**Langage de programmation**

**HAUTE ÉCOLE DE NAMUR-LIÈGE-LUXEMBOURG**

**Bloc 1**

Exercice 3 – Répétitives

Objectifs

* Utiliser les notions de bases vues : expressions, instructions, fonction principale, alternatives, répétitives…
* Définir une constante symbolique
* Utiliser putchar et getchar à bon escient
* Appliquer les conseils du Clean Code

Veiller à la portabilité du programme

# Introduction

Les exercices de Langage de programmation ont pour objectif de vous permettre de mettre en pratique les différentes notions vues dans les ateliers et lors des séances de mise en commun.

N’hésitez pas à travailler en dehors des séances d’exercices et avec les autres étudiants !

**Note.**

Dans les exemples qui suivent, les passages en texte normal sont à sortir tels quels.

Les parties en *italique souligné* correspondent aux entrées de l'utilisateur.

Les portions en **gras** varient en fonction des entrées.

**Il est vivement conseillé de commencer par un DA avant de vous lancer dans le code !**

# Exercice 1 : le plus petit et le plus grand

Comme vous allez le voir en « Architecture des ordinateurs », à chaque type correspond une valeur maximum et une valeur minimum. Ces valeurs sont définies sous la forme de constantes symboliques situées dans des fichiers d’entête faisant partie des librairies standards.

Cherchez le nom du fichier d’entête qui contient les définitions des constantes entières.

Donnez les noms des deux constantes associées aux valeurs maximum et minimum du type int.

Au cas où vous ne sauriez pas comment chercher, voici une idée de mots à proposer au moteur de recherche de votre navigateur : « librairie standard c ».

Faites de même pour le type double. Commencez par trouver le fichier d’entête.

Ces constantes symboliques seront régulièrement utilisées pour traduire certains algorithmes.

Pour la double librairie <float.h> et pour le int <limits.h>

# Exercice 2 : tables de multiplication

Écrivez un programme qui, étant donné un nombre naturel obtenu de l’utilisateur, affiche la table de multiplication (de 1 à 10) de ce nombre.  
À l'écran doit apparaitre, par exemple :

Nombre : *7*

**1 x 7 = 7  
2 x 7 = 14  
…  
10 x 7 = 70**

# Exercice 3 : tables de multiplication (bis)

Écrivez un programme qui affiche la table de multiplication (de 1 à 10) des nombres 5 à 7.

# Exercice 4 : tables de multiplication (ter)

Écrivez un programme qui affiche la table de multiplication (de 1 à 10) de chaque naturel obtenu de l’utilisateur (0 pour terminer).

# Exercice 5 : visite de château

Le tarif de visite d’un château est le suivant :

|  |  |
| --- | --- |
| Adulte (de 18 à 64 ans) | 7,50 euros |
| Senior (65 ans et plus) | 6,50 euros |
| Adolescent (de 12 à 17 ans) | 4 euros |
| Enfant (moins de 12 ans) | Gratuit |

Sachant que, pour une famille souhaitant visiter le château, on demande et récupère l’âge de chaque membre de la famille (-1 pour terminer), écrivez le programme qui calcule et affiche le prix total à payer par la famille pour effectuer la visite.

# Exercice 6 : le plus jeune et le plus vieux

Écrivez un programme qui, sur base de l’âge d'un certain nombre d'étudiants, obtenus de l’utilisateur, affiche l'âge du plus jeune ainsi que l'âge du plus âgé.

# Exercice 7 : obtention valide nombre positif

Écrivez le programme qui demande et récupère un **nombre entier**. Son contenu doit normalement être un nombre positif. Votre programme affiche "Erreur…" tant que le nombre ne respecte pas la condition. Lorsque le nombre est correcte le message "Correct ! " est affiché.

À l'écran doit apparaitre, par exemple :

Nombre : *-3*  
**Erreur...**Nombre : *5*  
**Correct !**

# Exercice 8 : concours

Les organisateurs d’un concours ont décidé d’attribuer les prix de la façon suivante :

Les 10 premiers reçoivent une prime. Celle-ci est de 400 € pour le premier, 300 € pour le deuxième. À partir du troisième, le montant est diminué à chaque fois de 20 € (le troisième reçoit 280 €, le quatrième 260 €…).

À partir de la onzième place, le lot de consolation est un t-shirt.

Sachant que l’on introduit pour chaque coureur, son identifiant (un code entier, -1 pour terminer) et sa place dans la course, écrivez le programme permettant d’afficher l’identifiant et la prime à recevoir pour les 10 premiers ainsi que le nombre de t-shirts à offrir.

# Exercice 9 : carrés de pairs

Étant donné un nombre entier n strictement positif obtenu de l’utilisateur, faites imprimer les n premiers nombres pairs et leur carré.  
Exemple :

Nombre : *6*  
Nombres pairs Carrés

**2 4**

**4 16**

**6 36**

**8 64**

**10 100**

**12 144**