

## TP n°3 : Chaînages et météo

### 1 Objectif

On dispose de fichiers de relevés provenant de stations météorologiques de la forme ci-dessous. Les cinq colonnes correspondent aux informations suivantes : le numéro de la station (un entier positif), l'année du relevé, le numéro du jour dans l'année (un entier entre 1 et 365 ou 366 si l'année est bissextile), l'heure du jour (un entier entre 0 et 23) et la température mesurée en degrés Celsius. Chaque fichier correspond à une année complète de relevés pour une même station.

3	2008	1	0	-5.66667
3	2008	1	1	-9
3	2008	1	2	-13.5667
3	2008	1	3	-9.9
3	2008	1	4	-5.1
...				
3	2008	366	19	15.8754
3	2008	366	20	15.7366
3	2008	366	21	12.0699
3	2008	366	22	7.63934
3	2008	366	23	4.73661

L'objectif est de réaliser des traitements sur ces fichiers de données au moyen de chaînages simples. Des fichiers de relevés sont disponibles au sein de l'outil `madoc`. Le code pour lire un fichier est donné ci-dessous, sachant qu'il est nécessaire d'inclure les *headers* `fstream` et `sstream`.

```
int station, annee, jour, heure;
double temperature;
std::ifstream ifs(filename);    // filename est le nom du fichier de relevés
if (ifs) {
    while (getline(ifs, ligne)) {
        std::istringstream iss(ligne);
        iss >> station;
        iss >> annee;
        iss >> jour;
        iss >> heure;
        iss >> temperature;
        ...    // ici traitement du relevé qui vient d'être lu dans le fichier
    }
}
```

### 2 Travail

#### 2.1 Complexité en mémoire

Quel est le nombre d'entiers et de réels présents dans un fichier de relevés tel que décrit ci-dessus ? Certaines informations sont-elles redondantes ? Proposer un format de fichier permettant de stocker un nombre minimum de données.

## 2.2 Date

Définir un type `DateMeteo` pour représenter une date (*année, jour, heure*) apparaissant dans un fichier de relevés. Rappelons qu'une année est bissextile si elle est divisible par 4 et non divisible par 100 ou bien si elle est divisible par 400. Ce type sera associé aux fonctions suivantes :

1. `initialiserDateMeteo(...)` : initialise une date ;
2. `comparerDateMeteo(...)` : compare deux dates et retourne 0 si les dates sont égales, un entier négatif si la première date précède la seconde ou un entier positif sinon.
3. `afficherDateMeteo(...)` : écrit une date dans un flot de sortie au même format que dans un fichier de relevés.

## 2.3 Relevé

Définir un type `ReleveMeteo` pour représenter un relevé de la forme  $(s, d, t)$  où  $s$  est une station,  $d$  est une date et  $t$  est une température. Ce type sera associé aux fonctions suivantes :

1. `initialiserReleveMeteo(...)` : initialise un relevé ;
2. `afficherReleveMeteo(...)` : écrit un relevé dans un flot de sortie au même format que dans un fichier de relevés.

La comparaison de deux relevés  $r_1 = (s_1, d_1, t_1)$  et  $r_2 = (s_2, d_2, t_2)$  consiste à définir une relation d'ordre totale  $<$  permettant de dire si  $r_1 = r_2$ ,  $r_1 < r_2$  ou  $r_2 < r_1$ . Par exemple, définissons une telle relation de la manière suivante

$$r_1 < r_2 \iff (t_1 < t_2) \text{ ou } (t_1 = t_2 \text{ et } (d_1 > d_2 \text{ ou } (d_1 = d_2 \text{ et } s_1 < s_2)))$$

consistant à ordonner les relevés par températures croissantes, puis par dates décroissantes à températures égales, puis par stations croissantes à températures et dates égales. Le type des fonctions de comparaison sera défini comme suit :

```
typedef int (* CompReleveMeteo)(const ReleveMeteo& rm1,
                                const ReleveMeteo& rm2);
```

