

SMP - TP1

PRÉAMBULE

Ce premier TP a pour objectif d'évaluer le niveau de programmation de la promotion, il n'est pas noté mais il sera corrigé et servira à ajuster le contenu et le niveau des cours suivants. Merci donc de ne surtout pas aller chercher la solution du voisin sous peine de fausser l'évaluation globale et d'augmenter artificiellement et inutilement le niveau de difficulté des cours!

Tous les exercices sont indépendants et peuvent être résolus dans n'importe quel ordre.
Merci de déposer vos solutions sur hippocampus par binôme et **d'accompagner vos codes de commentaires et d'un rapport pdf décrivant les algorithmes utilisés et les jeux d'essais.**

1 SUITE DE FIBONACCI

On rappelle que la suite de Fibonacci est donnée par ses deux premiers termes u_0 et u_1 et la récurrence $u_n = u_{n-1} + u_{n-2}$ pour $n \geq 2$.

Écrire un programme qui calcule les 20 premiers termes de la suite de Fibonacci.

2 DONNÉES ISSUES DE CAPTEURS

On s'intéresse à un capteur de pression atmosphérique renvoyant une pression atmosphérique en millibars. Les données sont acquises dans une boucle via `scanf()` ou `cin`.

La liste des valeurs du capteur se termine par `-1`.

Écrire un programme qui calcule le minimum, le maximum et la pression moyenne de la séquence de valeurs fournie au programme. Le programme ne doit pas stocker l'ensemble des valeurs en mémoire.

Exemple de séquence de valeur : 1013,1022,1021,1019,990,1013,1015,1005,-1

3 CRIBLE D'ERATHOSTÈNE

A partir d'une constante n définie en début de programme, écrire un programme pour appliquer le crible d'Erathostène sur les n premiers entiers. Pour rappel, l'algorithme consiste à considérer un tableau d'entiers dans lequel on va marquer comme non-premier tous les entiers divisibles par n'importe quel entier $i \leq \sqrt{n}$.

4 PGCD

Écrire une fonction qui calcule le PGCD de deux nombres.

5 POINTS ET POLYGONE : STRUCTURE DE DONNÉES

Concevoir une structure de données permettant de stocker un point : celui-ci sera défini par son nom (un seul caractère) et ses coordonnées entières.

Concevoir ensuite une structure polygone permettant de stocker les coordonnées des sommets d'un polygone.

Écrire une fonction permettant de calculer le périmètre d'un polygone passé en paramètre.

6 RÉPERTOIRE

À partir de la structure de données définie ci-dessous définissant une entrée dans un répertoire, écrire un programme de *gestion* de répertoire. Celui-ci devra être capable de :

- Créer un répertoire vide
- afficher l'ensemble des entrées présentes dans le répertoire
- insérer une nouvelle entrée dans un répertoire
- rechercher une entrée (par exemple par nom) dans le répertoire et l'afficher

```
struct entree {  
    string nom;  
    string prenom;  
    string telephone;  
};
```