

## ***PB– 6. Laborategiko Informea***

**Izena:** Eneko, Markel eta Gontzal **Data:** 2014-11-12



### ***Ikasitako kontzeptuak eta lortutako gaitasunak***

Behin laborategia bukatu dudala, eskatu dizkidaten ariketetan erabiltzen diren kontzeptuak ondo ulertu ditudala kontsideratzen dut, eta beraz ariketak errepikatzeko gai izango nintzatekela esan dezaket. Erantzun hurrengo galdera-sorta esanez ea gai zaren (ala ez) hor galdetzen dena egiteko.

Eneko		Bai	Ez
1	Badakit taula bateko elementu bat topatzen modu eragikorrean	Bai	
2	Badakit array bati dagokion datu mota berri bat sortzen	Bai	
3	a) eta b) bertsioen arteko desberdintasuna ondo ulertzen dut eta bakoitza noiz erabili behar den ere ondo ulertzen dut  a) type T_Osoak is array (1..100) of Integer;  b) type T_Osoak is array (1..100) of Integer;  type Info_Osoak   is record  Kopurua: integer;  Zenbakiak: T_Osoak;  end record;	Bai	

Gontzal		Bai	Ez
1	Badakit taula bateko elementu bat topatzen modu eragikorrean	Bai	
2	Badakit array bati dagokion datu mota berri bat sortzen	Bai	
3	<p>a) eta b) bertsioen arteko desberdintasuna ondo ulertzen dut eta bakoitza noiz erabili behar den ere ondo ulertzen dut</p> <p>a) type T_Osoak is array (1..100) of Integer;</p> <p>b) type T_Osoak is array (1..100) of Integer;</p> <p>type Info_Osoak is record</p> <p>Kopurua: integer;</p> <p>Zenbakiak: T_Osoak;</p> <p>end record;</p>	Bai	

Markel		Bai	Ez
1	Badakit taula bateko elementu bat topatzen modu eragikorrean	Bai	
2	Badakit array bati dagokion datu mota berri bat sortzen	Bai	
3	<p>a) eta b) bertsioen arteko desberdintasuna ondo ulertzen dut eta bakoitza noiz erabili behar den ere ondo ulertzen dut</p> <p>a) type T_Osoak is array (1..100) of Integer;</p> <p>b) type T_Osoak is array (1..100) of Integer;</p> <p>type Info_Osoak is record</p> <p>Kopurua: integer;</p> <p>Zenbakiak: T_Osoak;</p> <p>end record;</p>	Bai	



## **6. Laborategiko soluzioa**

8. Bilaketa matrize batean.  $N \times M$  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

### **proba\_matrizean\_badago.adb**

```
with Ada.Text_IO, Ada.Integer_Text_IO;
use Ada.Text_IO, Ada.Integer_Text_IO;

with bektore_eta_matrizeak; use bektore_eta_matrizeak;
with matrizean_badago;

procedure proba_matrizean_badago is
  -- programa honek matrizean_badago azpiprogramari deituko dio proba kasu
  ezberdinekin,
  -- zuen kodifikatutako azpiprogramaren funtzionamendua konprobatzeko

  M1: osokoenMatrize(1..4, 1..10);---bete zuek sortutako aldagai izenarekin,
  ---bektore_eta_matrizeak.ads fitxategian

  topatu daitekena

  Posizio_lerro, Posizio_zutabe: Integer;

begin
  M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (5, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
  -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
  Put_line("proba1: elementuaa matrizearen bukaeran dago");
  Put_line(" matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 1));");
  Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1));");
  Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 1));");
  Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
  Posizio_lerro, Posizio_zutabe));");
  put_line(" Eraitza izan beharko litzateke elementua=19 lerro=4 zutabe=10 ");
  matrizean_badago(M1, 19, Posizio_lerro, Posizio_zutabe);
  Put(19); Put(Posizio_lerro); Put(Posizio_zutabe);
  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 199),
         (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (5, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
```

```

-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba2: elementua 1. lerroaren bukaeran");
Put_line(" matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 199))");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_erro, Posizio_zutabe)");
put_line("Emitza izan beharko litzateke elementua=199  erro=1  zutabe=10
");
matrizean_badago(M1, 199, Posizio_erro, Posizio_zutabe);
Put(199); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba3: elementua 3.aren lerroaren bukaeran ");
Put_line(" matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_erro, Posizio_zutabe)");
put_line(" Emitza izan beharko litzateke elementua=199  erro=3  zutabe=10
");
matrizean_badago(M1, 199, Posizio_erro, Posizio_zutabe);
Put(199); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

M1 := ((1999, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba4: elementua matrizearen hasieran");
Put_line(" matrizean_badago((1999, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_erro, Posizio_zutabe)");
put_line("Emitza izan beharko litzateke elementua=1999  erro=1  zutabe=1
");
matrizean_badago(M1, 1999, Posizio_erro, Posizio_zutabe);
Put(1999); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (799, 3, 5, 7, 9, 11, 13, 15, 17, 19));
    -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba5: elementua azkeneko lerroaren hasieran");
Put_line("matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (799, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_lerro, Posizio_zutabe)");
put_line(" Emaizta izan beharko litzateke elementua=799  lerro=4  zutabe=1
");
matrizean_badago(M1, 799, Posizio_lerro, Posizio_zutabe);
Put(799); Put(Posizio_lerro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

M1 := ((1, 3, 5, 7, 9, 1111, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
    -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba6: elementua lehenengo lerroaren erdikaldean");
Put_line("matrizean_badago((1, 3, 5, 7, 9, 1111, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_lerro, Posizio_zutabe)");
put_line(" Emaizta izan beharko litzateke elementua=1111  lerro=1  zutabe=6
");
matrizean_badago(M1, 1111, Posizio_lerro, Posizio_zutabe);
Put(1111); Put(Posizio_lerro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 1111, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
    -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba7: elementua erdiko lerro batean, ez azkenekoan ez
lehenengoan");
Put_line("matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 1111, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), elementua,
Posizio_lerro, Posizio_zutabe)");

