LECTURE DE FICHIERS

1. Rappels de langage

Le programmeur n'accède pas directement aux fichiers, il demande au Système d'Exploitation de s'occuper de toute la gestion technique : pilotage des disques durs etc.

Pour ce faire, il suffit de créer un pointeur de type **`FILE *`** (en majuscules !), puis de demander au SE de créer un lien via ce pointeur. C'est le rôle de la fonction **`fopen`**. La lecture s'effectue à l'aide des fonctions **`fscanf`** et **`fprintf`**.

Lorsqu'on écrit un fichier, le plus simple est de mettre l'écriture dans une boucle **`for`**, en effet on connaît forcément le nombre d'éléments à écrire.

Par contre lors de la lecture, ce n'est pas le cas. Il faut alors être capable de savoir si l'on a atteint la fin du fichier. La fonction **`feof`** donne cette information. Elle s'utilise le plus souvent avec une boucle de type "tant que" : tant que (la fin du fichier n'est pas atteinte)

Enfin, une fois la lecture ou l'écriture finie, il faut fermer le fichier à l'aide de l'instruction **`fclose`**.

La syntaxe exacte des fonctions sus-citées est :

```
fopen( "chemin d'accès au fichier","mode"); // retourne une adresse de type FILE *
```

Le type d'accès est soit **`"r"`** pour la lecture, soit **`"w"`** pour l'écriture. Le chemin d'accès utilise des **`"/"`** même sous l'environnement Windows.

```
fprintf( variable_de_type_FILE, "chaine de format", liste de variables ) ;
fscanf( variable_de_type_FILE, "chaine de format", liste d'adresses de variables ) ;
feof( variable_de_type_FILE ) ;
fclose( variable_de_type_FILE) ;
```

2. Exercice

*Dans ce qui suit, toutes les fonctions à écrire seront de type **void**.*

Vous trouverez sur le bureau de votre poste de travail un fichier intitulé **`TP4.txt`**, si le fichier n'est pas présent, demandez-le à l'enseignant.

Ce fichier est constitué de deux colonnes **`X`** et **`Y`** (dans cet ordre) de **`n`** réels séparées par un espace.

★ Lisez le contenu du fichier **`TP4.txt`** dans votre programme en faisant un écho (affichage) à l'écran.

- ✦ Ne connaissant pas la longueur du fichier, vous créerez un type de variable pouvant contenir 1000 **`float`** appelé datas.
- ✦ Ecrivez une fonction qui lise le fichier **`TP4.txt`**, et détermine le nombre de points lus. Déterminez les paramètres d'entrée et de sortie de la fonction qui s'appellera **`lecture`**.
- ✦ A l'aide de cette fonction, lisez le contenu du fichier, puis dans la fonction **`main`**, affichez toutes les valeurs lues.

Le signal lu est bruité, afin de diminuer ce bruit, on procédera de la façon suivante : chaque fois qu'un des **`Y`** est supérieur à trois fois la moyenne du vecteur **`Y`**, on fixe la valeur de cet élément à zéro.

★ Filtrez le signal lu.

- ✦ Ecrivez une fonction appelée **`ecriture`** qui crée un fichier de type texte appelé **`filtre.txt`** et y écrive deux colonnes de valeurs, séparées par une tabulation, transmises en paramètres à la fonction. Déterminez les paramètres d'entrée et de sortie de la fonction.
- ✦ Créez une fonction qui calcule la valeur moyenne d'un vecteur. Cette fonction sera appelée **`moyenne`** vous déterminerez les paramètres d'entrée et de sortie.
- ✦ Calculez, dans la fonction principale la valeur moyenne des ordonnées lues (vecteur **`Y`**), puis appliquez l'algorithme décrit ci-dessus.
- ✦ Sauvegardez dans un fichier portant le nom filtre.txt les données ainsi traitées. Le fichier contiendra les vecteurs **`X`** et **`Y`** séparés par un espace.

