TP2 - BOUCLES SIMPLES – SÉLECTION MULTIPLE

Le but de ce TP est de renforcer les notions de boucles et d'accumulation d'une part et de se familiariser avec la notion de sélection multiple d'autre part.

Il permet encore de prendre connaissance des fonctions mathématiques standards.

Il permet enfin d'illustrer la division euclidienne (ou division de nombres entiers) basée sur le quotient et le reste.

Calcul de la moyenne de nombres lus au clavier

Il s'agit de faire un programme qui permet de lire 10 nombres entiers au clavier, qui calcule au fur et à mesure leur somme et affiche en toute fin leur moyenne.

Les nombres seront tous lus en utilisant **une seule variable** dont le nom ou l'identificateur sera `nombre`. Attention ! On n'utilise pas de tableau.

Préparation

1. Écrire l'algorithme du calcul de la moyenne en suivant la méthodologie vue en TD.
2. Préparer les jeux de test en remplissant les colonnes non grisées.

Valeurs fournies au clavier (Donner une liste de 10 valeurs dans chacune des cases ci-dessous)	Valeur que devrait afficher le Programme	Résultat affiché après exécution	Le fonctionnement est-il conforme ?

Réalisation en séance

1. Créer un nouveau répertoire `TLP_CPP\TP2` dans lequel on copie le fichier `TLP_CPP\modele.cpp` en le renommant `moy_nb.cpp`.
2. Ouvrir, dans **Code::Blocks**, un nouveau projet ayant pour fichier source `moy_nb.cpp`.
3. Traduire l'algorithme en C++ et l'éditer dans **Code::Blocks**.

4. Tester le programme. Pour cela vous utiliserez les jeux de test préparés précédemment et vous remplirez le tableau (cases grisées) pour comparer les résultats attendus et ceux que vous avez.
5. Corriger le code si nécessaire puis conclure.

Calculatrice avec menu

Il s'agit d'écrire un programme simulant une calculatrice rudimentaire en nombres entiers.

Le programme se présentera sous la forme d'une boucle qui affichera en début de chaque passage le menu suivant :

```
Menu de la calculatrice
     +  addition
     -  soustraction
     *  multiplication
     /  division euclidienne
     Q  quitter
Entrez votre choix :
```

Le choix de l'utilisateur est lu au clavier et demandé à nouveau s'il n'est pas valide (i.e. différent de tous les choix proposés).

Une fois le choix lu (et valide !), le programme affiche un message demandant à l'utilisateur la valeur des deux opérandes `oper_a` et `oper_b` puis les lit.

Suivant le choix de l'utilisateur, l'opération adéquate est appliquée sur les deux opérandes et le résultat (ou les résultats dans le cas de la division euclidienne) est affiché.

La boucle prend fin quand l'utilisateur a choisi de quitter et la fin du programme, dans ce cas, se réduit à afficher un message.

Préparation

1. Écrire l'algorithme en suivant la méthodologie vue en TD selon les quatre étapes suivantes :
   - Affichage du menu et Choix de l'utilisateur
   - Lecture des opérandes
   - Calcul du résultat
   - Affichage du résultat
2. Préparer les jeux de tests suivants (en remplissant les cases non grisées).

Jeux de tests de l'étape 1 (Affichage menu et choix utilisateur)

Caractère choix fourni au clavier	La boucle doit-elle se terminer ?	Si OUI valeur qui doit être affichée par le PP	Le fonctionnement est-il conforme ?

Jeux de tests de l'étape 2 (Lecture opérandes)

Valeurs fournies au clavier	Valeurs qui devraient être affichées par le programme principal	Valeurs affichées après exécution	Le fonctionnement est-il conforme ?

Jeux de tests du programme complet

Valeur fournie pour l'opération	Valeurs fournies pour les opérandes	Affichage attendu	Le fonctionnement est-il conforme ?

Réalisation en séance

1. Copier le fichier `TLP_CPP\modele.cpp` dans le dossier `TLP_CPP\TP2` en le renommant `calculette.cpp`.
2. Ouvrir, dans **Code::Blocks**, un nouveau projet ainsi que le fichier `calculette.cpp`.
3. Coder l'étape 1 du programme puis tester et valider cette étape à l'aide des jeux de tests préparés (cases grisées).
4. Compléter le programme précédent en y intégrant le code de l'étape 2, puis tester et valider cette étape à l'aide des jeux de tests préparés (cases grisées).
5. Compléter le programme précédent en y intégrant le code de l'étape 3 du programme puis tester et valider cette étape à l'aide des jeux de tests préparés (cases grisées).
6. Compléter le programme précédent en y intégrant le code de l'étape 4 du programme puis tester et valider cette étape à l'aide des jeux de tests préparés (cases grisées).
7. Valider le fonctionnement du programme complet.