

## Exemple 6 (Cours 1)

### Quelques informations sur les vecteurs et tableaux

#### Pour compiler:

Se mettre dans le répertoire qui contient ce fichier Taper:

```
python build.py -m Debug
python build.py -m Release
```

Si tout s'est bien passé : 6 exécutables ex0, ex0b, ex0c, ex1, ex1b, ex1c sont créés dans chaque répertoire install et install/debug

#### Pour exécuter:

Taper :

```
./install/ex0
```

```
./install/ex1
```

Pour comparer l'utilisation de tableaux "bas niveau" double \* et de vecteurs standard C++.

Attention : les résultats dépendent du test considéré, ne pas faire de conclusions générales d'après les résultats de ce test.

#### Autre test

Lancer un moniteur de ressources depuis un autre terminal ou avec le menu (linux, par exemple: **gnome-system-monitor**) ou avec le gestionnaire de tâches (windows, taper ctrl-alt-suppr puis cliquer sur l'onglet "Performance")

Puis, exécuter

```
./install/ex0b
```

```
./install/ex1b
```

Dans chaque exécution (ex0b, ex1b), le code s'arrête pour examiner la mémoire utilisée par le code dans le moniteur. Pour continuer l'exécution, taper une touche quelconque + return