrize moduan

```
Package datuak is
                type osokoen_matrizea is array (1..10, 1..10) of integer;
  end datuak;
```

Demagun Matrizea_pantailaratu (M: in osokoen_matrizea) idatzi nahi dugula

```
Procedure Matrizea_pantailaratu (M: in osokoen_matrizea) is
      begin
            for lerroa in 1..M'last(1) loop
                  for elem_zutabe in 1..M'last(2) loop
                   put(M(lerroa,elem_zutabe));
                     end loop;
            end loop;
End Matrizea_pantailaratu;
```

Ariketa: Matrizea bete eta idatzi matrize moduan

- Adibideak
  - Aldagaien deklarazioa:

matrize: osokoen_matrizea(1..5,1..10);

matriz

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

Matrizeak: Hasieraketa

```
for i in 1 .. 5 loop
    for j in 1 .. 10 loop
            matrize(i, j) := i + j;
    end
end
```

matriz

	1	2	3	4	5	6	7	8	9	10
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**
2	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**
3	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
4	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**

Adibide batzuk

- Honelako aldagaiak deklaratu ditzakegu:
  M1, M2: osokoen_matrizea(1..5, 1..10);
  M1(3,j):=123;

- Beste modu bat:
- type lerroa is array (1..10) of integer;
- type matrizea is array (1..5) of lerroa;
- M3,M4: matrizea;
- L:lerroa

- M3(3)(j):=123; -- bi indize behar dira

- F:=M3(3); -- F aldagaiak M matrizeko 3.lerroa jasoko du.