INF1005C - PROGRAMMATION PROCÉDURALE

Travail dirigé No. 2
Programmes simples
Entrées et sorties

Objectifs : Permettre à l'étudiant de faire ses premiers pas de programmation en langage C++.
Il apprendra à manipuler la structure de base d'un programme, les types de base ainsi que les entrées et les sorties du C++.
Durée : Une séance de laboratoire.
Remise du travail : Dimanche 3 février, avant 23 h 30.
Travail préparatoire : Leçon 4 sur Moodle, lecture des exercices et rédaction des algorithmes.
Directives : N'oubliez pas de mettre les entêtes de fichiers et de respecter le guide de codage (voir la dernière page de ce document pour les points à respecter).
Documents à remettre : Sur le site Moodle des travaux pratiques, vous remettrez l'ensemble des *fichiers .cpp* compressés dans un fichier .zip en suivant **la procédure de remise des TDs.**
Seulement 3 exercices seront corrigés (mais vous devez tous les faire)

Directives particulières

- Affichez toujours un message d'invite avant chaque saisie.
- Vous n'avez pas à valider les entrées.
- Vous n'avez pas à afficher les caractères accentués.
- Vous pouvez déclarer toutes les variables désirées.
- Utilisez le type **string** pour les chaînes de caractères.
- Pour chacun des exercices, utilisez des messages appropriés lors de l'affichage. Des chiffres seuls ne suffisent pas.
- Vous ne pouvez **pas utiliser** de structures de **décision** (incluant le `?:` qui n'est pas vu dans le cours) ni de structures de **répétition**. Indiquez en commentaire les endroits où il aurait été bien d'utiliser ces structures.

1. Soit un prénom composé (avec un trait d'union) entré par l'utilisateur (exemple « jean-luc »). Retrouvez le premier prénom (exemple « jean » pour « jean-luc ») puis affichez le message « Bonjour PremierPrénom » (exemple « Bonjour Jean »). La première lettre du prénom affiché doit être en majuscule (même si l'utilisateur l'a mis en minuscule). Indice : le type `string` possède des fonctions de recherche.

2. La surface d'un hexagone dont le rayon du cercle inscrit est r est donnée par : $2\sqrt{3}\,r^2$ Écrivez un programme qui demande le rayon d'un cercle, affiche son diamètre, sa circonférence et son aire. Le programme doit aussi afficher la différence entre la surface de l'hexagone et celle du cercle inscrit dans celui-ci.

3. Écrivez un programme qui génère aléatoirement un nombre réel entre 0 et 1 (utilisez les fonctions `srand()` et `rand()`, et la constante `RAND_MAX` ; voir l'exemple ci-dessous), et qui calcule directement son sinus (fonction sin(x)), puis en utilisant les **3 premiers termes** de la série :

$$\sin(x) \cong x - \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \frac{x^9}{9!} + \cdots$$

   Le programme affichera ensuite x et la différence entre l'approximation et la valeur réelle (telle que retournée par le `sin()` de `<cmath>`.

   Inspirez-vous de l'exemple suivant pour générer des nombres aléatoires entiers :

```cpp
#include <iostream>
#include <cstdlib> // Pour srand et rand.
#include <ctime> // Pour time.
using namespace std;
int main () {
    // Initialise le générateur aléatoire. (Une seule fois au début du programme)
    srand(time(nullptr));
    // Affiche un entier aléatoire entre 0 et la constante RAND_MAX.
    cout << rand();
}
```

4. Écrivez un programme qui demande de rentrer au clavier les coordonnées de 3 points 2D (donc en x et en y), les composantes étant des nombres réels. Il calcule ensuite le périmètre du triangle formé par les trois points (ne pensez pas au cas où ça fait une droite).
   Pour éviter la répétition de code, écrivez une fonction qui calcule la distance entre deux points dont le pseudo-code (syntaxe du TP1) est le suivant :

```
FONCTION calculerDistance (ax, ay, bx, by) :
    cx = bx - ax
    cy = by - ay
    RESULTAT racineCarrée(cx*cx + cy*cy)
```

5. Dans un fichier *machamp.txt*, vous avez les statistiques d'un Pokémon, ainsi que celles de deux de ses attaques. Écrivez un programme qui lit le fichier et crée un nouveau fichier *moves.txt* qui donne le dommage causé par chaque attaque selon un niveau de défense donné par l'utilisateur. Les fichiers doivent avoir la forme suivante (on vous fournit *machamp.txt* avec des valeurs différentes) :

machamp.txt	*moves.txt*
Machamp LVL 42 ATK 175 Strength 80 Superpower 120	Machamp vs DEF=100 Strength 52 Superpower 77

Ici, l'utilisateur aurait donné un niveau de défense de 100. Le nom du pokémon peut avoir plusieurs mots, mais les noms d'attaques sont en un seul mot et les valeurs doivent être écrites en entiers arrondis vers le bas.

Voici le pseudo-code pour la fonction de calcul de dommage :

```
FONCTION calculerDommage (lvl, atk, pwr, def) :
    RESULTAT ((2*lvl/5 + 2) * pwr * atk/def)/50 + 2
```

Par exemple, dans l'exemple ci-dessus, lvl=42 atk=175, def=100 et pwr=80 pour *Strength* et pwr=120 pour *Superpower*. Effectuez tous les calculs avec des entiers (pas de réels).

Les points du **guide de codage** à respecter **impérativement** pour ce TD sont les suivants :
(lire le guide de codage sur le site Moodle du cours pour la description détaillée de chacun de ces points)

3: noms des variables en lowerCamelCase
4: noms des constantes en MAJUSCULES
25: is/est pour booléen
27: éviter les abréviations (les acronymes communs doivent être gardés en acronymes)
29: éviter la négation dans les noms
33: entête de fichier avec vos noms et matricules
42: #include au début (mais après l'entête)
46: initialiser à la déclaration
47: pas plus d'une signification par variable
62: pas de nombres magiques dans le code
63-64: « double » toujours avec au moins un chiffre de chaque côté du point
79,81: espacement et lignes de séparation
85: mieux écrire le programme plutôt qu'ajouter des commentaires
87: préférer //

Plusieurs points du guide montrent comment bien indenter, mais comme dans ce TD la seule construction qui demande une indentation est la définition du « main », la forme du programme devrait être :

```
#include ...
Tous les débuts de lignes
sont alignés à gauche
...
int main()
{
    Tous les débuts de lignes
    sont alignés avec
    une seule indentation (tabulation)
    ...
}
```