

TD3 de Programmation C: itérations

L1 MIASHS, 2022-2023

Le but de ce TD est de réaliser différents programmes mettant en œuvre des itérations (avec `for` ou `while`), et bien sûr en continuant à utiliser des conditionnelles.

Exercice 1 (100 premiers entiers) Écrire les variantes suivantes d'un programme qui affiche les entiers de 1 à 100, les différences portant uniquement sur l'affichage :

Activité 1.1 Tous les nombres sont sur une même ligne, avec juste un espace pour les séparer.

Activité 1.2 Tous les nombres sont sur une même ligne, chacun étant formaté sur 4 caractères.

Activité 1.3 En plus, afficher un changement de ligne tous les 10 nombres, pour obtenir l'affichage suivant :

1	2	3	4	5	6	7	8	9	10
11	12	13	14	15	16	17	18	19	20
21	22	23	24	25	26	27	28	29	30
...									
91	92	93	94	95	96	97	98	99	100

Exercice 2 (Sélection de nombres) Afficher tous les entiers entre 1 et 100 qui sont divisibles par 7 ou par 13, puis ceux qui sont divisibles par 7 et par 13.

Exercice 3 (Factorielle) Calculer la valeur de $n!$ (factorielle n), la valeur de l'entier n étant demandée à l'utilisateur (et doit être positive ou nulle).

Remarque : pourquoi le résultat, si représenté par un entier, est-il étrange à partir d'une certaine valeur de n ?

Exercice 4 (Somme et moyenne) Calculer la somme et la moyenne de n nombres réels, la valeur de n étant demandée à l'utilisateur (tout comme chaque nombre réel).

Exercice 5 (Le nombre mystère) L'objectif est d'écrire un programme qui permet à l'utilisateur de tenter de retrouver un nombre entre 0 et 49 en un minimum de tentatives. Les activités suivantes vous permettent d'écrire progressivement ce programme.

Activité 5.1 Dans votre programme, incluez les bibliothèques `stdlib.h` et `time.h` à la suite de l'inclusion de `stdio.h`, pour avoir accès aux fonctions ci-dessous.

Dans votre programme, utilisez les instructions suivantes pour choisir un nombre aléatoire, stocké dans la variable `secret` :

```
srand(time(NULL)); /* initialisation à l'aide de l'heure actuelle */
secret = rand() % 50; /* calcul de la valeur aléatoire */
```

`srand` permet d'initialiser la fonction aléatoire (avec l'heure courante), pour ne pas retomber à chaque exécution sur les mêmes valeurs ; `rand` choisit un entier positif aléatoirement, l'opération modulo 50 permet d'avoir un résultat entre 0 et 49.

Activité 5.2 Compléter le programme en demandant à l'utilisateur de saisir une valeur entre 0 et 49, et boucler sur la demande tant qu'il n'a pas saisi une valeur comprise dans cet intervalle.

Activité 5.3 Compléter le programme en affichant un message indiquant si la valeur de l'utilisateur est trop petite ou trop grande par rapport au `secret`.

Activité 5.4 Compléter le programme pour boucler sur les tentatives de l'utilisateur jusqu'à ce qu'il trouve le nombre secret.

Activité 5.5 Compléter le programme pour qu'à la fin, le nombre de tentatives soit affiché avec un message approprié selon si il a trouvé en une seule tentative, en au plus 5, ou en plus de 5.

Exercice 6 (Rectangle) Écrire un programme qui affiche un rectangle de dièses dont la hauteur et la largeur sont demandées à l'utilisateur.

Exemple de hauteur 7 et largeur 4 : #####

#####

Exercices bonus

Exercice 7 (Factorielles) Lors de l'exercice du calcul de $n!$ (factorielle n), vous avez remarqué qu'à partir d'une certaine valeur de n , le résultat n'est pas correct. Écrire le programme qui calcule factorielle n sous deux formes : une première fois sous forme d'un entier, une seconde fois sous forme d'un réel. Afficher à partir de quelle valeur de n les résultats sont différents.

Exercice 8 (Matrice) Imaginez (sans la calculer) que vous dispose d'une matrice carrée de côté c , dont chaque case contient pour valeur la somme de son numéro de ligne et de son numéro de colonne. Écrire un programme qui calcule le nombre de valeurs dans la matrice qui sont divisibles par k (c et k sont saisis au début).