

*L'objectif de ce TP est de manipuler les **vecteurs contraints**. Pour cela, il est indispensable de comprendre et maîtriser la déclaration d'un type vecteur contraint et l'utilisation d'une variable de ce type à l'aide des attributs *First*, *Last*, *Range*, *Length*. Voir polycopié pages 23 à 27.*

Contexte applicatif pour ce TP

Le taux de pollution atmosphérique en dioxyde d'azote se mesure en $\mu\text{g}/\text{m}^3$. On souhaite écrire un programme qui permet de manipuler les 12 taux moyens de chaque mois d'une année (pour un lieu donné).

1. Vecteur contraint indicé par des entiers

Nous allons écrire les différentes déclarations, fonctions et procédures dans un paquetage **p_pollution** que nous utiliserons dans un programme principal **atmo**.

1.1. Ecrire un fichier de spécifications **p_pollution.ads**, contenant (entre autres) les déclarations des types **T_Pollution** et **TV_Pollution** et les en-têtes des procédures **Saisie** et **Affichage** suivantes :

```
subtype T_Pollution is float range 0.0..300.0;
type TV_Pollution is array(1..12) of T_Pollution;

procedure Saisie (V : out TV_Pollution);
-- {} => {V contient les 12 valeurs saisies par l'utilisateur}

procedure Affichage (V : in TV_Pollution);
-- {} => {Affiche le contenu du vecteur V sur plusieurs lignes de la forme :
        Taux moyen de pollution du mois n° ... = ... }
```

Compiler **p_pollution.ads** et corrigez-le si nécessaire.

Remarque : On ne peut utiliser les procédures **Lire** et **Ecrire** de **P_Esiut** que pour des variables de type **Integer**, **Float**, **Character** et **String**. Il faut donc créer l'équivalent de ces procédures pour les vecteurs de type **TV_Pollution**. C'est pour cela que vous développez ici une procédure (**Saisie**) pour lire tous les éléments du vecteur et une procédure (**Affichage**) pour les afficher.

1.2. Copiez le fichier **p_pollution.ads** dans **p_pollution.adb** et complétez le corps du paquetage.

Rappel : Pensez ici à utiliser au maximum les attributs des vecteurs (**First**, **Last**, **Range**, **Length**).

Compilez **p_pollution.adb** et corrigez-le si nécessaire.

1.3. Pour tester votre paquetage, vous allez écrire, dans un fichier **atmo.adb**, un programme principal **atmo** qui saisit puis affiche un vecteur de type **TV_Pollution**.

Compilez **atmo.adb**, puis testez-le.

2. Fonctions et procédures supplémentaires

Rajoutez dans votre paquetage les fonctions et procédures suivantes. Vous devrez les compiler une par une et rajouter dans le programme principal les instructions nécessaires qui permettront de les tester au fur et à mesure.

2.1. Une fonction `NbMoisPollues` qui calcule le nombre de mois pollués (c'est à dire dépassant la valeur de 60 $\mu\text{g}/\text{m}^3$) d'un vecteur donné.

```
function NbMoisPollues (V : in TV_Pollution) return integer;
-- {} => {résultat = nombre de mois où la pollution dépasse 60}
```

2.2. Une procédure `Extremes` qui donne le **taux** de pollution du mois le moins pollué et celui du mois le plus pollué, ainsi que les **indices** de ces mois.

```
procedure Extremes (V : in TV_Pollution;
                   Min, Max : out T_Pollution; IndMin, IndMax : out integer);
-- {} => {Min est le taux de pollution minimum, obtenu pour le mois IndMin,
--       Max est le taux de pollution maximum, obtenu pour le mois IndMax}
```

2.4. Une fonction `MoinsPollue` qui renvoie vrai si un lieu `v1` est moins pollué qu'un lieu `v2`. Pour cela, on compare les vecteurs mois par mois : pour le mois numéro 1, on regarde si le taux de pollution dans `v1` est plus faible que celui dans `v2` ; puis pour le mois numéro 2 ... On compte finalement le nombre de fois où le taux dans `v1` est plus faible que le taux dans `v2`.

```
function MoinsPollue (V1, V2 : in TV_Pollution) return boolean;
-- {} => {résultat = vrai si V1 est moins pollué que V2}
```

Pour tester cette fonction, vous comparerez dans le programme : un vecteur initialisé par agrégat à un autre vecteur saisi au clavier.

3. Extension : Vecteur contraint indicé par un type énuméré

Dupliquez vos 3 fichiers pour créer le paquetage `p_pollution_bis` et le programme `atmo_bis`.

Modifiez vos fichiers de façon à ce que les indices des vecteurs ne soient plus des entiers de 1 à 12 mais des valeurs de type énuméré de Janvier à Decembre (sans accent). Pour cela, vous allez déclarer le type énuméré suivant :

```
type T_IndiceMois is (Janvier, Fevrier, ... , Decembre);
```

Si vous avez bien programmé, vous n'avez qu'à modifier la déclaration du type des vecteurs, la déclaration de certaines variables et la gestion des entrées/sorties.