Corrigé du TP2

Disclaimer : Ce corrigé n'a pas pour but de vous fournir un détail des algorithmes répondant aux questions de l'énoncé, mais de vous résumer les notions clés dont vous aurez besoin pour résoudre ces exercices, les avertissements vis à vis de certaines erreurs classiques et les détails des passages les plus techniques, voire des précisions si vous souhaitez aller plus loin sur ces questions et plus généralement de synthétiser dans un même document la plupart des commentaires fait en TP afin qu'ils puissent profiter à tous. Pour obtenir un corrigé détaillé, il vous faudra donc vous référer à ce que vous avez pu faire en TP ou reprendre vous même les exercices à partir des notions présentées dans le présent corrigé pour écrire les algorithmes qui vous manquent.

En cas de difficultés, **n'hésitez pas à me contacter**, nous pourrons en discuter soit par mail soit en début de séance suivante. Ce corrigé n'a pour but que d'être une base de travail pour vous et de servir de complément par rapport à ce que vous avez fait en séance, il n'est en aucun cas fait pour se suffire à lui même.

Utilisation d'une boucle for

* Une boucle **for** permet de répéter une commande ou une suite de commandes qui peuvent dépendre d'un entier que l'on appelle **itérateur** qui s'incrémente à chaque passage dans la boucle.
* Deux structures qui fonctionnent pour une boucle for sont :

```
int i;
for (i=0; i<=Nmax; i++)
{
    commandes;
}
```

```
for (int i=0; i<=Nmax; i++)
{
    commandes;
}
```

* Il faut bien mettre des ; et non des , entre les conditions dans la parenthèse.

  Il ne faut pas mettre de ; après les parenthèses.

  La deuxième condition signifie tant que l'on a i<=Nmax, on continue la boucle, il ne s'agit PAS d'une condition d'arrêt, si on mettait i=Nmax, le programme ne fonctionnerait pas.
* Si vous avez des doutes, n'hésitez pas à regarder les premiers cas de la boucle for à la main en effectuant les calculs pour i=0 et i=1.

  Vous vérifierez ainsi que le programme fait bien ce que vous voulez, cela résout beaucoup de problèmes.
* Vous n'êtes pas obligés d'incrémenter i seulement de 1. Si vous voulez augmenter de plus, il faudra modifier la troisième condition i++ par i+=2 par exemple.

Utilisation des vecteurs

* Pour utiliser les vecteurs, il vous faut inclure la bibliothèque vector au début de votre programme avec la commande #include <vector>.
* Pour déclarer un vecteur, il ne suffit pas d'utiliser le type vecteur, mais déclarer également le type des éléments du vecteur par vector<double> par exemple si l'on veut un vecteur de réels. Deux stratégies s'offrent alors à vous pour créer un vecteur

  Vous créez un vecteur vide par un **vector<type> nom ;** puis vous ajoutez des cases en utilisant la commande **nom.push_back(élément)**

  Ou vous créez un vecteur d'une taille choisie n par la commande **vector<type> nom(n) ;** puis vous affectez les valeurs.
* Pour accéder à la i-ème case du vecteur, vous pouvez utiliser des crochets avec la commande **nom [i]**

  ATTENTION. En C++, les indices commencent à 0 et non à 1, la dernière case d'un tableau de taille n est donc la case $n-1$ et non la case n.
* Vous pouvez parcourir un vecteur à l'aide d'une boucle **for**, l'itérateur allant de 0 à l'indice (taille du vecteur -1) est le numéro de la case que l'on considère.

* Pour connaître la taille d'un vecteur, vous pouvez utiliser la commande vecteur.size()

  N'oubliez pas les parenthèses même s'il n'y a rien dedans.

* Pour appeler un élément d'un vecteur ou le modifier, vous pouvez utiliser la commande vecteur[indice]

  Ne confondez pas, dans le cas de la définition d'un vecteur de taille donnée, vous utilisez des parenthèses et dans le cas de l'appel de la case d'un vecteur, vous utilisez des crochets.

* Faites attention, la première case du vecteur est d'indice 0.

  La case d'indice vecteur.size() n'existe pas. Cependant, si vous tenter de l'utiliser C++ ne vous enverra pas de message d'erreur ! Cela peut être une erreur difficile à repérer.

Quelques fonctions auxilliaires

* Il est possible de se passer des fonctions suivantes dans le cadre du TP en les recodant, cependant, vous pouvez également les utiliser directement.
* Pour supprimer le dernier élément d'une liste : **nom.popback() ;**

  Dans ce cas, vous diminuez d'un la taille du vecteur.
* Pour repérer le premier indice du vecteur dans les fonctions suivantes, on utilise la fonction nom.begin() et non pas 0.
* Pour insérer un élément elem en position n :**nom.insert(nom.begin()+n,elem) ;**
* Pour supprimer l'élément à la position n : **nom.erase(nom.begin()+n) ;**

Rappels quant au modulo

* a%b signifie en c++ a modulo b, il s'agit du reste de la division euclidienne de a par b.
* Une utilisation à retenir du modulo est que n%2 vaut 0 si n est pair et 1 si n est impair.

Précisions sur l'exercice *B*)

* Pour utiliser les fonctions mathématiques, il faut inclure la bibliothèque cmath au début de votre programme. Vous pouvez alors utiliser les fonctions sin ou exp de façon usuelle.
* Une grosse difficulté de cet exercice est de bien effectuer les sommes sur les bons indices. Le plus simple et efficace est encore une fois de faire les premiers cas à la main pour vérifier que votre programme fait bien ce que vous voulez.

Quelques conseils pour retravailler le TP

* La syntaxe de la boucle for est un peu lourde, retenez la bien et ne la confondez pas avec les autres boucles vues précédemment.
* N'hésitez pas à tester à la main les premiers cas et les derniers cas d'une boucle for. C'est vraiment un moyen efficace de traquer les erreurs de code.

  Vous pouvez vous entraîner à ça avec les exemples vus en cours, TD ou TP.

  Cela peut vous sembler rébarbatif, voire un peu amateur, mais je vous garantis que même quand vous gagnez en expérience, vous continuez à faire ce genre de calculs, même si souvent vous pouvez vous aider pour cela d'un débugger.
* Plus que les quelques commandes sur les vecteurs, il est très important pour la suite que vous compreniez comment parcourir un vecteur à l'aide d'une boucle for. C'est le cœur de la deuxième partie de ce TP et quelque chose d'important pour la suite.
* Plus généralement, C++ est un langage qui est largement utilisé, n'hésitez pas à chercher sur internet en cas de difficultés, si vous en êtes capable en anglais, vous trouverez facilement des réponses à vos questions.
* En cas de doute, vous pouvez également consulter la documentation C++.