

## Ariketa 1: Urbanizazioa

Urbanizazio bat osatzen duten eraikin eta horien barruan dauden apartamentuen informazioa (salneurria, egungo jabea eta antzinako jabearen informazioak besteak beste), hurrengo datu moten definizioak ditugu. Urbanizazioak 20 eraikinez osatua dago, eta eraikin bakoitzak 10 solairu ditu. Solairu batean egon daiteken apartamentu kopurua aldakorra da; 1tik 8ra, hau da, solairu batzuetan apartamentu bakar bat egon daiteke aldiz beste batzuetan 6 , edo 8, e.a.

```
type Identifikazioa is record
    Izen,
    Abizen : String (1 .. 20);
    NAN    : Integer;
end record;

type Info_Apartamentu is record
    Salneurria      : Integer;
    Egungo_jabea    : Identifikazioa;
    Aurreko_jabea   : Identifikazioa;
end record;

Solairuko_apartamentu_kop_maximoa: constant Integer := 8;

type Taula_Apartamentuak is array
    (1 .. Solairuko_apartamentu_kop_maximoa) of Info_Apartamentu;

type Solairu is record
    Apartamentuak: Taula_Apartamentuak;
    Apartamentu_kop: Integer; -- 1 eta 8 arteko balioa
end record;

Max_Solairu : constant Integer := 10;

type Eraikina is array (1 .. Max_solairu) of Solairu;

Num_Bloques : constant Integer := 20;

type Urbanizazioa is array (1 .. Bloke_kop) of Eraikina;
```

Hurrengo espezifikazioa jarraitzen duen azpiprograma idatz ezazue:

```
function Ondasun_handienaren_jabearen_NAN (U: in Urbanizazioa) return Integer

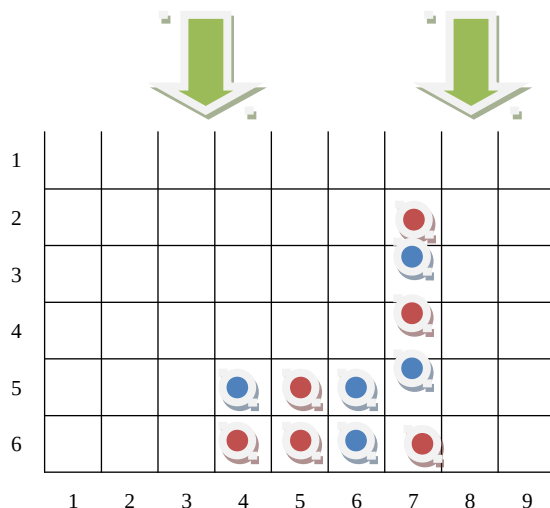
-- aurre:
-- post: Emaizta Urbanizazioan apartamentuetan inbertitua duen ondasunik handienaren
--- jabearen NAN zenbakia bueltatuko dugu. KONTUZ!!! patrimonioa ez da apartamentu kopurua
baizik eta dauzkan apartamentuen salneurri gehiketa

function Dago ( NAN : in Integer; Lp : in Lista_Pertsonak ) return Integer is
    -- postbaldintza: emaitza, Lp listan agertzen den NANaren posizioa izango da.
    --Ez badago, 0 izango da emaitza

function nan_maximoa (Lp : in Lista_Pertsonak ) return Integer is
    -- aurrebaldintza: LP-k behintzat elementu bat du
    -- postbaldintza: emaitza, LPn ondare handiena duenaren NANA izango da
```

## Ariketa 2: Konekta4

Konekta4 2 jokalarari txandaka jolasteko joko bat da. Jokalari bakoitzak kolore bat izango du esleituia (gorria edota urdina), eta bere txanda denean, esleituia duen koloreko fitxa bat sartu beharko du 9 zutabe eta 6 errenkada duen taula bertikal batean. Taula bertikala denez, fitxak taularen goiko partean uztean, fitxak bertikalki jauzten dira, zutabean libre dagoen posizio baxuena arte. Helburua, kolore bereko 4 fitxa jarraian konektatzea izango da. Konexioa bertikalki, zein horizontalki edo diagonalki gertatu daiteke. 4 fitxatako konexioa lortzen duen lehenengoak irabazten du.



fitxa\_kokatu azpiprograma inplementatzea eskatzen da. Ohartu, ez dela eskatzen joko osoa inplementatzea (beste hainbat azpiprograma beharko lirateke kasu horretan).

fitxa\_kokatu azpiprogramak, tableroa edukita, kolore bat eta fitxa zein zutabean sartu nahi den jakinda, jokaldi horrekin 4 fitxa konektatzen diren adieraziko du, aldi berean tableroa eguneratuta utziz (ikus espezifikazioa).

