with Ada.Strings.Unbounded;         use Ada.Strings.Unbounded;

with Registre;                      use Registre;

procedure test\_registre is

    R:T\_Access;

    d1,d2,d3:T\_Date;

    procedure Exemple\_Registre(*R*: in out T\_Access) is

        begin

            Start\_RG(18,R);

            pragma assert(not Est\_Vide\_RG(R));

            AddKey(2,R);

            AddKey(15,R);

            AddKey(1,R);

            AddKey(5,R);

            AddKey(4,R);

            AddKey(19,R);

            AddKey(33,R);

            AddKey(25,R);

            AddKey(42,R);

            AddKey(35,R);

        end Exemple\_Registre;

    procedure Tester\_Exemple\_Registre is

        begin

            Exemple\_Registre(R);

            --Tester l'existence des éléments ajoutés dans le registre.

            pragma assert(Existe\_RG(2,R));

            pragma assert(Existe\_RG(15,R));

            pragma assert(Existe\_RG(1,R));

            pragma assert(Existe\_RG(5,R));

            pragma assert(Existe\_RG(4,R));

            pragma assert(Existe\_RG(19,R));

            pragma assert(Existe\_RG(33,R));

            pragma assert(Existe\_RG(25,R));

            pragma assert(Existe\_RG(42,R));

            pragma assert(Existe\_RG(35,R));

            --Tester la modification.

            ModifyKey(42,49,R);

            pragma assert(Existe\_RG(49,R));

            pragma assert(not Existe\_RG(42,R));

            ModifyKey(35,40,R);

            pragma assert(Existe\_RG(40,R));

            pragma assert(not Existe\_RG(35,R));

            --Tester la suppresion

            Delete\_RG(5,R);

            pragma assert(not Existe\_RG(5,R));

            Delete\_RG(40,R);

            pragma assert(not Existe\_RG(40,R));

            --Tester les différents attributs du registre.

            AddName(18,To\_Unbounded\_String("Younes Saoudi"),R);

            AddName(19,To\_Unbounded\_String("Latifa Achour"),R);

            AddName(2,To\_Unbounded\_String("ElHassan Saoudi"),R);

            pragma assert(Name(18,R)=To\_Unbounded\_String("Younes Saoudi"));

            pragma assert(Name(19,R)=To\_Unbounded\_String("Latifa Achour"));

            pragma assert(Name(2,R)=To\_Unbounded\_String("ElHassan Saoudi"));

            AddBirthP(19,To\_Unbounded\_String("Oujda"),R);

            AddBirthP(2,To\_Unbounded\_String("Taza"),R);

            AddBirthP(18,To\_Unbounded\_String("Meknès"),R);

            pragma assert(BirthP(19,R)=To\_Unbounded\_String("Oujda"));

            pragma assert(BirthP(2,R)=To\_Unbounded\_String("Taza"));

            pragma assert(BirthP(18,R)=To\_Unbounded\_String("Meknès"));

            AddBirthD(19,18,MAI,1966,R);

            AddBirthD(18,18,NOVEMBRE,1999,R);

            AddBirthD(2,16,DECEMBRE,1960,R);

            CreateDate(d1,18,MAI,1966);

            CreateDate(d2,18,NOVEMBRE,1999);

            CreateDate(d3,16,DECEMBRE,1960);

            pragma assert(BirthD(19,R)=d1);

            pragma assert(BirthD(18,R)=d2);

            pragma assert(BirthD(2,R)=d3);

        end Tester\_Exemple\_Registre;

    begin

        Tester\_Exemple\_Registre;

    end test\_registre;