

Exemple 4 : Déroulement de boucle

Pour compiler:

Se mettre dans le répertoire qui contient ce fichier

Taper:

```
python build.py -version Release
```

Si tout s'est bien passé : un fichier `ex_1_4` (ou `ex_1_4.exe`) est créé dans le répertoire `build/Debug`

Dans les commandes de compilation affichées à l'écran, noter les options `-g -O0`

Pour exécuter:

Taper :

```
./build/Debug/ex_1_4 n
```

où `n` est un entier positif (taille des vecteurs) si `n` n'est pas spécifié, le code prend `n = 10000000`

On calcule l'expression :

```
y = a * x + b
```

où `x` et `y` sont des vecteurs de taille `n`, `a` et `b` sont des scalaires

On compare les temps calcul de 2 versions:

- une boucle sur `n` itérations (chaque itération calcule 1 composante de `y`)
 - une boucle sur `n/4` itérations (chaque itération calcule 4 composantes)
-

Recompiler en mode optimise

Taper

```
python build.py -version Release
```

Dans les commandes de compilation affichées à l'écran, noter l'option `-O3`

Relancer l'execution du code

Taper :

```
./build/Release/ex_1_4
```

Comparer les temps des 2 versions