### Datu moten definizioak

```
package konekta4 is
  Max_Zutabe: constant integer:=9;
  Max_Errenkada: constant Integer:=6;
  type T_Kutxatxo is (Gorria,Urdina,Ezer);
  subtype T_Fitxa is T_Kutxatxo range Gorria..Urdina;
  subtype T_Errenkada is Integer range 1..Max_Errenkada;
  subtype T_Zutabe is Integer range 1..Max_Zutabe;
  type T_Taula is array (1..Max_Errenkada,1..Max_Zutabe) of T_Kutxatxo;
end konekta4;
```

### Garatu behar den azpiprograma

```
{procedure/function? Fitxa_kokatu (Taula:?? T_Taula; Kolore:?? T_Fitxa; Zutabea:
?? T_Zutabe; IrabaziBai:?? Boolean.....) ??
  -- Aurre: Zutabea parametroak markatutako zutabearen gutxienez fitxa bat
  sartzeko lekurik badago.
  -- Post: Taula eguneratu da, Kolore-ko fitxa bat Zutabe-an dagoen lehenengo
  errenkada librean (behetik hasita) kokatuz. Gainera, partida irabazi den
  adieraziko da; irabaziBai-k true balioko du 4 fitxa konektatzen direnean.
```

```

with konprobatu_fitxak_norantzan;
function/procedure fitxa_kokatu_ikasleak(Taula:in out T_Taula; Kolore: in
T_Fitxa ;Zutabe:in T_Zutabe; IrabaziBai:out Boolean) ... is
    -- Aurre: Zutabea parametroak markatutako zutabeen gutxienez fitxa bat
    sartzeko lekurik dago.

    -- Post: Taula eguneratu da, Kolore-ko fitxa bat Zutabe-an dagoen lehenengo
    errenkada librean (behetik hasita) kokatuz. Gainera, partida irabazi den
    adieraziko da; irabaziBai-k true balioko du 4 fitxa konektatzen direnean.

function/procedure konprobatu_fitxak_norantzan_ikasleak (X) ... is
    -- Sarrera: Taula, errenkada, zutabea eta norantza.
    -- Irteera: Boolean bat. Azpiprograma honen eginkizuna taulan kolore
    berdineko 4 fitxa jarraian dauden ala ez aztertzea izango da.
    -- Gure inplementazioan, azpiprograma honi 4 aldiz deitzen diogu, taulako 4
    norantzak aztertzeko, hau da:
    -- Norantza:
    -- 1 Bertikala: bertikalean kolore berdineko 4 fitxa dauden begiratuko da
    -- 2 Horizontala: horizontalean kolore berdineko 4 fitxa dauden begiratuko da
    -- 3 Diagonala beherantz: beherantzko diagonalean kolore berdineko 4 fitxa
    dauden begiratuko da
    -- 4 Diagonala gorantz: gorantzako diagonalean kolore berdineko 4 fitxa
    dauden begiratuko da
    -- OHARRA: ez duzue zertan gure inplementazio berdina egin behar

begin
case Norantza is
    when 1 =>
        Errenkada_Inkrementatu:=1;           Zutabea_Inkrementatu:=0;
    when 2 =>
        Errenkada_Inkrementatu:=0;           Zutabea_Inkrementatu:=1;
    when 3 =>
        Errenkada_Inkrementatu:=1;           Zutabea_Inkrementatu:=1;
    when 4 =>
        Errenkada_Inkrementatu:=-1;          Zutabea_Inkrementatu:=1;
    when others=>
        Errenkada_Inkrementatu:=0;           Zutabea_Inkrementatu:=0;
end case;

    -- aztertu taulako posizioak

function/procedure hasieratu_taula_ikasleak(X)... is
    -- Sarrera: Taula
    -- Irteera: Taulako posizio guztiak "EZER" hitzarekin beteko da eta taula
    itzuliko da.

function/procedure eskatu_zutabea_ikasleak(X) ... is
    -- Sarrera: Taula
    -- Irteera: Taulako zutabe baten zenbakia. Azpiprogramak erabiltzaileari
    taulako zutabe
    -- baten zenbakia eskatuko dio. Zutabea, beteta dagoen ala ez aztertuko da.
    Beteta badago,
    -- berriz eskatuko dio zenbakia, bestela, zenbakia itzuliko du.

function/procedure inprimatu_taula_ikasleak (X) ... is
    -- Sarrera: Taula
    -- Irteera: Taula pantailaratuko da
    --Enumeratuak idatzi (put) ahal izateko:
    package P_Laukitxo_IO is new Enumeration_IO(T_Kutxatxo); use P_Laukitxo_IO;

```