**Langage de programmation**

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

**Bloc 1**

Exercice 5 – Fonctions

Objectifs

* Utiliser les notions de bases vues : expressions, instructions, fonction principale, alternatives, répétitives…
* Définir des fonctions et les appeler
* 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 : pair ou impair

Écrivez la fonction estPair qui, sur base d'un entier reçu en argument, détermine s'il est pair ou non et renvoie cette information à la fonction appelante.

Écrivez l’appel de la fonction dans la fonction principale…

# Exercice 2 : qu'y a-t-il au menu ?

Écrivez la fonction (en fait c'est une procédure) afficheMenu qui a pour but d’afficher le menu suivant :

1. Nouveau personnage   
2. Nouvelle partie   
3. Partie en cours   
4. Quitter

Écrivez une fonction choixLu qui a pour but d’obtenir de l’utilisateur, le numéro de l’option choisie et de le transmettre à la fonction appelante. Elle fait appel à la fonction afficheMenuavant l’obtention du choix.

La fonction principale doit faire appel à la fonction choixLu et afficher le message adéquat ("Créer un nouveau personnage ", "Commencer une nouvelle partie", "Continuer la partie sauvegardée") ou quitter le programme.

# Exercice 3 : afficher les dégâts

Écrivez la fonction afficheDégats qui reçoit deux arguments, les points de dégât du personnage (entier) et un modificateur (réel <= 1) *déterminé selon la distance de la cible*, fait le produit de ces deux valeurs et affiche le résultat.

La fonction principale, après avoir obtenu les deux nombres, fait appel à la fonction afficheDégats.

# Exercice 4 : calculer les dégâts

Écrivez la fonction dégatsEffectifsqui reçoit deux arguments, les points de dégât du personnage (entier) et un modificateur (réel <= 1) *déterminé selon la distance de la cible*, fait le produit de ces deux valeurs et renvoie le résultat à la fonction appelante.

La fonction principale, après avoir obtenu les deux nombres, fait appel à la fonction dégatsEffectifset affiche les points de dégâts effectifs.

# Exercice 5 : somme des carrés

Imprimez la somme des carrés des nombres compris entre 1 et un nombre obtenu (entier positif) de l'utilisateur. Pour cela, écrivez les trois fonctions/procédures suivantes et faites-y appel dans la fonction principale.

Avant de rentrer dans la phase de codage, écrivez la précondition et la postcondition de chacune des fonctions, ainsi que les éventuels paramètres formels et résultats.

## Obtenir un nombre valide

Écrivez une fonction qui permet d’obtenir un entier valide.

Prototype :

Pré :

Post :

## Calculer la somme des carrés…

Écrivez une fonction qui permet de calculer la somme des carrés des entiers compris entre 1 et le nombre reçu en argument.

Prototype :

Pré :

Post :

## Afficher

Écrivez une fonction qui affiche à l'écran le message suivant :

"La somme des carrés des entiers compris entre 1 et ", <nombre obtenu>, " est ", <somme des carrés>, "."

Prototype :

Pré :

Post :

# Exercice 6 : The Voice…

Le but de cet exercice est de définir quelques fonctions génériques de statistique et de les utiliser dans un programme.

**Commencez par faire un diagramme d'actions et faites-le valider par le professeur !**

La première phase (étape 1 à 6) est de définir ces différentes fonctions. Elles permettent de présenter le contenu d’un tableau statistique (c’est-à-dire un tableau d’entiers), soit sous la forme d’un texte (répétitions ou fréquences) soit sous la forme d’un diagramme en bâtonnets.  
La taille maximale de tous les tableaux est déterminée par une constante symbolique TAILLE\_MAX.

## Étape 1 : 1re présentation : nombre pour chaque option

Écrivez la fonction afficheTableau selon la spécification suivante.

┌────────────────┐↓ tabStat, taille

│ afficheTableau │

└────────────────┘

*Précondition :* tabStat[0..taille-1] est un tableau statistique.

*Postcondition* : le contenu de tabStat[0..taille-1] a été affiché au format suivant :

Une image contenant capture d’écran, horloge, nombre, Police

Description générée automatiquement

Option 1 : 7

Option 2 : 12

Option 3 : 19

Option 4 : 0

Option 5 : 2

Top-down et bottom-up…

On peut concevoir les fonctions en partant de la « grosse » fonction puis en construisant des fonctions annexes qui s’occupent de problèmes de plus en plus petits/faciles/abordables. C’est l’approche top-down. Pour l’implémentation, on peut préférer commencer par coder les « petites » fonctions d’abord puis passer ensuite aux fonctions qui les utilisent. C’est l’approche bottom-up.

## Étape 2 : 2e présentation : pourcentage pour chaque option

Écrivez la fonction afficheTableauFrequences selon la spécification suivante.

**Attention**: ellese sert de la fonction frequences décrite ci-après.

┌──────────────────────────┐↓ tabStat, taille

│ afficheTableauFrequences │

└──────────────────────────┘

*Précondition* : tabStat[0..taille-1]est un tableau statistique

*Postcondition* : le contenu de tabStat[0..taille-1] a été affiché au format suivant:

Une image contenant capture d’écran, horloge, nombre, Police

Description générée automatiquement

Option 1 : 17,50%

Option 2 : 30,00%

Option 3 : 47,50%

Option 4 : 0,00%

Option 5 : 5,00%

## Étape 3 : fonction auxiliaire de calcul de fréquences

Écrivez la fonction frequences selon la spécification suivante.