```

```

put_line(" Eaitza izan beharko litzateke elementua=1111  lerro=3  zutabe=6
");
matrizean_badago(M1, 1111, Posizio_erro, Posizio_zutabe);
Put(1111); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 1311, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba8: elementua azkeneko lerroaren erdikaldean");
Put_line(" matrizean_badago((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("                (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("                (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("                (7, 3, 5, 7, 9, 11, 1311, 15, 17, 19)), elementua,
Posizio_erro, Posizio_zutabe)");
put_line(" Eaitza izan beharko litzateke elementua=1311  lerro=4  zutabe=7
");
matrizean_badago(M1, 1311, Posizio_erro, Posizio_zutabe);
Put(1311); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

end proba_matrizean_badago;

```

### matrizean\_badago.adb

```

with bektore_eta_matrizeak;
use bektore_eta_matrizeak;

procedure matrizean_badago (L: in osokoenMatrize; zen: in integer; lerro,
zutabe: out integer) is

    aurkituta: boolean;

begin

    aurkituta := False;
    lerro := 1;
    loop exit when( lerro > L'last(1) or aurkituta = True );
    zutabe := 1;
    loop exit when( (zutabe > L'last(2)) or aurkituta = True );

        if( zen = L(lerro, zutabe) ) then
            aurkituta := True;
            lerro := lerro - 1;
        else
            zutabe := zutabe + 1;
        end if;
    end loop;
    lerro := lerro + 1;
end loop;

```

```
end matizean_badago;
```

9. Matrize baten maximoa.  $N \times M$  osokoen matrize bat hartuta, matrize horretan elementurik handiena zein den eta bere posizioa kalkulatzeko azpiprograma idatzi. Zenbakia errepikaturik balego, aurreneko posizioa itzuli. Erabil proba\_maximoa\_kalkulatu.adb, maximoa\_kalkulatu.adb probatzeko

