

Exercice 1 :

- a) Compilez, corrigez, et exécutez le programme en utilisant l'IDE de votre choix:

```
#include <iostream.h>
#define multiplier (x,y) (x * y)

int main(void)
{
    int a;
    int b;
    cout << "Entrez une valeur :" << endl;
    cin << a;
    cout << "Entrez une autre valeur :" << endl;
    cin << b;

    cout << multiplier (a+1, b+1) << endl;
}
```

- b) Au regard de (a), explorez la possibilité d'utiliser une fonction avec le mot clé inline et, dans certaines circonstances qui seront à préciser, les mots clés constexpr et constexpr vus en cours.

Exercice 2 :

- 1) Ecrire un programme C++ qui affiche la table de multiplication d'un chiffre entre 1 et 9 saisi par l'utilisateur.
- 2) Ecrire un programme C++ qui affiche l'intégralité de la table de multiplication parfaitement alignée (en utilisant la librairie [<iomanip>](#))

Exercice 3 :

- 1) Ecrire un programme C++ qui analyse un texte à partir d'un fichier en affichant le nombre de lignes, de mots et de lettres (l'utilisation de la classe «stringstream» est préconisée voire impératif).
- 2) Modifier le programme pour qu'il affiche aussi le nombre d'occurrences de chaque lettre de l'alphabet (sans différencier les majuscules des minuscules).

Exercice 4 :

- 1) Ecrire une classe « TextRectangle » :
 - La classe est composée de deux attributs privés [width](#) et [height](#), deux entiers positifs inférieurs à 10. Développer les *getters* et les *setters* de ces deux attributs.

- Développer un constructeur autour de ces deux attributs.
- Deux méthodes `getArea()` et `getPerimeter()` permettant d'obtenir respectivement la surface et le périmètre du rectangle.
- Développer une méthode `print()` qui affiche l'ensemble des caractéristiques du rectangle.
- Développer une méthode `drawArea(const char& pattern)` qui dessine le rectangle en version texte en utilisant le motif passé en paramètre.
- Développer une méthode `drawPerimeter(const char& pattern)` qui dessine le périmètre du rectangle en version texte en utilisant le motif passé en paramètre.

2) Tester cette classe en l'instanciant dans la fonction `main()`, avec lecture des hauteur et largeur à partir de l'entrée standard.