**Attention**: ellese sert de la fonction effectifTotal décrite ci-après.

┌────────────┐↓ tabStat, taille

│ frequences │

└────────────┘↓ tabFrequences

*Précondition* : tabStat[0..taille-1] est un tableau statistique

*Postcondition* : tabFrequences[0..taille-1] est le tableau des fréquences qui correspond à tabStat[0..taille-1]

Une image contenant horloge, capture d’écran, nombre, Police

Description générée automatiquement Une image contenant capture d’écran, horloge, Police, nombre

Description générée automatiquement

## Étape 4 : fonction auxiliaire de calcul de l’effectif total

Écrivez la fonction effectifTotal selon la spécification suivante.

┌───────────────┐↓ tabStat, taille

│ effectifTotal │

└───────────────┘↓ effectifTot

*Précondition* :tabStat[0..taille-1] est un tableau statistique

*Postcondition* : effectifTot est la somme des valeurs de tabStat[0..taille-1]

## Étape 5 : 3e présentation : diagramme en bâtonnets

Écrivez la fonction afficheBatons selon la spécification suivante.

***Attention***: ellese sert de la fonction valeurMaximale décrite ci-après.

┌───────────────┐↓ tabStat, taille

│ afficheBatons │

└───────────────┘

*Précondition* : tabStat[0..taille-1] est un tableau statistique

*Postcondition* : le contenu de tabStat[0..taille-1] a été affiché au format suivant

Une image contenant horloge, capture d’écran, nombre, Police

Description générée automatiquement

Option 1 : \*\*\*\*\*\*\*\*\*

Option 2 : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option 3 : \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Option 4 :

Option 5 : \*\*

**Remarque : la plus grande barre fera 50 étoiles de long.**

## Étape 6 : fonction auxiliaire qui détermine le maximum d’un tableau d’entiers

┌────────────────┐↓ tabStat, taille

│ valeurMaximale │

└────────────────┘↓ max

*Précondition :* tabStat[0..taille-1] est un tableau statistique

*Postcondition* : max est la valeur maximale de tabStat[0..taille-1]

Une image contenant horloge, capture d’écran, nombre, Police

Description générée automatiquement

La seconde phase (étape 7 à 11) permet d’utiliser ces fonctions dans un cas concret. En effet, à l'occasion de la prochaine saison de The Voice Belgique, la production désire utiliser un petit programme permettant aux spectateurs de voter pour la prestation qu’ils ont préférée (numérotées de 1 à N) lors de l’enregistrement du jour (hors « live »).

À la sortie de la salle d’enregistrement, un ordinateur est disponible. Dans un premier temps, le responsable du plateau rentrera le nombre d’artistes qui ont participé à cet enregistrement.

Ensuite, chaque spectateur pourra entrer le numéro de sa prestation préférée (sur base de la liste des noms d’artistes ou groupes et du titre interprété).

Une fois tous les votes entrés, l'administrateur pourra obtenir l'affichage des résultats (a) sous forme numérique, (b) sous forme de fréquences et (c) sous forme d’un diagramme en bâtons.

Pour cela, suivez les étapes suivantes…

## Étape 7 : obtenir le nombre de groupes

Écrivez la fonction permettant d’obtenir un nombre de groupes valide !

┌─────────────────┐

│ nbGroupesValide │

└─────────────────┘↓ nbGroupes

*Précondition* : /

*Postcondition* : nbGroupes est le nombre de groupes entré par l'administrateur, vérifiant   
2 nbGroupes TAILLE\_MAX

## Étape 8 : initialiser le tableau

Écrivez la fonction qui initialise le tableau de statistique.

┌─────────────┐↓ nbGroupes

│ initTabStat │

└─────────────┘↓ tabStat

*Précondition* : nbGroupes 2

*Postcondition* : tabStat est un tableau statistique initialisé avec 0 vote pour chacun des nbGroupes groupes

## Étape 9 : obtenir un vote valide

Écrivez la fonction qui permet d’obtenir un vote valide.

┌────────────┐↓ nbGroupes

│ voteValide │

└────────────┘↓ vote

*Précondition* : nbGroupes 2

*Postcondition* : vote est le numéro du groupe entré par le spectateur, vérifiant 1 vote nbGroupes ou vote = -1 (le code pour mettre un terme aux votes)

## Étape 10 : obtenir les votes

Écrivez la fonction qui met à jour le tableau de statistiques sur base des votes obtenus… Cette fonction fait appel à voteValide.

┌────────┐↓ tabStat, nbGroupes

│ saisie │

└────────┘↓ tabStat

*Précondition* : tabStat[0..nbGroupes-1] est un tableau statistique et nbGroupes 2

*Postcondition* : tabStat est mis à jour en fonction des votes entrés par les différents spectateurs

## Étape 11 : fonction principale

Écrivez la fonction principale sur base de tout ce qui a déjà été précisé !