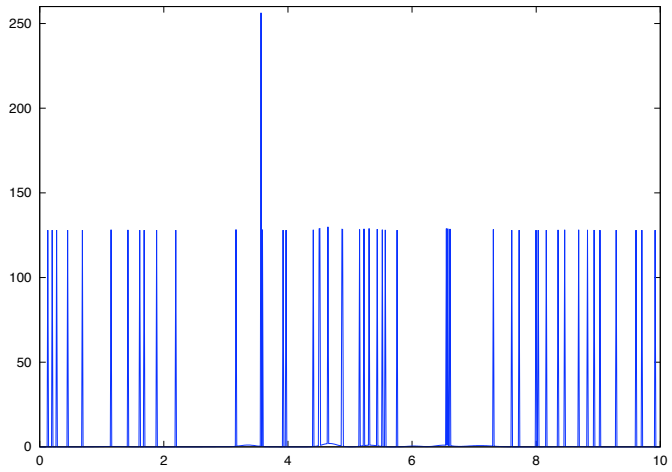
★ Lancez le logiciel Octave, puis dans la fenêtre qui s'ouvre tapez les commandes suivantes :

```
cd Desktop
M = load -ascii filtre.txt ;
x = M( : , 1 ) ;
y = M( : , 2 ) ;
plot(x,y)
```

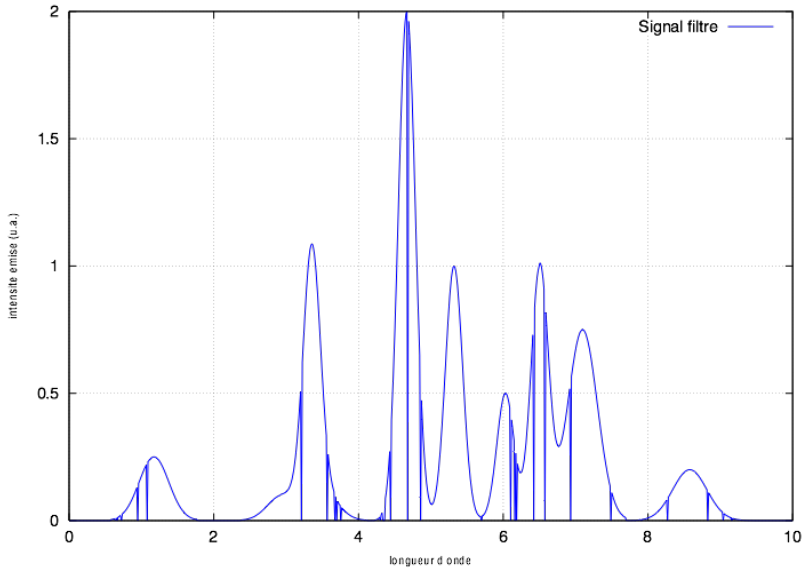
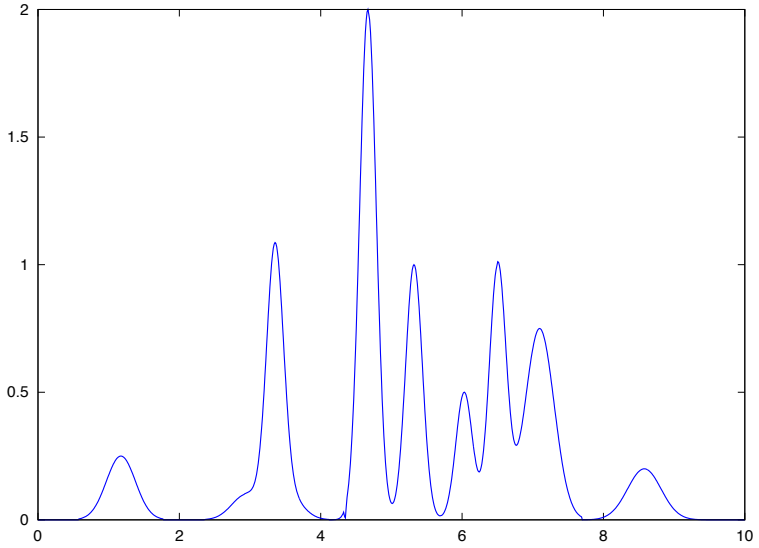
Une fenêtre graphique devrait s'ouvrir contenant la courbe des points que vous avez filtré.

★ On peut améliorer le filtrage en modifiant la procédure utilisée en utilisant un filtrage en deux étapes. On procède comme précédemment, si **`Yi`** est supérieur à trois fois la valeur moyenne de **`Y`**, alors on impose **`Yi`** à zéro, en même temps, dans un vecteur auxiliaire **`Z`**, préalablement initialisé à zéro, on impose **`Zi`** à un. Dans une deuxième phase, si **`Zi`** vaut 1 alors on remplace **`Yi`** par la moyenne des deux points environnants.

Signal non filtré



Signal filtré (mise à zéro)



Signal Filtré (deux passes)