


```

pragma Inline (Tri_Par_Insertion);
§-111111111
procedure Tri_Par_Insertion ( A : in out Vecteur) is
§-111111111
§ § I, J      : Index range A' RANGE;
§ § T      : Élément;      -- type de la variable temporaire est générique
§ § L      : Index range A' RANGE:= A' FIRST;
§ § U      : Index range A' RANGE:= A' LAST;
§ begin
§   I := L;
§   while I /= U loop
§ §   T := A(Index' Succ(I));
§ §   J := Index' Succ(I);
§ §   while J /= L and then A(Index' Pred(j)) > T loop
§ § 5   A(J) := A(Index' Pred(J));
§ § 5   J := Index' Pred(J);
§ § 5   end loop;
§ §   A (J) := T;
§ §   I := Index' Succ(I);
§ § end loop;
§ end Tri_Par_Insertion;
§
§-111111111
generic
§ type Élément      is private;
§ type Index        is ( <> );
§ type Vecteur      is array (Index range <>) of Élément;
§ with procedure Échanger ( A, B : in out Élément );
§ with function ">" (A, B : in Élément) return Boolean is <>;
§ procedure Tri_En_Bulle ( X : in out Vecteur);

```



```

§ § if Last : Index := X' Last;
§ § 111111111
§ procedure Échanger ( A, B : in out Élément );
§ § a111111111
§ pragma Inline ( Échanger );
§ § 111111111
§ procedure Échanger ( A, B : in out Élément ) is
§ § a111111111
§ § if Temp : Élément;
§ § begin
§ §   Temp := A;
§ §   A := B;
§ §   B := Temp;
§ § end Échanger;
§ begin
§   for I in reverse X' First .. Index' Pred(Index(X' Last)) loop
§ §   for J in Index' First .. I loop
§ § 5 7 if X(J) > X(Index' Succ(J)) then
§ § 5 5 4 Échanger ( X(J), X(Index' Succ(J)));
§ § 5 5 end if;
§ § 5 end loop;
§ § end loop;
§ end Tri_En_Bulle2;
§
§ if type Couleur_Type is ( Rouge, Jaune, Vert, Bleu, Magenta, NavajoRed, PapayaWhip);
§ if type Tableau_Couleur_Type is array ( Couleur_Type range <> ) of Couleur_Type;
§ if A : Tableau_Couleur_Type := ( Magenta, Vert, NavajoRed, Jaune, Rouge, Bleu,
PapayaWhip );
§ § 111111111
§ procedure Échanger_Couleur is new Échanger ( Couleur_Type );

```

```

§a11111111
§_11111111
§0 procedure Tri_Couleur is new Tri_En_Bulle ( Élément => Couleur_Type,
Index      => Couleur_Type,
Vecteur    => Tableau_Couleur_Type,
Échanger   => Échanger_Couleur);
§§
§a11111111
§_11111111
§0 procedure Put (A : in Tableau_Couleur_Type) is
§_11111111
§§
§§ begin
§_11111111 for i in A' Range loop
§§ 71 Put_Line ( Couleur_Type' Image(A(i)));
§§ end loop;
§ end Put;
§
§i type Tableau_Caractère_Type is array ( Integer range <> ) of Character;
§i B : Tableau_Caractère_Type := ('z', 'B', 'r', 'e', 'p', 'A', '>', '=', NUL,
§      'C', 'Q', 'à', 'é', 'è', 's', ')', '(', '0',
§      ' ', 'k', ACK);
§_11111111
§0 procedure Tri_Caractère is new Tri_En_Bulle2 ( Character, Integer,
Tableau_Caractère_Type);
§a11111111
§_11111111
§0 procedure Put (A : in Tableau_Caractère_Type) is
§_11111111
§§ XXXXXX
§0 package Int_I0 is new Ada.Text_I0.Integer_I0 ( Integer );

```

```

§ §E|||||||
§ § use Int_IO;
§ §begin
§ §for i in A'Range loop
§ § 71Set_Col ( 2 ); Put ( Character'Pos(A(i)), 5, 16); -- Code hexadecimal
§ § 71Set_Col ( 10); Put ( A ( i)); -- le caractère
§ § 71Set_Col(20); Put_Line ( Character'Image(A(i))); -- le littéral énumératif
§ §end loop;
§ §end Put;
§
§
§i type Étudiant_Type is
§ record
§ Matricule : String ( 1 .. 6 );
§ Nom : String ( 1 .. 20 );
§ Prénom : String ( 1 .. 15);
§ end record;
§i type Tableau_Étudiant_Type is array ( Integer range <> ) of Étudiant_Type;
§i Étudiants : Tableau_Étudiant_Type := (
§ ("123456", "Bleau", "Jos"),
§ ("163234", "Larivée", "Pierre"),
§ ("043562", "Doe", "John"));
§_11111111
§§function Comparer_Matricule ( A, B : in Étudiant_Type ) return Boolean is
§_11111111
§ §begin
§ §return A.Matricule > B.Matricule;
§ §end Comparer_Matricule;
§_11111111
§§function Comparer_Nom ( A, B : in Étudiant_Type ) return Boolean is
§_11111111

```

```

§ §
§ begin
§ return A.Nom > B.Nom;
§ end Comparer_Nom;
§_11111111
§ procedure Put ( A : in Tableau_Étudiant_Type ) is
§_11111111
§ § use Ada.Strings.Fixed;
§ § use Ada.Strings;
§ begin
§ for i in A'Range loop
§ § 71Set_Col ( 5 ); Put ( A(i).Matricule ); -- le matricule
§ § 71Set_Col ( 14); Put ( Trim (A(i).Nom, Right)); -- le nom, les espaces à droite
enlevés,
§ § 71Ada.Text_IO.Put ( ", "); Put ( Trim (A(i).Prénom, Right)); -- suivi d'une , et
le prénom
§ § 71New_Line;
§ § end loop;
§ end Put;
§_11111111
§ procedure Échanger_Étudiant is new Échanger ( Étudiant_Type );
§_11111111
§_11111111
§ procedure Tri_Par_Matricule is new Tri_En_Bulle ( Élément => Étudiant_Type,
§ § Index => Integer,
§ § Vecteur => Tableau_Étudiant_Type,
§ § Échanger => Échanger_Étudiant,
§ § ">" => Comparer_Matricule );
§_11111111
§_11111111

```

