

TP - structures

Objectif(s)

- ★ Un exercice de révision sur les boucles.
- ★ Première utilisation des structures en C.

Exercices obligatoires

Exercice 1

Question 1

Écrivez un programme qui demande un nombre de valeurs à saisir à l'utilisateur, permet la saisie de ces valeurs, et enfin affiche la plus grande valeur saisie ainsi que son rang. Si l'utilisateur annonce 6 valeurs saisies, et saisit 45 6 35 345 462 24, le programme affichera : *La plus grande valeur est 462, c'est la 5 ème valeur de la suite.*

Question 2

Vous avez du utiliser une boucle `for ...` Sinon, modifiez votre programme pour utiliser cette boucle.

Question 3

Même exercice, mais l'utilisateur ne saisit plus le nombre de valeurs avant, il termine la suite par un -1. Écrire un programme qui demande à l'utilisateur de saisir au clavier une suite de nombres. La série s'arrêtera dès qu'un nombre négatif sera saisi. Le programme affichera alors le plus grand nombre saisi.

Exercice 2 – Les structures en C

Question 1

Écrivez une structure `struct date` pour définir une date du type "10 octobre 2024". On représentera les jours et les mois par des entiers pour l'instant, entre 1 et 31 pour le jour selon les mois, entre 1 et 12 pour le mois.

Question 2

Écrivez une fonction dont le prototype est `struct date cree_date(int, int, int)` qui prend 3 entiers respectivement le numéro du jour, le numéro du mois et l'année et qui retourne une `struct date` remplie avec les valeurs. On suppose que les valeurs données sont correctes.

Question 3

Écrivez une fonction `void affiche_date(struct date)` qui affiche la date avec le format JJ / MM /
↪ AAAA.

Testez vos fonctions `date cree_date` et `affiche_date`.

Question 4

Écrivez une fonction qui prend en entrée deux dates. La fonction renvoie 1 si les deux dates sont égales, 0 sinon. Testez votre fonction avec au minimum 3 jeux d'essai bien choisis..

Question 5

Écrivez une fonction `nombre_jour_mois` qui prend en paramètre le numéro d'un mois et son année et retourne le nombre de jour du mois. On pourra réutiliser la fonction `est_bissextile` du TD1, qui renvoyait un `int`, 0 pour faux et 1 pour vrai.

Testez votre fonction avec au minimum 5 jeux d'essai bien choisis.

Question 6

Écrivez une fonction qui vérifie qu'une date est valide. La fonction renverra 1 si la date est valide sinon 0. Pour aller plus loin : le calendrier grégorien fait suite au calendrier julien. Il a été mis en place le 15 octobre 1582.

Testez votre fonction avec au minimum 5 jeux d'essai bien choisis.

Question 7

À partir d'une date donnée, écrivez une fonction qui calcule la date du lendemain.

Renforcements

Exercice 3 – Date

Question 1

Écrivez une fonction qui indique le nombre de jours d'écart entre deux (struct) dates $d1$ et $d2$ passées en paramètre. Si $d1 < d2$ le nombre de jours sera un nombre positif, et négatif dans le cas contraire.

Question 2

La journée de la semaine sera codée de la manière suivante : dimanche \rightarrow 0, lundi \rightarrow 1, ... samedi \rightarrow 6, En utilisant la date du jour et la journée du jour comme référence, écrivez une fonction qui prend en paramètre une date et retourne le numéro de la journée correspondante.

Question 3

Écrivez un programme qui demande votre jour de naissance et indique le jour exact de cette date.

Question 4 Calculer le jour de la semaine d'une date

Le site https://fr.wikibooks.org/wiki/Curiosités_mathématiques/Trouver_le_jour_de_la_semaine_avec_une_date_donnée propose différentes méthodes pour calculer le jour de la semaine à partir d'une date. Nous proposons d'utiliser la méthode 2.

Extrait :

Pour une date de la forme jour/mois/année où « jour » prend une valeur de 01 à 31, « mois » de 01 à 12 et « année » de 1583 à 9999, utiliser la formule :

$$c = (14 - \text{mois})/12$$

En fait, $c = 1$ pour janvier et février, $c = 0$ pour les autres mois.

$$a = \text{année} - c$$

$$m = \text{mois} + 12 * c - 2$$

$$j = (\text{jour} + a + a/4 - a/100 + a/400 + (31 * m)/12) \bmod 7$$

La réponse obtenue pour j correspond alors à un jour de la semaine suivant :

0 = dimanche, 1 = lundi, 2 = mardi, etc.

Écrivez une fonction `jour_semaine` qui prend en entrée une date et calcule le jour de la semaine.