Programmation pour le calcul scientifique
Année : 2018-2019
Formation : L3 Ingénieurie Mathématique

TP1 : Rappels

1 Rappel : entrées/sorties

Pour afficher une variable à l'écran, on utilise la commande `cout` :

```
cout <<  variable_a_afficher << endl;
```

Pour lire des données rentrées par l'utilisateur au clavier, on utilise la commande `cin` :

```
cin >> variable_a_lire;
```

Dans les deux cas, il faut préciser au début du programme qu'on inclut la bibliothèque `iostream` :

```
#include <iostream>
```

2 Calcul des termes d'une suite

On définit la suite (u_n) par

$$u_1 = 1;\ u_2 = 3, u_n = \frac{1}{u_{n-2}} + u_{n-1}$$

Ecrire un programme qui, au choix de l'utilisateur :
- calcule la somme des n premiers termes de la suite (n donné par l'utilisateur),
- calcule le produit des n premiers termes de la suite (n donné par l'utilisateur),
- affiche les n premiers termes de la suite (n donné par l'utilisateur),

3 Racines d'un polynôme de degré deux

Ecrire un programme qui prend en entrée trois valeurs a, b et c et affiche le(s) solutions(s) si elles existent de l'équation

$$a \cdot x^2 + b \cdot x + c = 0.$$

4 Suite de Syracuse

Soit $\phi : \mathbb{N}^* \to \mathbb{N}^*$ l'application telle que $\phi(n) = \frac{n}{2}$ si n est pair et $\phi(n) = 3n + 1$ si n est impair. On appelle suite de Syracuse toute suite $(s_n)_{n\in\mathbb{N}}$ définie par son premier terme $s_0 \in \mathbb{N}^*$ et par la relation $\forall n \in \mathbb{N},\ s_{n+1} = \phi(s_n)$.
La conjecture de Syracuse postule que $(s_n)_{n\in\mathbb{N}}$ finit toujours par atteindre la valeur 1 et répète le motif 1, 4, 2, 1, 4, 2.... En dépit de la simplicité de son énoncé, cette conjecture n'a pas encore été prouvée.
Ecrire un programme qui lit la valeur de s_0 rentrée au clavier par l'utilisateur, calcule et affiche à l'écran les termes successifs de la suite de Syracuse, jusqu'à atteindre la valeur 1. Ce programme devra utiliser la commande `switch`.

5 Opérations sur des matrices creuses

Soit A une matrice quelconque, mais qui contient de nombreux éléments nuls, par exemple une matrice tridiagonale. On décide de stocker uniquement les éléments non nuls de A avec ce qu'on appelle un stockage creux. Le but de l'exercice est d'effectuer des opérations sur A en utilisant uniquement ce stockage creux.

1. Créez une structure qui contient un entier n, et trois tableaux (en utilisant la classe `vector`) respectivement appelés `ligne`, composé d'entiers, `colonne`, composé d'entiers, et `valeur`, composé de réels. Ces trois tableaux représentent respectivement les indices des lignes, les indices des colonnes, et les valeurs des éléments non-nuls d'une matrice.
2. Ecrire un programme qui calcule la trace de la matrice ainsi représentée, en utilisant uniquement les tableaux `ligne`, `colonne`, et `valeur`.
3. Ecrire un programme permettant de calculer la norme infinie de la matrice, en utilisant uniquement les tableaux `ligne`, `colonne`, et `valeur`.
4. Ecrire un programme permettant de faire le produit de la matrice par un vecteur `x` (de la classe `vector`), en utilisant uniquement les tableaux `ligne`, `colonne`, et `valeur`. Un conseil : écrire la formule sur un papier avant pour avoir les idées claires !
5. Ecrire un programme permettant, à l'aide des fonctions ainsi écrites, d'appliquer l'algorithme de la puissance itérée.