#### proba\_maximoa\_kalkulatu.adb

```
with Ada.Text_IO, Ada.Integer_Text_IO;
use Ada.Text_IO, Ada.Integer_Text_IO;

with bektore_eta_matrizeak; use bektore_eta_matrizeak;
with maximoa_kalkulatu;

procedure proba_maximoa_kalkulatu is
  -- programa honek maximoa_kalkulatu azpiprogramari deituko dio proba kasu
  ezberdinekin,
  -- zuen kodifikatutako azpiprogramaren funtzionamendua konprobatzeko

  M1: osokoenMatrize(1..4, 1..10);---bete zuek sortutako aldagai izenarekin,
  ---bektore_eta_matrizeak.ads fitxategian

  topatu daitekena

  Maximo, Posizio_lerro, Posizio_zutabe: Integer;

begin
  M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (5, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
  -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
  Put_line("proba1: maximoa matrizearen bukaeran dago");
  Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 1))");
  Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
  Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
  Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
  Posizio_lerro, Posizio_zutabe)");
  put_line(" Emaita izan beharko litzateke Maximo=19 lerro=4 zutabe=10 ");
  maximoa_kalkulatu(M1, Maximo, Posizio_lerro, Posizio_zutabe);
  Put(Maximo); Put(Posizio_lerro); Put(Posizio_zutabe);
  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 199),
         (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (5, 3, 5, 7, 9, 11, 13, 15, 17, 1),
         (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
  -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
  Put_line("proba2: maximo 1. lerroaren bukaeran");
  Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 199))");
  Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
  Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
  Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
  Posizio_lerro, Posizio_zutabe)");
```



```

put_line("Eaitza izan beharko litzateke Maximo=199  lerro=1  zutabe=10
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba3: maximo 3.aren lerroaren bukaeran ");
Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
Posizio_erro, Posizio_zutabe)");
put_line(" Eaitza izan beharko litzateke Maximo=199  lerro=3  zutabe=10
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1999, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba4: maximo matrizearen hasieran");
Put_line(" Maximoa_kalkulatu((1999, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
Posizio_erro, Posizio_zutabe)");
put_line("Eaitza izan beharko litzateke Maximo=1999  lerro=1  zutabe=1
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (799, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba5: maximo azkeneko lerroaren hasieran");
Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");

```

```

Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (799, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
Posizio_erro, Posizio_zutabe)");
put_line(" Eaitza izan beharko litzateke Maximo=799  erro=4  zutabe=1
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 1111, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba6: maximo lehenengo erroaren erdikaldean");
Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 1111, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
Posizio_erro, Posizio_zutabe)");
put_line(" Eaitza izan beharko litzateke Maximo=1111  erro=1  zutabe=6
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (5, 3, 5, 7, 9, 1111, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 13, 15, 17, 19));
-- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba7: maximo erdiko erro batean, ez azkenekoan ez
lehenengoan");
Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("          (5, 3, 5, 7, 9, 1111, 13, 15, 17, 199)");
Put_line("          (7, 3, 5, 7, 9, 11, 13, 15, 17, 19)), Maximo,
Posizio_erro, Posizio_zutabe)");
put_line(" Eaitza izan beharko litzateke Maximo=1111  erro=3  zutabe=6
");
maximoa_kalkulatu(M1, Maximo, Posizio_erro, Posizio_zutabe);
Put(Maximo); Put(Posizio_erro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

M1 := ((1, 3, 5, 7, 9, 11, 13, 15, 17, 1),
        (3, 3, 5, 7, 9, 11, 13, 15, 17, 1),

```

```

        (5, 3, 5, 7, 9, 11, 13, 15, 17, 199),
        (7, 3, 5, 7, 9, 11, 1311, 15, 17, 19));
        -- to lo mismo que hacer M1(1,1) := 1; ... M1(4, 10) := 19;
Put_line("proba8: maximo azkeneko lerroaren erdikaldean");
Put_line(" Maximoa_kalkulatu((1, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("                (3, 3, 5, 7, 9, 11, 13, 15, 17, 1)");
Put_line("                (5, 3, 5, 7, 9, 11, 13, 15, 17, 199)");
Put_line("                (7, 3, 5, 7, 9, 11, 1311, 15, 17, 19)), Maximo,
Posizio_lerro, Posizio_zutabe)");
put_line(" Emaizta izan beharko litzateke Maximo=1311 lerro=4 zutabe=7
");
maximoa_kalkulatu(M1, Maximo, Posizio_lerro, Posizio_zutabe);
Put(Maximo); Put(Posizio_lerro); Put(Posizio_zutabe);
New_Line(3);
Put_Line("Sakatu return jarraitzeko");
Skip_Line;
New_Line(3);

```

```

end proba_maximoa_kalkulatu;

```

### maximoa\_kalkulatu.adb

```

with bektore_eta_matrizeak;
use bektore_eta_matrizeak;

```

```

procedure maximoa_kalkulatu (L: in osokoenMatrize; zen, lerroa, zutabea: out
integer) is
    -- pre: M-k edukiko du behintzat elementu bat
    -- post: matrize horretan elementurik handiena zein den eta bere posizioa (X,Y
koordinatuak)
    -- bueltatuko ditu

