



Algorithmique et langage C

TP n°3

Tests et boucles

Amina Nahali
anahali@ece.fr

Préparation des exercices

Première utilisation du débogueur

Copiez-collez le code source suivant dans Codeblocks, ajoutez les points d'arrêts ( `int varA, varB = 0;`) demandés et testez le **mode debug** ().
Testez notamment les boutons :  et . **Que font-ils ? Que permet de voir le débogueur ?**

```
#include <stdio.h>
#define MAX_I 4
#define MAX_J 3

int main() {
    int varA = 0, varB = 0;
    int i = 0, j = 0, cpt = 0;
    for(i = 1; i < MAX_I; i++) {
        for(j = 1; j < MAX_J; j++) {
            // Mettre un point d'arrêt à la ligne suivante :
            varA = i * j;
            // Mettre un point d'arrêt à la ligne suivante :
            varB = i + j - i;
            // Mettre un point d'arrêt à la ligne suivante :
            printf("Ceci est le %dème printf !\n", ++cpt);
        }
    }
    return 0;
}
```

Exercice 1

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

- Saisir deux entiers au clavier
- Afficher le plus grand des deux dans la console

Exercice 2

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

- Saisir la largeur et la longueur d'un rectangle au clavier
- Afficher le périmètre et l'aire de ce rectangle dans la console

Exercice 3

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

- Saisir un entier au clavier
- Indiquer dans la console si cet entier est un multiple de 3 et s'il est supérieur ou égal à 10.

3 devra être stocké dans une constante (objectif pédagogique uniquement). Essayez de modifier la valeur de la constante et constatez l'erreur.

Exercice 4

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

Un cinéma voudrait un logiciel permettant de gérer sa billetterie. Nous sommes en charge de la partie tarification (réductions en fonction de l'âge). Cette partie affiche le prix du billet, en fonction de l'âge et du statut d'étudiant.

- de 0 à 12 ans exclus : tarif enfant (4 €)
- de 12 à 17 ans, ou étudiant de moins de 27 ans : tarif jeune (6 €)
- à partir de 65 ans : tarif senior (6 €)
- sinon : plein tarif (9 €)

*Quelles valeurs pourrions-nous transformer en **macros** ? Modifiez votre code en conséquence.*

Exercice 5

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

Un distributeur propose trois boissons froides (1, 2 et 3) et deux boissons chaudes (10 et 11). 

En service, il demande à l'utilisateur.rice le numéro de la boisson et affiche le nom de celle-ci dans la console.

Si l'utilisateur.rice tape un numéro qui ne correspond à aucune boisson, le distributeur devra indiquer un message d'erreur.

Vous avez le choix des boissons.

Exercice 6

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

- Saisir trois notes sur 20 (ex : 13,4) au clavier
- Calculer la moyenne des notes en prenant soin de vérifier que chaque note soit bien comprise entre 0 et 20 (une nouvelle note est demandée à l'utilisateur tant qu'elle n'est pas conforme).

Exercice 7

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

Dans une école primaire, le/la directeur.rice doit connaître le nombre total d'élèves.

Pour cela, il/elle saisit le nombre de classes ouvertes, puis ensuite, pour chacune des classes créées, son nombre d'élèves.

Le programme affichera à la fin le nombre total d'élèves dans l'école.

Exercice 8

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

- Saisir un nombre entier
- Tant que ce nombre n'est ni un multiple de 7, ni un multiple de 2, le programme lui demande de le resaisir.
- 💡 Aide slide suivante.

Exercice 8 - Aide

Nous souhaitons obtenir un nombre à la fois multiple de 2 et de 7 :

```
(n % 2 == 0 && n % 7 == 0)
```

La boucle **while** va donc se répéter tant qu'on n'a pas ce qu'on veut :

Donc l'inverse de ce que l'on veut.

```
while( !(n % 2 == 0 && n % 7 == 0) )
```

Ce que l'on peut simplifier en :

Voir les équivalences remarquables : $\neg(A \wedge B) \equiv \neg A \vee \neg B$.

