

```

with Ada.Strings.Unbounded;           use Ada.Strings.Unbounded;
with arbre_genealogique;
package Registre is

    TYPE T_Registre is LIMITED PRIVATE;
    TYPE T_Access is LIMITED PRIVATE;
    TYPE T_Mois is (JANVIER, FEVRIER, MARS, AVRIL, MAI, JUIN, JUILLET,AOUT, SE
PTEMBRE, OCTOBRE, NOVEMBRE, DECEMBRE);
    TYPE T_Date is private;

    --FONCTIONS/PROCEDURES SECONDAIRES--

    --Créer/modifier une date.
    procedure CreateDate(D: in out T_Date; Jour: in Integer; Mois: in T_Mois;
Annee:in Integer);
    --Afficher unde date.
    procedure Afficher_Date (Date : in T_Date);

    --TESTS--
    --Initialiser un Registre vide
    procedure Init_RG(RG: in out T_Access);
    --Initialiser un Registre.
    procedure Start_RG(Cle: in Integer; Reg: in out T_Access);
    --Vérifier si un registre est vide.
    function Est_Vide_RG(Reg: in T_Access) return Boolean;
    --Vérifier si une clé existe dans le registre.
    function Existe_RG(Cle: in Integer;Reg: in T_Access) return Boolean;

    --FONCTIONS/PROCEDURES ELEMENTAIRES--

    --Retourne le nom d'un clé.
    function Name(Cle: in Integer; Reg:in T_Access) return Unbounded_String;
    --Retourne la date de naissance d'une clé.
    function BirthD(Cle: in Integer; Reg: in T_Access) return T_Date;
    --Retourne l'année de naissance d'une clé.
    function BirthY(Cle: in Integer; Reg: in T_Access) return Integer;
    --Retourne le lieu de naissance d'une clé.
    function BirthP(Cle: in Integer; Reg: in T_Access) return Unbounded_String
;

    --Multiplie toutes les clés du registre par 10.
    procedure RG_Multiplier_10(RG: in out T_Access);

    --FONCTIONS/PROCEDURES DE RECHERCHE--

```

```

--Rechercher le registre d'une clé.
function Rech_Reg(Cle: in Integer; Reg: in T_Access) return T_Access;

--AJOUT/SUPPRESSION--

--
Ajouter une clé au registre. L'ajout sera dans l'ordre croissant pour faciliter l'accès.
procedure AddKey(Cle: in Integer; Reg: in out T_Access);
--Supprimer une clé.
procedure Delete_RG(Cle: in Integer; Reg: in out T_Access);
--Détruire le registre.
procedure Detruire_RG(RG: in out T_Access);

--MODIFICATIONS--

--Modifier la valeur d'une clé.
procedure ModifyKey(Cle: in Integer; NewCle: in Integer; Reg: in out T_Access);
--Attribuer/Modifier un nom COMPLET à une clé.
procedure AddName(Cle: in Integer; Nom: in Unbounded_String; Reg: in out T_Access);
--
Attribuer/Modifier une date de naissance à une clé. L'age sera attribué automatiquement.
procedure AddBirthD(Cle: in Integer; Jour: in Integer; Mois: in T_Mois; Annee: in Integer; Reg: in out T_Access);
--Attribuer/Modifier un lieu de naissance à une clé.
procedure AddBirthP(Cle: in Integer; Lieu: in Unbounded_String; Reg: in out T_Access);
--Ajouter un conjoint à une clé.
procedure Ajouter_Conjoint(Cle, Conjoint: in Integer; RG: in out T_Access);
--#####SECONDE PARTIE#####

PRIVATE
    TYPE T_Date is
        RECORD
            Jour : Integer;
            Mois : T_Mois;
            Annee : Integer;
        end RECORD;
    TYPE T_Access is ACCESS T_Registre;
    TYPE T_Registre is
        RECORD
            Cle: Integer;
            Nom_Complet: Unbounded_String;

```

```
        Date_Naissance: T_Date;
        Lieu_Naissance: Unbounded_String;
        Age: Integer;
        Conjoint: arbre_genealogique.Arbre_Binaire_Character.Piles_Cle.T_
Pile;
        Suivant: T_Access;
    end RECORD;
end Registre;
```