

TP3 – CALCULATRICE AVEC SOUS-PROGRAMMES

Le but de ce TP est d'aborder les sous-programmes. On retrouve ici les structures de contrôle vues précédemment : l'alternative et les boucles. À ces structures, on ajoute les notions de procédure et de fonction. **Le TP comprend trois parties :**

- La modification de sous-programmes comportant des erreurs
- La modification du programme « calculatrice avec menu » (TP précédent) en y insérant des sous-programmes
- La création d'un sous-programme supplémentaire permettant de faire le calcul de factorielle n.

Calcul du PGCD de deux nombres entiers

Le PGCD de deux nombres entiers est le **Plus Grand Commun Diviseur**. La description de l'algorithme qui permet de le calculer est la suivante :

- On compare les deux nombres entiers.
- On échange le plus grand par le résultat de la soustraction du plus petit au plus grand.
- Tant que les deux nombres ne sont pas égaux on reprend les deux points précédents.

Une fois que les deux nombres sont égaux, le résultat obtenu est le PGCD.

Exemple : le PGCD de 15 et de 20 est 5

$$\text{PGCD}(15 ; 20) = \text{PGCD}(15 ; 5) = \text{PGCD}(10 ; 5) = \text{PGCD}(5 ; 5) = 5$$

Pour réaliser le programme correspondant, nous allons utiliser deux sous-programmes :

- Le sous-programme `echange()` qui échange le contenu de deux paramètres formels p et q
- Le sous-programme `pgcd()` qui renvoie en résultat le PGCD des 2 paramètres formels a et b.

Le programme complet est fourni mais comporte des erreurs qu'il faut détecter et corriger.

Préparation

1. Représenter la vue externe de chacun de ces sous-programmes.
2. En déduire l'en-tête puis donner l'algorithme de chacun d'eux.

Réalisation en séance

1. Créer un nouveau projet TP03 puis copier le fichier `pgcd.cpp` se trouvant dans le répertoire `COMMUN\1A\CPP\TP03` dans votre répertoire `TLP_CPP\TP03`.
2. Enlever le fichier `main.cpp` du projet et ajouter le fichier source `pgcd.cpp`.
3. Compiler et exécuter le programme.
4. Tester le programme. Analyser les erreurs puis corriger le code.

Calculatrice avec menu

L'objectif de cet exercice est de **modifier le programme** réalisé lors du TP précédent en le **découpant en sous-programmes**.

Préparation

1. Relire la description du programme permettant de réaliser une calculatrice avec menu du TP précédent.
2. Représenter la vue externe d'un sous-programme `menu()` qui affiche les choix possibles et retourne le choix de l'opération. Le choix retourné par le sous-programme `menu()` doit appartenir à la liste proposée. Donner l'en-tête du sous-programme et son algorithme.
3. Représenter la vue externe du sous-programme `lire_operandes()` qui lit et renvoie deux entiers. En déduire son prototype et son algorithme.
4. Reprendre l'algorithme du programme **calcullette** du TP précédent en le modifiant de manière à ce qu'il fasse appel aux sous-programmes `menu()` et `lire_operandes()`.

Réalisation en séance

1. Copier le fichier `TLP_CPP\TP02\calcullette.cpp` dans le dossier `TLP_CPP\TP03` puis ouvrir ce fichier dans **Code::Blocks**
2. Traduire l'algorithme du programme en C++ et l'éditer dans **Code::Blocks**.
3. Tester le programme en validant le fonctionnement de chaque sous-programme individuellement.
4. Corriger le code.

Rajout d'une fonctionnalité supplémentaire : factorielle

On définit factorielle n (et on note $n!$) la quantité $1 \times 2 \times 3 \dots \times n$.

Ainsi $5! = 1 \times 2 \times 3 \times 4 \times 5 = 120$

Le but de cet exercice est de modifier le programme **calculatrice avec menu** afin qu'il propose la possibilité de calculer $n!$

Préparation

1. Représenter la vue externe du sous-programme `factorielle()` qui reçoit en paramètre formel un entier n et qui retourne la valeur de $n!$. En déduire son prototype.
2. Écrire l'algorithme du sous-programme `factorielle()`.

Réalisation en séance

1. Traduire l'algorithme du sous-programme `factorielle()` en C++.
2. Modifier le programme principal afin de valider le sous-programme `factorielle()` (le programme principal existant étant mis en commentaires c'est à dire précédé de `/` et terminé par `*/`).
3. Modifier le programme principal afin qu'il propose le calcul de $n!$
4. Corriger le code si nécessaire.

Bonus – formatage de l'affichage des résultats

Dans le but d'afficher correctement les résultats des calculs de factorielles pour des valeurs de n supérieures à 17, il est nécessaire de passer par des variables de type réel.

Comme le résultat d'un calcul de factorielle est nécessairement entier, on souhaite afficher uniquement la partie entière du résultat du calcul réalisé avec des variables de type réel. On fait appel à la bibliothèque `iomanip` pour formater l'affichage du résultat.

Préparation

1. Consulter la page <http://www.cplusplus.com/reference/iomanip/setprecision/>

Réalisation en séance

1. À partir de l'exemple de la page Web, utiliser les outils `fixed` et `setprecision` de façon à afficher les parties entières des résultats de calculs.