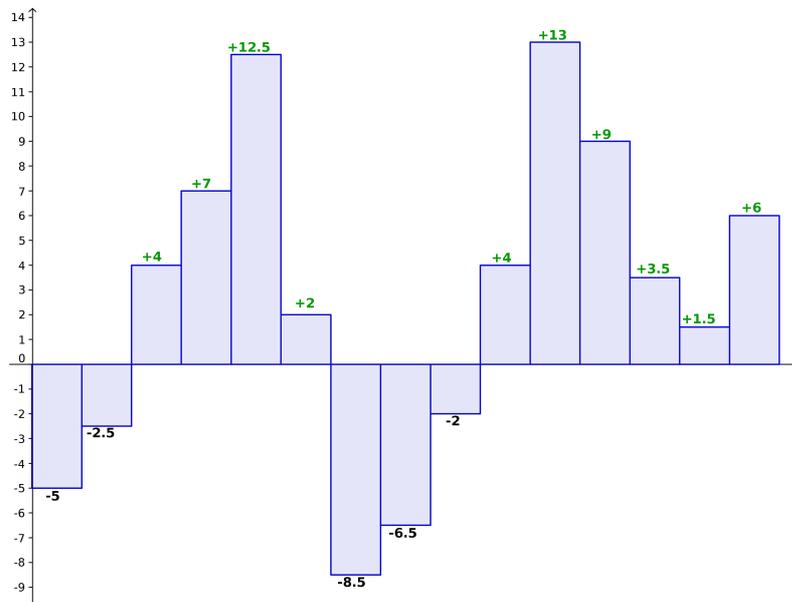


TP5 – TABLEAUX À UNE DIMENSION

Le but de ce TP est de se familiariser avec la notion de tableaux et de renforcer les notions sur les sous-programmes.

Les différentes parties du TP sont basées sur le même thème : l'analyse d'une série de mesures de température. Ces mesures constituent les éléments d'un tableau sur lequel les traitements seront réalisés à l'aide de sous-programmes tels que la saisie des valeurs du tableau, l'affichage de ces valeurs, la recherche des extremums, la recherche de la valeur la plus proche de zéro et enfin le calcul des effectifs de plusieurs classes de températures.

Voici un exemple de représentation d'une série de mesures de températures.



- Le nombre des mesures à analyser est compris entre 0 inclus et 100 exclu
- Chaque mesure est comprise entre $-273,5^{\circ}\text{C}$ (zéro absolu) et $5526,0^{\circ}\text{C}$ (estimation de la température à la surface du soleil).

Saisie et affichage d'une série de valeurs

Il s'agit de réaliser un programme qui permet de lire N mesures de température et de les afficher de deux manières différentes : toutes les valeurs sur une ligne unique ou une valeur par ligne. Le programme principal doit présenter un menu sous la forme suivante (comme dans le TP2) :

Voulez-vous :

- (1) entrer des températures
- (2) afficher la liste des températures
- (3) quitter

Votre choix ?

Préparation

1. Préparer une série d'une quinzaine de mesures de température. Il est possible d'utiliser le fichier Geogebra : `analyse_temperatures.ggb` disponible sur Moodle pour valider les calculs statistiques à réaliser dans la suite du TP.

2. Préparer la déclaration des constantes correspondant au nombre maximum de mesures `N_MAX` et aux valeurs extrêmes d'une mesure `T_MIN` et `T_MAX`.

Préparer la déclaration du type `t_mes` correspondant au tableau de `N_MAX` mesures

3. Écrire l'algorithme du sous-programme `lireTemp()` qui commence par demander à l'utilisateur le nombre de mesures à analyser puis les valeurs de chaque mesure. Ces valeurs sont enregistrées dans un tableau de type `t_mes`.

Les valeurs saisies par l'utilisateur doivent être vérifiées. Le nombre de mesure doit être compris entre 0 inclus et 100 exclu. Les mesures doivent être comprises entre $-273,5^{\circ}\text{C}$ inclus et 5526°C inclus.

