

TP : Algorithmes et langage C

Evaluation : les exercices 2-5 sont à rendre avant le 30 novembre 2023.

Exercice 1 (sous Linux) :

Créer un répertoire `tp_algo`

```
mkdir tp_algo
```

Aller dans ce répertoire et regarder ce qu'il y a dedans

```
cd tp_algo
```

```
ls -la
```

Créer le fichier `coucou.c` (écrire `coucou !` à l'écran)

Editer son contenu avec `xemacs`.

```
xemacs coucou.c &
```

Compiler dans la fenêtre d'exécution avec la commande

```
gcc -o coucou coucou.c
```

Exécuter le programme dans la fenêtre de commande

```
./coucou
```

Regarder quels fichiers ont été créés et les détruire avec la commande

```
rm nom_de_fichier
```

Exercice 2 : impression d'une matrice de Hilbert

Demander à l'utilisateur un entier compris entre 1 et 6 puis imprimer de façon claire la matrice de Hilbert d'ordre `n`.

La matrice de Hilbert (a_{ij}) est telle que $a_{ij} = 1 / (i + j + 1)$.

Le spécificateur de format qui permet d'écrire un réel en précisant le nombre total `p` de caractères et le nombre `d` de chiffres après la virgule est le suivant: `%p.df` ; par exemple `%8.3f` si on veut 8 caractères dont 3 décimales (en comptant le point).

Exercice 3 : nombres premiers

Lire un entier `N` strictement positif et imprimer tous les nombres premiers compris entre 1 et `N`.

Exemple : si on indique l'entier 10, le résultat doit être : 1 2 3 5 7.

A. Utiliser seulement une fonction `main`

B. Utiliser une fonction `est_premier`

Exercice 4 : recherche dichotomique

L'utilisateur indique la valeur d'un "réel" a et deux "réels" nommés \min et \max . Ces trois réels seront codés avec des variables de type `double`. Le programme contient la définition de la fonction $f(x) = x^3 - a$. On pourra avoir une variable globale a ; la fonction f aura pour prototype : `double f(double x)` ; le paramètre x correspond à la valeur de la variable x et la valeur de retour à la valeur de $f(x)$.

Il s'agit de chercher une éventuelle solution comprise entre \min et \max de l'équation $f(x) = 0$ et cela par dichotomie.

Si $f(\min)$ et $f(\max)$ sont de même signe, le programme indique qu'il ne peut pas répondre à la question et s'arrête. Dans le cas contraire, la solution est calculée à 10^{-6} près.

On prévoira le programme pour qu'il puisse fonctionner si on change de fonction f .

Un point technique : pour saisir un `double` avec `scanf`, l'indicateur de format est `%lf`.

Exercice 5 algorithme de Horner

Réaliser l'implémentation du calcul de la valeur d'un polynôme P en un point en utilisant l'algorithme de Horner :

$$P(x) = (((...(a_n x + a_{n-1})x + a_{n-2})... a_1) + a_0).$$

Le polynôme sera saisi en indiquant son degré puis tous ses coefficients qui seront mis dans un tableau.

Après la saisie du polynôme, le programme demandera à l'utilisateur s'il veut l'évaluer; si oui, celui-ci répondra par le caractère 'o' ; il devra alors donner la valeur de x pour lequel il demande le calcul de $P(x)$. le programme redemandera alors si l'utilisateur veut à nouveau évaluer le polynôme P , et cela jusqu'à ce que l'utilisateur réponde par le caractère 'n'.

Un point technique : L'instruction :

```
fflush(stdin);
```

devra probablement être utilisée avant la saisie d'un caractère pour vider le buffer de lecture (qui pourrait contenir un caractère de retour à la ligne).
