

**R 6 . A . 0 5 / C + +**

**T P 1**

**P o i n t e u r s - T a b l e a u x -  
C h a î n e d e c a r a c t è r e s**

**2 0 2 4 - 2 0 2 5**

UTILISATION OBLIGATOIRE DE L'IDE « VSCODE » POUR TOUTES LES SÉANCES.

CET IDE PEUT ETRE INSTALLE :

- sur un linux d'un PC personnel
- sur un linux Ubuntu disponible de base sur Windows11 après avoir installé WSL « Windows Subsystem for Linux » (mais sans doute pas sur un linux Ubuntu sous Windows10)

Chaque fois que l'on créera sous *VSCode* un espace de développement, on créera obligatoirement l'arborescence suivante :

```
          /<nom_projet>
        /
       /src      |      \ws      \bin
      *.cpp      |      Makefile  *.bin
      *.h        |
                 \      *.o
```

Pour ce TP1, le projet C++ à créer sous VSCode s'appelle « ProjetTP1 ».

### Exercice N°1

Ecrire un lanceur *main* qui affiche simplement la chaîne de caractères "Ca marche !" à l'écran.

Bien séparer le fichier source (*Exo1.cpp*) qui contient le lanceur *main* du fichier d'en-tête *Exo1.h*. Ce fichier d'en-tête importe la bibliothèque standard `<iostream>` et définit l'espace de nommage *std*. Pour éviter les inclusions multiples de ce fichier d'en-tête, rajouter les directives de compilation conditionnelles.

Contenu du fichier *Exo1.h*.

```
#ifndef EXO1_H
#define EXO1_H

#include <iostream>
using namespace std ;

#endif
```

Contenu du fichier *Exo1.cpp*.

```
#include "Exo1.h"

int main() {...}
```

Pour compiler et exécuter le programme, écrire dans le fichier *Makefile* une cible qui s'appellera *lanceurExo1* (attention aux tabulations !!) :

```
GPP = g++ -Wall
SRC = ./src
BIN = ./bin
```

```
lanceurExo1 :
    @echo Compilation lanceurExo1
    $(GPP)      $(SRC)/Exo1.cpp -o    $(BIN)/Exo1.bin
```

Pour l'exécution, sous *VSCode* cliquer sur "Makefile" dans le menu de la barre de gauche puis cliquer sur l'icône "Run the selected binary target in the terminal".

### Exercice N°2

Ecrire un programme (*Exo2.cpp*) capable de lire une chaîne de caractères entrée au clavier et terminée par la touche ENTREE (utiliser *cin.getline(...)* **obligatoirement**).

La longueur max de la chaîne sera de 120 caractères; elle sera définie sous forme d'une constante (*#define MAX 120*) dans le fichier *Exo2.h*.

Inclure *<string.h>* (notamment pour l'usage de *strlen(...)*). Donc rajouter la ligne *#include <string.h>* dans le fichier *Exo2.h*.

Afficher ensuite :

- La longueur totale de la chaîne.
- Un à un les caractères (une chaîne de caractères est aussi un tableau).

Pour compiler le programme, écrire dans le fichier *Makefile* une cible supplémentaire qui s'appellera *lanceurExo2* (attention aux tabulations !!) :

```
GPP = g++ -Wall
SRC = ./src
BIN = ./bin
```

```
lanceurExo1 :
    @echo Compilation lanceurExo1
    $(GPP)      $(SRC)/Exo1.cpp -o    $(BIN)/Exo1.bin
```

```
lanceurExo2 :
    @echo Compilation lanceurExo2
    $(GPP)      $(SRC)/Exo2.cpp -o    $(BIN)/Exo2.bin
```

Exercice N°3

Ecrire un programme (*Exo3.cpp*) qui lit une chaîne de caractères au clavier (utiliser *cin.getline(...)* **obligatoirement**) et la mémorise dans un tableau *char\* pCh*.

On désire compter et afficher les occurrences des lettres de l'alphabet contenues dans « pCh » en ne distinguant pas les majuscules des minuscules. Le nombre d'occurrences pour chaque lettre sera stocké dans un tableau « tab » de dimension 26. On utilisera un pointeur « p1 » pour parcourir la chaîne « pCh » et un pointeur « p2 » pour parcourir « tab ».

Afficher seulement le nombre d'occurrences des lettres qui apparaissent au moins une fois dans le texte.

**Rappel** : un caractère est représenté en mémoire par son code ASCII ( 'A' = 65, 'B' = 66, ..., 'a' = 97, 'b' = 98, ...). Dès lors un caractère peut aussi être considéré comme un entier codé sur 1 octet.

Pour compiler le programme, écrire dans le fichier *Makefile* une cible supplémentaire qui s'appellera *lanceurExo3* (attention aux tabulations !!) :

GPP = g++ -Wall

SRC = ./src

BIN = ./bin

lanceurExo1 :

    @echo Compilation lanceurExo1

    \$(GPP) \$(SRC)/Exo1.cpp -o \$(BIN)/Exo1.bin

lanceurExo2 :

    @echo Compilation lanceurExo2

    \$(GPP) \$(SRC)/Exo2.cpp -o \$(BIN)/Exo2.bin

**lanceurExo3 :**

**@echo Compilation lanceurExo3**

**\$(GPP) \$(SRC)/Exo3.cpp -o \$(BIN)/Exo3.bin**

Références nécessaires à l'ensemble du TP1

- Les flux d'entrées/sorties *cin* et *cout* instances respectives des classes de la bibliothèque standard C++, « istream » et « ostream ».
- Les cours 1 et 2 C++.

La définition des classes de la bibliothèque standard est visible sur les sites :

<http://www.cplusplus.com/ref>

<http://www.cppreference.com>