**Langage de programmation**

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

**Bloc 1**

Exercice 2 – Alternatives

Objectifs

* Utiliser les notions de bases vues : expressions, instructions, fonction principale, alternatives…
* 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.

# Exercice 1 : Walibi

Dans un parc d’attractions, certaines attractions ne sont pas accessibles aux personnes dont la taille est soit inférieure à une taille minimum soit supérieure à une taille maximum.

Écrivez un programme qui demande et récupère la taille d’une personne et détermine si cette personne est autorisée à accéder à l’attraction. Les différentes tailles sont exprimées en centimètres.

La variable tailleMin contient la taille minimum et la variable tailleMax, la taille maximum. Commencez par déclarer et affecte une valeur sensée à ces deux variables (par exemple 100 et 200).

Ce programme affiche le message « Accès refusé : taille trop petite » ou « Accès refusé : taille trop grande » ou « Accès autorisé ».

À l'écran doit apparaitre, si la taille est trop petite, par exemple :

Taille : 9*5*  
**Accès refusé : taille trop petite**

Ou, dans le cas où la taille est trop grande, par exemple :

Taille : 210  
**Accès refusé : taille trop grande**

Ou, dans le cas où la taille est dans les limites autorisées :

Nombre : *120*  
**Accès autorisé**

# Exercice 2 : signe du produit

Écrivez un programme qui, étant donnés deux **nombres réels** obtenus de l'utilisateur, permet de déterminer le signe du produit (positif, négatif ou nul), sans effectuer l'opération mais en se basant sur les signes de chacun des deux nombres.

À l'écran doit apparaitre, par exemple :

Opérande 1 : *25.5*

Opérande 2 : *33.8*

**Produit positif**

Ou, par exemple :

Opérande 1 : *25.5*

Opérande 2 : *-8*

**Produit négatif**

Ou, par exemple :

Opérande 1 : *-25.5*

Opérande 2 : *-8*

**Produit positif**

# Exercice 3 : Flamenco (sans AND/OR)

Un centre culturel organise un atelier « Flamenco » dont le tarif annuel est de

* 100 euros pour les enfants de 6 à 11 ans,
* 120 euros pour les jeunes de 12 à 18 ans,
* 160 euros pour les personnes de plus de 18 ans.

Écrivez le programme qui détermine et affiche le prix à payer par une personne qui s’inscrit à l’atelier et dont on obtient l’âge.  
Remarque : il faut 6 ans minimum pour s’inscrire et si ce n’est pas le cas, votre programme doit afficher un message le signalant.

# Exercice 4 : somme

Écrivez un programme qui demande et récupère 3 **nombres entiers**.  
Ce programme affiche "Correct !" si le troisième nombre entré est bien la somme des deux premiers ; sinon, il affiche un message d'erreur et la bonne solution…

À l'écran doit apparaitre, si la somme est correcte, par exemple :

Nombre 1 : *15*  
Nombre 2 : *21*  
Somme : *36*

**Correct !**

Ou, dans le cas où la somme est erronée, par exemple :

Nombre 1 : *15*  
Nombre 2 : *21*  
Somme : *35*

**Incorrect, vous auriez du repondre 36 !**

# Exercice 5 : en prévision de la suite…

Écrivez le programme qui demande et récupère un **nombre entier**. Son contenu doit normalement être un nombre positif. Votre programme affiche respectivement "Correct !" ou "Erreur…" suivant la valeur de l'entier.

À l'écran doit apparaitre, si le nombre est correct, par exemple :

Nombre : *5*  
**Correct !**

Ou, dans le cas où la nombre est erronée, par exemple :

Nombre : *-3*  
**Erreur...**

Ou encore :

Nombre : *0*  
**Erreur...**