Le sous-programme doit renvoyer le tableau des mesures saisies et le nombre de mesures au programme principal.

4. Écrire l'algorithme du sous-programme `afficherTemp()` qui affiche la série de mesures sous deux formes différentes : toutes les mesures sur une seule ligne ou une mesure par ligne.

Le sous-programme reçoit en entrée le tableau des mesures, le nombre des mesures à afficher et un booléen indiquant si l'affichage doit oui ou non se faire sur une ligne unique.

5. Écrire l'algorithme du programme principal dont le rôle est de présenter le menu décrit ci-dessus et d'appeler les deux sous-programmes précédents.

Reprendre le travail effectué lors du TP2 en utilisant une variable de type entier pour la sélection des différents choix disponibles.

Dans le cas où le nombre de mesures de température est égal à zéro après l'appel à `lireTemp()`, on affiche un message à l'utilisateur indiquant que le traitement ne peut pas se faire.

Réalisation en séance

1. Ouvrir, dans **Code::Blocks**, un nouveau projet ainsi que le fichier `modele.cpp` qui sera renommé en `analysetemp.cpp`.

2. Coder le programme principal puis tester son fonctionnement en remplaçant les appels de sous-programmes par l'affichage du nom du sous-programme concerné.

3. Compléter le programme précédent en y intégrant les deux sous-programmes `lireTemp()` et `afficherTemp()`.

Analyse de la série de mesures de température

Dans cette partie, on aborde l'analyse de la série de mesures de températures avec la recherche des extremums, la recherche de la température la plus proche de zéro et enfin le calcul des effectifs par classes.

Relativement à la partie précédente, le menu du programme principal doit être complété. En version finale, on doit disposer des choix suivants :

```
Voulez-vous :
(1) entrer des températures
(2) afficher la liste des températures
(3) afficher la température la plus proche de 0
(4) afficher les extremums
(5) afficher les effectifs des 4 classes
    ]-10, -5] ]-5, 0] ]0, 5] ]5, 10]
(6) quitter
Votre choix ?
```

Préparation

1. Écrire l'algorithme du sous-programme `minmaxTemp()` dont le but est de trouver la température minimum et la température maximum de la série de mesures.

Ce sous-programme doit recevoir le tableau de mesures et le nombre de ces mesures. Il doit renvoyer les deux valeurs correspondant au minimum et au maximum des valeurs du tableau de mesures.

2. Écrire l'algorithme du sous-programme `Tproche0()` dont le but est de trouver la mesure de température la plus proche de 0°C.

Ce sous-programme doit recevoir le tableau de mesures et le nombre de ces mesures. Il doit renvoyer la valeur la plus proche de 0°C.

Si deux mesures de signes opposés ont la même valeur absolue, on retient la mesure positive.

L'algorithme de ce sous-programme utilise la fonction `fabs()` de la bibliothèque standard sur les mathématiques qui renvoie la valeur absolue d'un nombre réel.

3. Écrire l'algorithme du sous-programme `calculEffectif()` qui renvoie le nombre de mesures de température comprises entre deux bornes.

Ce sous-programme doit recevoir le tableau de mesures, le nombre de ces mesures et les deux bornes de la classe de températures demandée. Il renvoie l'effectif de cette classe, c'est à dire le nombre de mesures de températures comprises entre les deux bornes.

Pour utiliser ce sous-programme, on calcule les effectifs des quatre classes suivantes :]-10, 5] ;]-5, 0] ;]0, 5] ;]5, 10].

Réalisation en séance

1. Compléter le programme principal de la partie précédente un y intégrant un par un les sous-programmes demandés.

Pour valider chaque sous-programme, afficher les valeurs intermédiaires en vis-à-vis de la mesure analysée et de son indice.

Par exemple, dans le cas du sous-programme `minmaxTemp()` on affiche pour chaque valeur du tableau son indice puis les évolutions des valeurs minimum et maximum. Le but est de reconstituer la trace de l'évolution des variables telle qu'elle a été étudiée en TD.