3. Implémenter une fonction de comparaison réalisant la relation d'ordre décrite ci-dessus.

Enfin, on définit le type des prédicats sur les relevés, c'est-à-dire des fonctions prenant en paramètre un relevé et retournant un booléen représentant la valeur de vérité d'une propriété.

```
typedef bool (*PredicatReleveMeteo)(const ReleveMeteo& rm);
```

## 2.4 Chaînage

Définir le type `Chainage` des chaînages simples de relevés météorologiques de type `ReleveMeteo`, avec accès aux maillons de tête et de queue en temps constant. Ce type sera associé aux fonctions suivantes :

1. `initialiserChainage(...)` : initialise un chaînage ;
2. `estVide(...)` : teste si un chaînage est vide, i.e., ne contenant aucun élément ;
3. `reinitialiser(...)` : réinitialise un chaînage en retirant tous ses éléments ;
4. `insérerTete(...)` : insère un élément en tête du chaînage ;
5. `insérerQueue(...)` : insère un élément en queue du chaînage ;

6. `retirerTete(...)` : retire l'élément de tête d'un chaînage ;
7. `retirerQueue(...)` : retire l'élément de queue d'un chaînage ;
8. `insérerOrdre(...)` : insère un élément dans un chaînage dans l'ordre donné par une fonction de comparaison de type `CompReleveMeteo` (passée en paramètre) ;
9. `plusPetitElement(...)` : retourne l'élément le plus petit selon l'ordre donné par une fonction de comparaison de type `CompReleveMeteo` (passée en paramètre) ;
10. `retirerTous(...)` : retire tous les éléments d'un chaînage vérifiant une propriété donnée par un prédicat de type `PredicatReleveMeteo` (passé en paramètre) ;
11. `afficherChainage(...)` : écrit les éléments d'un chaînage sur un flot de sortie.

## 2.5 Opérations spécifiques aux chaînages de relevés

On souhaite réaliser les traitements suivants sur chaque fichier de relevés après avoir créé un chaînage contenant tous les relevés du fichier :

1. Afficher les relevés correspondants aux températures minimum et maximum dans l'année en utilisant la fonction `plusPetitElement(...)` et les fonctions de comparaison adaptées.
2. `calculerTemperatureMoyenne(...)` : calcule la température moyenne ;
3. `calculerEcartMoyenTemperature(...)` : si le chaînage contient les températures consécutives  $t_1, t_2, \dots, t_k$  en supposant le chaînage trié par dates croissantes, calcule l'écart moyen entre deux températures consécutives par la formule suivante :

$$\frac{1}{k-1} \times \sum_{i=2}^k |t_i - t_{i-1}|$$

4. `retirerDefauts(...)` : retire les éléments d'un chaînage (sauf la tête et la queue) tels que l'écart de température avec les éléments précédents et suivants est supérieur à un écart maximum autorisé (passé en paramètre) ;
5. Appliquer la fonction précédente telle que l'écart maximum autorisé soit égal à deux fois l'écart moyen. Puis supprimer les éléments en tête et en queue.
6. À nouveau, afficher les relevés correspondants aux températures minimum et maximum, calculer la température moyenne, et observer les différences par rapport aux résultats produits aux questions 1 et 2.
7. En considérant tous les fichiers donnés (une même station et plusieurs années consécutives) observe t-on un réchauffement climatique en moyenne des températures sur une année ? Quelle est l'année la plus chaude ? Existe t-il un pic ?

## 3 Réalisation du TP et rendu

Ce travail est à réaliser de préférence en binôme ou éventuellement de manière individuelle. Il sera encadré pendant trois séances de TP.

Il faudra rendre les programmes sources et un petit rapport exposant les résultats obtenus au plus tard le jour de la troisième séance de TP à 23h59 selon les modalités fixées par chaque chargé de TP.