

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 sous forme réelle. Les nombres seront tous lus en utilisant **une seule variable**. Attention ! Pour que le résultat du **calcul de la moyenne soit réel**, il faut que le numérateur ou le dénominateur de la division soit lui-même un nombre réel.

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

- Dans **Code::Blocks**, créer un nouveau projet TP02 puis copier le fichier `modele.cpp` en le renommant `moyenne10entiers.cpp`.
- Dans le projet TP02, ajouter le fichier source `moyenne10entiers.cpp`.
- Traduire l'algorithme en C++ et l'éditer dans **Code::Blocks**.
- Tester le programme. Pour cela, utiliser les jeux de test préparés et remplir le tableau (cases grisées) avec les résultats du programme.
- Corriger le code si les résultats du programme ne correspondent pas puis conclure.

Calculatrice avec menu

Il s'agit d'écrire un programme simulant une calculatrice rudimentaire sur des 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 :
```

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

Pour l'addition, la soustraction et la multiplication, l'opération adéquate est appliquée sur les deux opérandes et le résultat est affiché.

Dans le cas de la division euclidienne, ce sont le quotient et le reste qui sont affichés.

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 du programme principal (en suivant la méthodologie vue en TD) de façon à réaliser les actions suivantes :
 - Étape 1 : Affichage du menu et saisie du choix de l'utilisateur
 - Étape 2 : Traitement du choix de l'utilisateur à l'aide de l'instruction SUIVANT. Pour chaque opération, la suite des actions est :
 - a) Lecture des opérandes
 - b) Calcul du résultat
 - c) Affichage du résultat
2. Préparer les jeux de tests ci-après (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

- Copier le fichier `TLP_CPP\modele.cpp` dans le dossier `TLP_CPP\TP2` en le renommant `calcullette.cpp`.
- Ouvrir, dans le projet TP02, le fichier `calcullette.cpp`.
- Coder l'étape 1 du programme puis tester et valider cette étape à l'aide des jeux de tests préparés (cases grisées).
- 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).
- Valider le fonctionnement du programme complet.

Bonus Ajout du calcul de x^n

Il s'agit d'ajouter un calcul supplémentaire au menu de la calculatrice. Dans ce but il faut reprendre l'algorithme de l'exercice de la séance de Travaux Dirigés n°2 sur le calcul de x^n .

Préparation

1. Traduire l'algorithme de l'exercice 3 du TD 2 en C++
2. Préparer un jeu de tests correspondant aux valeurs limites du TD

Réalisation en séance

1. Dans le même projet **Code::Blocks**, compléter le menu avec le nouveau choix de calcul de puissance.
2. Dans la partie traitement des calculs, ajouter le code de l'exercice de TD en adaptant la lecture des opérandes.