C++ - TP 6
Problème: notes d'étudiants (4h)

On considère un professeur qui possède un tableau de notes de ses étudiants aux différents contrôles du trimestre. Les étudiants sont numérotés de 1 à $NBETU$, les notes sont numérotées de 1 à $NBNOTES$. Le tableau de notes $Notes$ est consitué de $NBETU * NBNOTES$ cases. Dans ce tableau, dans les $NBETU$ premières cases sont stockées les notes du premier contrôle, à savoir $Notes[0]$ contient la note de l'étudiant 1 au contrôle 1, $Notes[1]$ contient la note de l'étudiant 2 au contrôle 1...$Notes[NBETU-1]$ la note de l'étudiant $NBETU$ au contrôle 1. Puis on stocke les notes du contrôle numéro 2, à savoir $Notes[NBETU]$ contient la note de l'étudiant 1 au contrôle 2... Les notes seront stockées au format réel.

1 Initialisation

Déclarez les constantes NBETU (égale à 25) et NBNOTES (égale à 5). Déclarez aussi la constante $TAILLE = NBETU*NBNOTES$. Dans votre *main*, déclarez le tableau $Notes$. Codez une fonction `init_alea(float t[], int taille)` qui remplit un tableau avec des valeurs aléatoires réelles comprises entre 0 et 20, avec une précision de 10^{-2}. Faites aussi une fonction d'affichage `affiche(int n,float t[], int taille)` qui affiche les notes de l'étudiant numéro n à tous les contrôles. Dans votre *main*, testez vos fonctions sur le tableau $Notes$. Vous afficherez notamment la liste des notes de chaque étudiant.

2 Import d'un tableau test

Pour faciliter la vérification de vos algorithmes, nous vous proposons pour la suite du TP de tester vos programmes avec un tableau que nous vous donnons sous Claco. Il s'appelle $Notes\_bis$ et se trouve dans le fichier `TP6_notes.h` que vous inclurez à votre projet comme fait lors du TP5. Dans votre main, supprimez la déclaration et l'initialisation du tableau $Notes$, et affichez simplement le tableau $Notes\_bis$.

3 Calculs pour un étudiant

1. Ecrire une fonction qui calcule et renvoie la meilleure note d'un étudiant donné (le numéro est passé en paramètre).
2. Ecrire une fonction qui renvoie le numéro du contrôle où un étudiant donné a eu sa plus mauvaise note.
3. Ecrire une fonction qui calcule la moyenne "améliorée" d'un étudiant donné, càd en excluant sa plus mauvaise note.
4. Dans votre main, demandez à l'utilisateur de taper un numéro d'étudiant, puis affichez les trois informations ci-dessus. Par exemple pour le tableau $Notes\_bis$, pour l'étudiant numéro 13 vous devez trouver 19.44 comme meilleure note, 4 comme numéro de contrôle le plus mauvais, et 10.46 comme moyenne privée du plus mauvais contrôle. Pour l'étudiant 25, vous devez trouver $(19.15, 3, 14.75)$ comme triplet.
5. Modifier votre programme pour redemander à l'utilisateur un nouveau numéro d'étudiant jusqu'à ce qu'il tape 0 pour quitter le programme.

4 Statistiques

6. Ecrire une fonction qui prend en paramètre un tableau $m1$ de réels de taille $NBETU$ et qui le remplit avec les moyennes "améliorées" de chaque étudiant sur tous les contrôles. Autrement dit, $m1[0]$ contiendra la moyenne de l'étudiant 1, $m1[1]$ la moyenne de l'étudiant 2... Testez votre fonction dans le main en affichant le tableau avec les moyennes améliorées de tous les étudiants.
7. Ecrire une fonction qui prend en paramètre un tableau $m2$ de réels de taille $NBNOTES$ et qui le remplit avec les moyennes de la classe pour chaque contrôle. Autrement dit, $m2[0]$ contiendra la moyenne de la classe au contrôle 1, $m2[1]$ la moyenne pour le contrôle 2... Testez votre fonction puis affichez

le tableau ainsi calculé. Pour le tableau de notes *Notes_bis*, vous devez trouver pour $m2$ le quintuplet $(9.81, 10.52, 9.30, 10.24, 10.22)$.