```

```

    lerro, zutabe: integer;

```

```

begin

```

```

    lerroa:= 1 ; zutabea := 1;
    zen := L(1,1);
    lerro := 1;

```

```

    loop exit when( lerro > L'last(1) );
        zutabe := 1;
        loop exit when( (zutabe > L'last(2)));

```

```

            if(L(lerro,zutabe) > zen) then
                zen := L(lerro,zutabe);
                lerroa := lerro;
                zutabea := zutabe;
            end if;

```

```

            zutabe := zutabe + 1;

```

```

        end loop;

```

```

        lerro := lerro + 1;

```

```

    end loop;

```

```

end maximoa_kalkulatu;

```

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

#### proba\_ordenatu\_txertatuz.adb

```
with Ada.Text_IO;
use Ada.Text_IO;

with defs_lab; use defs_lab;
with ordenatu_txertatuz, idatzi_Lista;

procedure proba_ordenatu_txertatuz is
  -- este programa hace llamadas al subprograma ordenar_por_inserción
  -- para comprobar si su funcionamiento es correcto

  Lista1, Lista_emaiza: Osokoen_zerrenda;

begin

  Lista1.zenbakiak(1) := 1;
  Lista1.kopurua:= 1;
  Put_line("proba1: elementu bakarreko bektorea: (1)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_txertatuz(Lista1, Lista_emaiza);
  Idatzi_Lista(Lista_Emaiza);
  Put_line(" da ");
  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6;
  Lista1.kopurua := 5;
  Put_line("proba2: hainbat zenbaki ez ordenatuez osatutako bektorea: (3 1 4 8
6)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_txertatuz(Lista1, Lista_emaiza);
  Idatzi_Lista(Lista_Emaiza);
  Put_line(" da ");

  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6; Lista1.zenbakiak(6) := 23;
```

```

    Lista1.zenbakiak(7) := 5; Lista1.zenbakiak(8) := 13; Lista1.zenbakiak(9) :=
25;
    Lista1.zenbakiak(10) := 7;
    Lista1.kopurua := 10;
    Put_line("proba3: hainbat zenbaki ez ordenatuez osatutako bektorea
(erreserbatu dugun guztia beteta): (3 1 4 8 6 23 5)");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_txertatuz(Lista1, Lista_emaiza);
    Idatzi_Lista(Lista_Emaiza);
    Put_line(" da ");
    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

    Lista1.zenbakiak(1) := 1; Lista1.zenbakiak(2) := 3; Lista1.zenbakiak(3) := 5;
    Lista1.zenbakiak(4) := 7; Lista1.zenbakiak(5) := 9;
    Lista1.kopurua:= 5;
    Put_line("proba4: lista ordenatua: (1 3 5 7 9)");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_txertatuz(Lista1, Lista_emaiza);
    Idatzi_Lista(Lista_Emaiza);
    Put_line(" da ");

    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

    Lista1.kopurua:= 0;
    Put_line("proba5: lista hutsa: ( )");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_txertatuz(Lista1, Lista_emaiza);
    Idatzi_Lista(Lista_Emaiza);
    Put_line(" da ");

    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

end proba_ordenatu_txertatuz;

```

### ordenatu\_txertatuz.adb

```
with defs_lab;
use defs_lab;

with txertatzeko_posizioa_topatu, eskuinera_mugitu;

procedure ordenatu_txertatuz (L1 : in Osokoen_zerrenda; L_Ordenatua : out
Osokoen_zerrenda) is

    Pos : Integer;
    i: integer;

begin

    L_Ordenatua.kopurua := 1;
    i := 2;
    L_Ordenatua.zenbakiak(1) := L1.zenbakiak(1);

    loop exit when(i > L1.kopurua);

        txertatzeko_posizioa_topatu(L1.zenbakiak(i), L_Ordenatua, Pos);

        eskuinera_mugitu(L_Ordenatua, Pos, L_Ordenatua);

        L_Ordenatua.kopurua := L_Ordenatua.kopurua + 1;

        L_Ordenatua.zenbakiak(Pos) := L1.zenbakiak(i);

        i := i + 1;

    end loop;

end ordenatu_txertatuz;
```

