

```

-----
-- Nom fichier      : Manipuler_Tableau.adb          --
-- Auteur           : R.                            --
-- Date             : 26/10/2008                    --
--                                                         --
-- But              : Manipulation de tableau        --
--                                                         --
-- Date de modif.   :                               --
-- Raison           :                               --
-- Modules appeles  : Ada.Numerics.Float_Random      --
--                 : Generateur de nombres aleatoires entre 0.0 et 1.0 --
-- Mat. particulier :                               --
--                                                         --
-- Compilation      : GNAT                          --
-- Edition de liens : GNAT                          --
-- Mode d'execution : Console                       --
-----

with Ada.Text_IO;           -- d'entree/sortie de texte
with Ada.Integer_Text_IO;  -- entree/sortie d'entiers
with Ada.Float_Text_IO;   -- entree/sortie de reel
with Ada.Numerics.Float_Random; -- Module generateur aleatoire
use Ada.Text_IO;
use Ada.Integer_Text_IO;
use Ada.Float_Text_IO;
use Ada.Numerics.Float_Random;

-- Procedure principale
procedure Manipuler_Tableau is

-- valeur max des nombres generes,
Valeur_Max : constant := 200.0;
Longueur_Max : constant := 50;

-- Type T_Vecteur d'entiers compris entre 1 et integer(Valeur_Max)= 100
type T_Vecteur is array (Natural range <> ) of Natural range 0..Natural(Valeur_Max);

-- Specifications des fonctions et des procedures utilisees
-- *****
procedure Presentation;
function Saisir(Taille_Maximale : Integer) return Integer;
procedure Afficher_Vecteur(Vecteur : in T_Vecteur);
procedure Remplir_Vecteur(Genrateur : in Generator;
                          ValeurMax : in Float;
                          Vecteur : out T_Vecteur);
function Calculer_Moyenne (Vecteur : T_Vecteur) return Float;
-- *****

```

```
-----  
-- Nom          : Presentation  
-- But          :  
-- Parametres   :  
-----
```

```
procedure Presentation is  
begin  
  Put_line("*****");  
  Put_Line("* Programme qui calcule la moyenne du plus petit et du *");  
  Put_Line("* plus grand element d'un tableau rempli avec aleatoirement*");  
  Put_line("*****");  
end Presentation;
```

```
-----  
-- Nom          : Saisir  
-- But          :  
-- Parametres   :  
-- Exception    :  
-----
```

```
function Saisir(Taille_Maximale : Integer) return Integer is  
  Longueur : Integer;  
begin  
  loop  
    Put (" Entrer la taille maximale de votre tableau compris entre 0-"  
         &Integer'Image(Longueur_Max) &" : ");  
    Get(Longueur);  
    Skip_Line;  
    if (Longueur <= Taille_Maximale) and (Longueur > 0) then  
      return Longueur;  
    end if;  
  end loop;  
end Saisir;
```

```
-----  
-- Nom          : Remplir_Vecteur  
-- But          :  
-- Parametres   :  
-- Exception    :  
-----
```

```
procedure Remplir_Vecteur(Genrateur : in Generator;  
                          ValeurMax : in Float;  
                          Vecteur   : out T_Vecteur) is  
begin  
  for I in Vecteur'range loop  
    Vecteur(I) := Natural(Random(Genrateur) *ValeurMax);  
  end loop;  
end Remplir_Vecteur;
```

```
-----  
-- Nom          : Afficher_Vecteur  
-- But          :  
-- Parametres   :  
-- Exception    :  
-----
```

```
procedure Afficher_Vecteur(Vecteur : in T_Vecteur) is  
begin  
    New_Line;  
    Put_Line(" Le vecteur genere est : ");  
    Put(" [");  
    for I in Vecteur'range loop  
        Put(Vecteur(I),4);  
    end loop;  
    Put("]");  
    New_Line(1);  
end Afficher_Vecteur;
```

```
-----  
-- Nom          : Calculer_Moyenne  
-- But          :  
-- Parametres   :  
-- Exception    :  
-----
```

```
function Calculer_Moyenne (Vecteur : T_Vecteur) return Float is  
    Minimum : Natural := Vecteur (Vecteur'First(1));  
    Maximum : Natural := Vecteur (Vecteur'First(1));  
begin -- Moyenne  
    -- Parcours de la matrice pour trouver le minimum et le maximum  
    -- Traiter toutes les lignes de la table  
    for Indice in Vecteur'Range(1) loop  
        if Vecteur(Indice) < Minimum then  
            Minimum := Vecteur(Indice);  
        elsif Vecteur(Indice) > Maximum then  
            Maximum := Vecteur(Indice);  
        end if;  
    end loop;  
    -- Calcule et livrer la moyenne  
    return Float(Minimum + Maximum) / 2.0;  
end Calculer_Moyenne;
```

```
-----  
-- Declaration de variables  
-----
```

```
-- variables : dimensions du tableau, generateur aleatoire  
Gen          : Generator ;
```

```
-- variable mon choix  
Mon_Choix    : Character := 'O'; -- variable mon choix
```

```
-- Corps de la procedure principale
```

```
begin -- Manipuler_Tableau;
```

```
  Presentation; -- presentation de l'application pour l'utilisateur
```

```
  Reset(Gen,0); -- appel de la procedure pour initialiser le generateur avec la valeur 0
```

```
  while (Mon_Choix = 'O' or Mon_Choix = 'o') loop
```

```
    Calcul : declare          -- declarer un bloc pour tableau dynamique
```

```
      Vect : T_Vecteur(1..Saisir(Longueur_Max)) ;
```

```
    begin
```

```
      Remplir_Vecteur(Gen, Valeur_Max, Vect);
```

```
      Afficher_Vecteur(Vect);
```

```
      Put("la moyenne du plus petit et du plus grand element du tableau est : ");
```

```
      Put(Calculer_Moyenne(Vect),0,0,1);
```

```
      New_Line(1);
```

```
    end Calcul;
```

```
    New_Line;
```

```
    Put( " voulez vous continuer ? tapez [O/N] : " );
```

```
    Get(Mon_Choix);
```

```
    Skip_Line;
```

```
  end loop;          -- boucle de repetition
```

```
end Manipuler_Tableau; -- fin programme
```