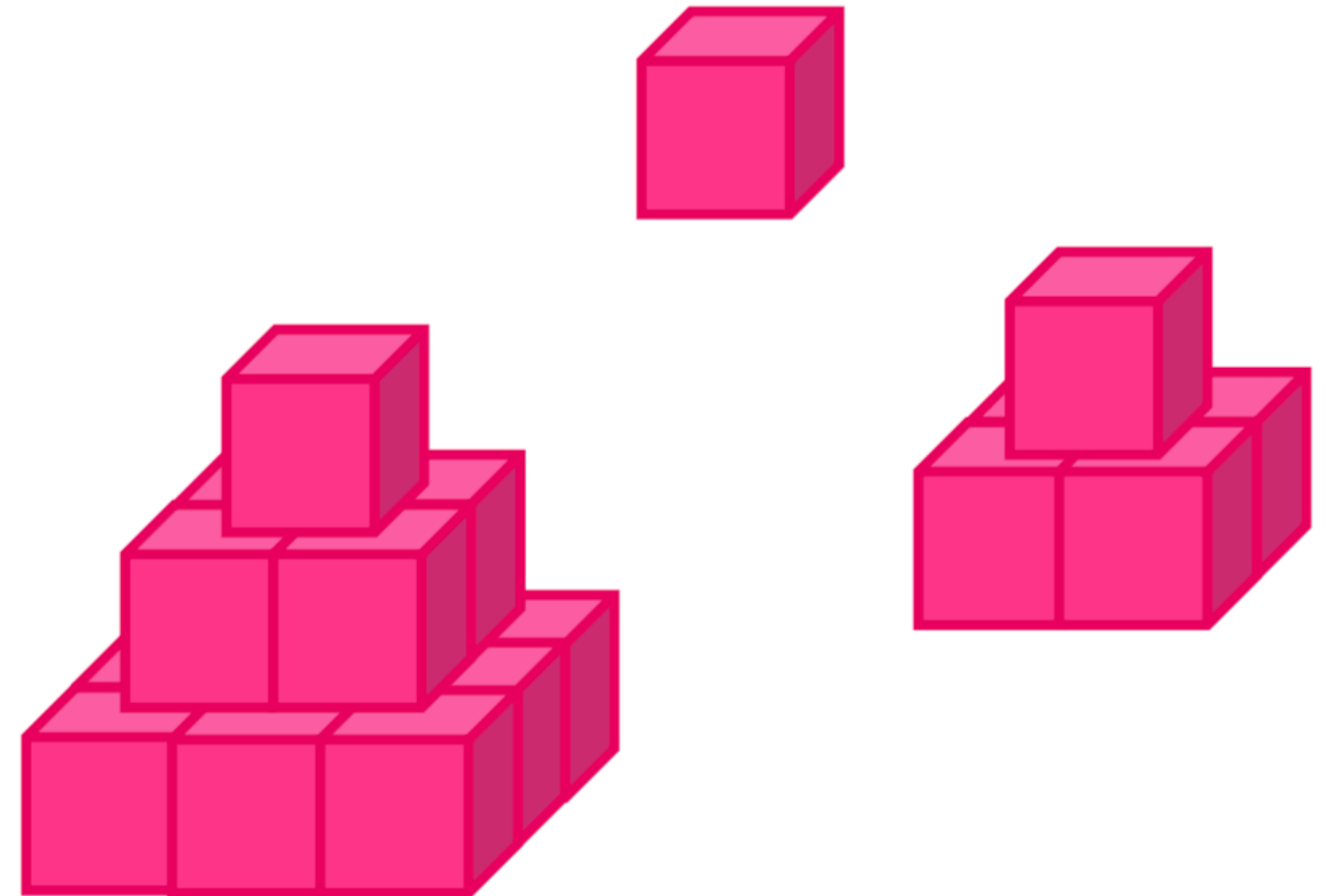
```
while( n % 2 != 0 || n % 7 != 0 )
```

Exercice 9

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

Dans une ville, les habitants décident de construire une pyramide. Ils souhaitent savoir combien d'étages ils pourront construire avec le nombre de pierres qu'ils possèdent.

Le programme demande de saisir le nombre de pierres disponibles, et affiche le nombre d'étages qu'il sera possible de construire avec.



Exercice 10

Réalisez un algorithme et le programme correspondant permettant de satisfaire le cahier des charges suivant :

Nous souhaitons concevoir un programme permettant de calculer la moyenne de nombres entiers positifs saisis au clavier.

L'utilisateur.rice saisit des nombres positifs au clavier, puis pour indiquer la fin, saisit un nombre négatif (condition pour finir la saisie). L'ordinateur affiche alors la moyenne des nombres saisis dans la console.

Vous penserez à tous les cas possibles et testerez votre algorithme et votre programme en effectuant leur trace d'exécution (débugueur Codeblocks).