

Série 6: les fonctions (2)

Lien avec le [MOOC Initiation à la Programmation \(en C++\)](#)
Lien avec ICC-Théorie en complément du MOOC

La solution des exercices du MOOC est disponible sur le MOOC.

ExC 1 : mesure de temps calcul : Ce type de mesure dépend de la machine. Cela dit l'ordre de complexité devrait se manifester indépendamment d'une machine sur les mesures avec les valeurs les plus grandes de N . Pour une comparaison robuste, il faut procéder en calculant le rapport de la mesure obtenue pour une valeur N et celle obtenue pour **un multiple $K*N$** . Si ce rapport est aussi de l'ordre de K alors l'ordre de complexité est linéaire en $O(N)$. Si le rapport des mesures est en K^2 , alors l'ordre de complexité est en $O(N^2)$, etc... Il faut faire des tests pour N suffisamment grand car le temps pris par les entrées-sorties est non-négligeable et perturbe la mesure ; ce temps ne nous intéresse pas.

ExC 2 : codage de César

```
#include <iostream>
using namespace std;

bool codage(char &c, int dec);

int main()
{
    int val(0);

    do {
        cout << "Entrez décalage : " ;
        cin >> val;
    } while(val < -13 or val > 13);

    char c(0);
    bool next(true);

    do {
        cin.get(c) ;
        next = codage(c, val);
        if(next)
            cout << c ;
    } while(next);

    return 0;
}

bool codage(char &c, int dec)
{
    if( c == ' ' or c == '\n' or c == '.') {
        return true;
    }

    if (c <= 'Z' and c >= 'A'){

        if(c + dec > 'Z')      c = 'A' + (dec - ('Z' - c));
        else if(c + dec < 'A') c = 'Z' + (dec + (c - 'A'));
        else                  c = c + dec ;
        return true;
    }
    return false;
}
```