### txertatzeko\_posizioa\_topatu.adb

```
with defs_lab;
use defs_lab;

procedure txertatzeko_posizioa_topatu (zenb: in Integer ; L: in osokoen_zerrenda;
Pos: out Integer ) is

    i: integer;

begin

    i := 1;
    Pos := 1;

    loop exit when (i > L.kopurua);

        if(L.zenbakiak(i) < zenb ) then
            Pos := Pos + 1;
        end if;

    end loop;
```

```
        i := i + 1;  
    end loop;  
end txertatzeko_posizioa_topatu;
```

#### **eskuinera\_mugitu.adb**

```
with defs_lab;  
use defs_lab;  
  
procedure eskuinera_mugitu (zerrenda: in Osokoen_zerrenda ; Pos: in integer ;  
    zerrenda_mugituta: out Osokoen_zerrenda) is  
  
begin  
    for i in reverse Pos..zerrenda.kopurua loop  
        zerrenda_mugituta.zenbakiak(i+1) := zerrenda.zenbakiak(i);  
    end loop;  
  
end eskuinera_mugitu;
```

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

#### proba\_ordenatu\_hautatuz.adb

```
with Ada.Text_Io;
use Ada.Text_Io;

with defs_lab; use defs_lab;
with ordenatu_hautatuz, idatzi_Lista;

procedure proba_ordenatu_hautatuz is
  -- este programa hace llamadas al subprograma ordenar_por_inserción
  -- para comprobar si su funcionamiento es correcto

  Lista1: Osokoen_zerrenda;

begin
  Lista1.zenbakiak(1) := 1;
  Lista1.kopurua := 1;
  Put_line("proba1: elementu bakarreko bektorea: (1)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_hautatuz(Lista1);
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6;
  Lista1.kopurua := 5;
  Put_line("proba2: hainbat zenbaki ez ordenatuez osatutako bektorea: (3 1 4 8
6)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_hautatuz(Lista1);
  Idatzi_Lista(Lista1);
  Put_line(" da ");

  New_Line(3);
  Put_Line("Sakatu return jarraitzeko");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6; Lista1.zenbakiak(6) := 23;
```



```

    Lista1.zenbakiak(7) := 2; Lista1.zenbakiak(8) := 13; Lista1.zenbakiak(9) :=
25;
    Lista1.zenbakiak(10) := 7;
    Lista1.kopurua:= 10;
    Put_line("proba3: hainbat zenbaki ez ordenatuez osatutako bektorea
(erreserbatu dugun guztia beteta): (3 1 4 8 6 23 2 13 25 7)");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_hautatuz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

    Lista1.zenbakiak(1) := 1; Lista1.zenbakiak(2) := 3; Lista1.zenbakiak(3) := 5;
    Lista1.zenbakiak(4) := 7; Lista1.zenbakiak(5) := 9;
    Lista1.kopurua := 5;
    Put_line("proba4: lista ordenatua: (1 3 5 7 9)");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_hautatuz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");

    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

    Lista1.kopurua := 0;
    Put_line("proba5: lista hutsa: ( )");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_hautatuz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");

    New_Line(3);
    Put_Line("Sakatu return jarraitzeko");
    Skip_Line;
    New_Line(3);

end proba_ordenatu_hautatuz;

```

### ordenatu\_hautatuz.adb

```
with defs_lab;
use defs_lab;

with minimoaren_posizioa_topatu;

procedure ordenatu_hautatuz (L : in out osokoen_zerrenda ) is
  -- pre:
  -- post: L contiene los valores que tenía inicialmente, ordenados
  --        de menor a mayor
  Pos_Min, aux, aux2 : Integer;
begin
  aux:= 1;
  loop exit when aux >= L.kopurua;
    minimoaren_posizioa_topatu(aux, L, Pos_Min);
    aux2:= L.zenbakiak(aux);
    L.zenbakiak(aux) := L.zenbakiak(Pos_Min);
    L.zenbakiak(Pos_Min):= aux2;
    aux := aux + 1;
  end loop;
end ordenatu_hautatuz;
```

### minimoaren\_posizioa\_topatu.adb

```
with defs_lab;
use defs_lab;

procedure minimoaren_posizioa_topatu (hasiera : in Integer; L: in
Osokoen_zerrenda; Pos_Min: out Integer) is
  min: integer:=integer'last;
  aux: integer;

begin
  aux:= hasiera;
  loop exit when aux > L.kopurua;
    if min > L.zenbakiak(aux) then
      Pos_Min:= aux;
      min := L.zenbakiak(aux);
    end if;
    aux := aux + 1;
  end loop;
end minimoaren_posizioa_topatu;
```