8. Ecrire une fonction qui retourne pour un étudiant de numéro reçu en paramètre, le nombre de contrôles où cet étudiant a eu plus que la moyenne du groupe. Testez votre fonction dans le main. Par exemple, pour l'étudiant 6, vous devez trouver 2, pour l'étudiant numéro 25 vous devez trouver 3.
9. Ecrire une fonction qui renvoie le numéro du contrôle où il y a le plus gros écart entre la plus mauvaise et la meilleure note (pour le tableau test, c'est le contrôle numéro 4).
10. Pour ce professeur, un étudiant valide le module si sa moyenne "améliorée" est d'au moins 10, et que sa plus mauvaise note est supérieure à 4. Affichez la liste des numéros des étudiants qui valident le module. Pour le tableau test, vous devez trouver les étudiants : 4 5 9 13 15 16 17 18 21 25.

Pour les plus rapides... (*"Si rapide que soit notre course, nous n'atteindrons jamais l'horizon."*, Robert Sabatier)

5 Mots triangulaires

Le n-ième terme ($n > 0$) de la séquence des *nombres triangulaires* est donnée par $t(n) = \frac{n(n+1)}{2}$ (c'est-à-dire, le n-ième nombre triangulaire est la somme des entiers de 1 à n).

Donc les 10 premiers nombres triangulaires sont : 1, 3, 6, 10, 15, 21, 28, 36, 45, 55, ...

En convertissant chaque lettre d'un mot en un nombre correspondant à sa position dans l'alphabet ($a = 1, b = 2, \ldots, z = 26$) et en ajoutant ces valeurs, nous formons la valeur d'un mot. Par exemple, la valeur du mot *sky* est $19 + 11 + 25 = 55 = t(10)$.

Si la valeur d'un mot est un nombre triangulaire, alors nous appellerons ce mot un *mot triangulaire*.

Dans les fichiers `TP_fable1.h` et `TP_fable2.h`, vous trouverez 2 constantes et 2 tableaux de caractères associés (les constantes représentent les tailles des tableaux). Chaque tableau contient des mots, décrits caractère par caractère (1 caractère par case, uniquement des caractères minuscules entre `a` et `z`, les accents et ponctuation ont été supprimés, certains mots peuvent donc ne comporter qu'une seule lettre). Chaque mot est séparé d'un autre par un espace.
Exemple : le tableau ci-dessous contient la phrase *je suis à l'IUT* (il y a 5 mots distincts),

j	e		s	u	i	s		a		l		i	u	t

et il serait défini comme ceci,

```
const int N=15;
char T[N]={'j','e',' ','s','u','i','s',' ','a',' ','l',' ','i','u','t'};
```

A faire : Le 1er tableau (celui de `TP_fable1.h`) contient 13 mots triangulaires. Combien le second tableau (`TP_fable2.h`) en contient-il ?

6 Médiane

Nous allons reprendre le fichier utilisé lors du TP5, `TP5_data.h`, disponible sous Claco, dans lequel vous trouverez la déclaration d'une constante et d'un tableau. Importez le fichier `TP5_data.h` :

```
#include "TP5_data.h"
```

A faire : Nous souhaitons calculer la *valeur médiane* du tableau. La valeur médiane est la valeur (du tableau) pour laquelle on trouve autant de valeurs supérieurs qu'inférieures, ou qui minimise cette différence (les nombres de valeurs supérieurs et inférieurs peuvent être différents). Si plusieurs nombres sont candidats pour être la valeur médiane, on gardera le premier dans le tableau (pas le plus petit).
Utilisez la fonction *compte* faite pendant le TP 5, elle vous servira dans cet exercice.