

Travaux Pratiques: Programmation en C

Responsable: John SAMUEL

Année: 2017-2018

Contact: john.samuel@cpe.fr

Lisez bien tout le document avant de commencer à programmer.

Instructions

1. Tous les travaux pratiques sont basés sur les aspects que vous avez déjà appris pendant vos cours. Utilisez bien vos supports de cours.
2. Travaillez en binôme. N'oubliez pas de numéroter vos groupes (groupeA01, groupeB01,... etc.)
3. Il est obligatoire de citer toutes les sources (e.g. internet, groupes)
4. Les séances sont encadrées par 2 enseignants.
5. Ne pas utiliser des bibliothèques externes.
6. **Commenter vos programmes**

Exercice 2 (3 points)

Objectif:

Comprendre les pointeurs et les structures.

Questions

1. ★ Imaginez que vous gérez les notes de cinq étudiants. En utilisant uniquement les tableaux, écrivez un programme `etudiant.c` qui déclare, initialise et affiche les détails de ces cinq étudiants. Pour chaque étudiant, on est intéressé par son nom, son prénom, son adresse, et ses notes dans 2 modules (Programmation en C, Système d'exploitation).
2. ★ Ecrivez un programme `etudiant2.c` pour la gestion de données étudiantes (5 étudiants) en utilisant `struct`. Les détails pour chaque étudiant restent les mêmes comme pour la question 1(nom, prénom etc.) Vous pouvez utiliser `strcpy` (comme on a vu pendant les cours) pour initialiser les noms, prénoms etc.
3. ★★ Une couleur en format RGBA contient 4 valeurs : rouge (R), vert (G), bleu (B) et alpha (A). Chaque valeur (R, G, B, A) est un octet. Créez un programme `couleurs.c` en utilisant `struct`. Ensuite, créez et initialisez un tableau de 10 couleurs. Pensez à initialiser les couleurs en notation hexadécimale (r : 0xef, g : 0x78 etc.).
4. ★★ Réutilisez le programme `variables.c` (TP1) et écrivez un programme `ptrvariables.c` qui affecte et affiche les valeurs des variables de différents types de base (char, short, int, long int, long long int, float, double, long double) en utilisant leurs adresses. N'oubliez pas d'afficher l'adresse et la valeur de ces variables avant ou après la manipulation par leurs adresses (en utilisant les pointeurs).
5. ★★★ Créez un programme `tableauptr.c` et écrivez le code pour créer deux tableaux différents de types int et float. Remplissez les valeurs aléatoires (vous pouvez utiliser la fonction `rand()` déclarée en `stdlib.h`). Pour les deux tableaux, si l'indice est divisé

par 2, multipliez la valeur à cette position par 3. N'utilisez pas la notation indicielle pour parcourir les tableaux (e.g, `i [3]`, `i [5]` etc.). Utilisez les pointeurs.

6. ★★★Ecrivez un programme `chercher.c` qui crée un tableau de 10 phrases (un tableau de tableau de caractères). Sans utilisant les fonctions de bibliothèques standards ou externes, cherchez si une phrase est dans le tableau ou pas. (BONUS, si vous utilisez `char **`, c'est-à-dire l'adresse d'une variable qui contient l'adresse d'une variable `char`)

Fichiers

`etudiant.c`, `etudiant2.c`, `couleurs.c`, `ptrvariables.c`, `tableauptr.c`, `chercher.C`

Astuces

1. Pour la question 5, pensez à utiliser `*(s+i)` etc (e.g, si `s` est un pointeur de type `int *`).

Compte rendus

Compte rendu en format `.zip` en un seul fichier.