

Examen de Techniques et Implémentation de Méthodes Numériques UE EMEAT1B1

Supports de cours et TP autorisés

Téléphones et autres appareils électroniques interdits

Durée : 2h00

Connexion : Connectez-vous avec le compte examen `ex_m1eea_xyz` où `xyz` est le numéro qui vous a été fourni. Le mot de passe est `jsupe2008`.

Directives : Il est vivement conseillé de traiter les questions les unes après les autres et de valider au fur et à mesure chacun de vos morceaux de programme rajoutés.

Travail demandé Il faut écrire en langage C différentes parties d'un même programme à compléter qui se trouve à la fin du sujet et qui vous sera donné par l'enseignant.

Il faut pour cela compléter le programme au fur et à mesure **en testant à chaque étape** que ce que vous avez réalisé correspond à ce qui est demandé!

Les en-têtes de fonction sont donnés! Respectez leur syntaxe!

A faire en premier :

1. Ouvrir un terminal.
2. Copier le répertoire `Examen-TIMN-17-18-session1` en tapant la commande suivante dans le terminal :

```
cp -r /home/partage/commun/M1_EEA_TIMN/Examen_TIMN_17-18_session1 .
```
3. Aller dans le répertoire `Examen_TIMN_17-18_session1`.
4. Editer le fichier `modele-examen-timn-17-18-session1.c`
5. Compiler avec `gcc -Wall -o prog modele-examen-timn-17-18-session1.c`
6. Exécuter avec `./prog`

Manipulation d'une équation récurrente... Nous souhaitons réaliser des manipulations sur des équations récurrentes.

Une équation récurrente entre une suite d'entrées E_k et une suite de sorties S_k peut être écrite sous la forme suivante :

$$b_n.s_{k+n} + \dots + b_0.s_{k+0} = a_m.e_{k+m} + \dots + a_0.e_{k+0}$$

Nous souhaitons stocker proprement les éléments a_i et b_j des 2 suites récurrentes de l'équation. Pour cela, nous utiliserons une structure composée de 4 champs :

- l'ordre m de la suite des entrées E_k ,
- l'ordre n de la suite des sorties S_k ,
- un tableau dynamique pour les coefficients a_i de la suite d'entrées E_k ,
- un tableau dynamique pour les coefficients b_j de la suite de sorties S_k ,

Questions

1. Remplir la déclaration de la structure `struct Equation_Recurrente{...}`; en vous inspirant de celle que vous aviez écrit pour les polynômes en TP.

Compilez, testez !

2. Ecrire la fonction `struct Equation_Recurrente Saisie_Affichage()` qui doit permettre de saisir manuellement les champs de la structure m , n , ainsi que les coefficients des séries E_k et S_K . Etant donné que les coefficients des séries sont stockés dans des tableaux dynamiques, il sera important de faire l'allocation dynamique de ces tableaux. Une fois que toute la saisie sera faite, un affichage "élégant" permettra de vérifier que les valeurs ont été bien saisies.

Complétez le programme principal pour qu'il utilise cette fonction.

Compilez, testez !

3. En automatique, l'écriture d'une équation récurrente peut servir à calculer le gain statique.

Soit l'équation : $b_1.s_{k+1} + b_0.s_k = a_0.e_k$

Ecrire la fonction `float Gain_Statique1(struct Equation_Recurrente)` qui calcule le gain statique à partir des coefficients de l'équation récurrente avec la formule suivante en évitant les divisions par 0 :

$$gain = \frac{\sum a_i}{\sum b_j}$$

Le gain calculé par la fonction est renvoyé au programme principal.

Complétez le programme principal pour qu'il utilise cette fonction.

Compilez, testez !

Vérifiez le calcul de cette fonction sur l'équation récurrente suivante :

$$5.s_{k+1} + 3.s_k = 10.e_k$$

4. On corrige l'équation récurrente en ajoutant un paramètre K variable afin d'avoir un gain statique est égal à 1 (dans le cas où un correcteur de type intégrateur a été introduit dans la boucle de commande).

On obtient l'équation : $b_1.s_{k+2} + (b_0 - b_1).s_{k+1} + (a_0.K - b_0).s_k = a_0.K.e_k$

Ecrire la fonction `float Gain_Statique2(struct Equation_Recurrente, float K)` qui calcule le gain statique. Complétez le programme principal afin de pouvoir prendre en compte cette nouvelle équation récurrente et vérifiez que le gain statique reste égal à 1 pour des valeurs de $K > 0$. Attention, le nombre d'éléments de la suite S_k a changé !

Compilez, testez !

```
//
//
//  Numero compte examen :
//  Nom Etudiant :
//  Prenom Etudiant :
//

////////// Examen TIMN - session 1 - 19 décembre 2017
#include <stdio.h>
#include <stdlib.h>

/* Nom de la structure à utiliser et à ne pas modifier */
struct Equation_Recurrente{

};

/* En-têtes des fonctions
   Il ne faut pas les modifier */

struct Equation_Recurrente Saisie_Affichage();
float Gain_Statique1(struct Equation_Recurrente);
float Gain_Statique2(struct Equation_Recurrente, float K);

int main(void)
{
    /* Le main est à faire évoluer au cours de l'examen */

    return 0;
}
```