## 12. Burbuia metodoa jarraituz bektore bat ordenatu

### proba\_ordenatu\_burbuiaz.adb

```
with Ada.Text_IO;
use Ada.Text_IO;

with defs_lab; use defs_lab;
with ordenatu_burbuiaz, idatzi_Lista;

procedure Proba_ordenatu_burbuiaz is

  Lista1: osokoen_zerrenda;

begin
  Lista1.zenbakiak(1) := 1;
  Lista1.kopurua:= 1;
  Put_line("proba1: elementu bakarreko bektorea: (1)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_burbuiaz(Lista1);
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  New_Line(3);
  Put_Line("Pulsa return para continuar");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6;
  Lista1.kopurua := 5;
  Put_line("proba2: hainbat zenbaki ez ordenatuez osatutako bektorea: (3 1 4 8
6)");
  Put_line(" sarrerako lista: ");
  Idatzi_Lista(Lista1);
  Put_line(" da ");
  put_line(" eta zure emaitza ");
  ordenatu_burbuiaz(Lista1);
  Idatzi_Lista(Lista1);
  Put_line(" da ");

  New_Line(3);
  Put_Line("Pulsa return para continuar");
  Skip_Line;
  New_Line(3);

  Lista1.zenbakiak(1) := 3; Lista1.zenbakiak(2) := 1; Lista1.zenbakiak(3) := 4;
  Lista1.zenbakiak(4) := 8; Lista1.zenbakiak(5) := 6; Lista1.zenbakiak(6) := 23;
  Lista1.zenbakiak(7) := 2; Lista1.zenbakiak(8) := 13; Lista1.zenbakiak(9) :=
25;
  Lista1.zenbakiak(10) := 7;
  Lista1.kopurua := 10;
  Put_line("proba3: hainbat zenbaki ez ordenatuez osatutako bektorea
(erreserbatu dugun guztia beteta): (3 1 4 8 6 23 2 13 25 7)");
```

```

    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_burbuiaz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    New_Line(3);
    Put_Line("Pulsa return para continuar");
    Skip_Line;
    New_Line(3);

```

```

Lista1.zenbakiak(1) := 1; Lista1.zenbakiak(2) := 3; Lista1.zenbakiak(3) := 5;
Lista1.zenbakiak(4) := 7; Lista1.zenbakiak(5) := 9;
Lista1.kopurua:= 5;
Put_line("proba4: lista ordenatua: (1 3 5 7 9)");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_burbuiaz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");

```

```

New_Line(3);
Put_Line("Pulsa return para continuar");
Skip_Line;
New_Line(3);

```

```

Lista1.kopurua:= 0;
Put_line("proba5: lista hutsa: ( )");
    Put_line(" sarrerako lista: ");
    Idatzi_Lista(Lista1);
    Put_line(" da ");
    put_line(" eta zure emaitza ");
    ordenatu_burbuiaz(Lista1);
    Idatzi_Lista(Lista1);
    Put_line(" da ");

```

```

New_Line(3);
Put_Line("Pulsa return para continuar");
Skip_Line;
New_Line(3);

```

end Proba\_ordenatu\_burbuiaz;

ordenatu\_burbuiaz.adb

```

with defs_lab;
use defs_lab;

```

procedure ordenatu\_burbuiaz (t: in out osokoen\_zerrenda) is

```

    aux, num_pasatakoak, i: Integer;
    ez_ordenatua: boolean;
begin
    num_pasatakoak:=1;
    loop exit when ((num_pasatakoak > t.kopurua-1) or (ez_ordenatua = false));
        i:=1;
        ez_ordenatua:=false;
        loop exit when i > t.kopurua-1; --5. Elementua ez atzitzeko, ez baitago
            if t.zenbakiak(i) > t.zenbakiak(i+1) then ---aldatu
                aux:=t.zenbakiak(i);
                t.zenbakiak(i):=t.zenbakiak(i+1);
                t.zenbakiak(i+1):=aux;
                ez_ordenatua:=true;
            end if;
            i:=i+1;
        end loop;
        num_pasatakoak:= num_pasatakoak+1;
    end loop;
end ordenatu_burbuiatz;

```