

**Sciences des organisations
L2 - Semestre 4**

**UE Y06 - Sciences du digital
(Informatique)**

Exercices de TD

Lucie GALAND
Florian SIKORA

TD 1 - Variables

Exercice 1 : Variables VBA

Q1.1 Quels sont les affichages réalisés par `MsgBox` lors de l'exécution des instructions suivantes ?

```
Dim a As Byte, b As Integer, c As Long
Dim d As Single, e As Double
Dim str1 As String, str2 As String

a = 200
b = 2
c = -3
MsgBox "a = " & a
MsgBox "b = " & b & " et c = " & c
c = 45000
d = 3.5: e = -3.4
MsgBox "c, d et e valent " & c & ", " & d & " et " & e

str1 = "Bonjour"
str2 = "Bienvenue dans l'UE47 !"
MsgBox str1 & ". " & str2
```

Q1.2 Quels sont les affichages réalisés par `MsgBox` lors de l'exécution des instructions suivantes ?

```
Dim a As Integer, b As Integer
Dim val As Single
Dim bo1 As Boolean, bo2 As Boolean

a = 2: b = 10
a = 50
MsgBox "a = " & a & " et b = " & b
b = a
a = b
MsgBox "a = " & a & " et b = " & b
val = -2000
MsgBox "val"

bo1 = True
bo2 = False
MsgBox "bo1 vaut " & bo1
bo1 = bo2
MsgBox "bo1 vaut " & bo1
MsgBox "Fin des instructions."
```

Q1.3 Corriger le programme suivant, puis donner les affichages réalisés par MsgBox lors de l'exécution des instructions suivantes.

```
Dim a As Byte, b As Byte
Dim val As Single, bol As Boolean

a = 2
MsgBox "a = " & a
MsgBox b
val = "Hello"
MsgBox val
bol = Vrai
MsgBox bol
c = a
MsgBox "c vaut & c"
b = a, val = - 50
a = 300
MsgBox Fin des instructions
```

Exercice 2 : Opérateurs VBA

Q2.1 Donner les valeurs des variables a, b, c1, c2 et c3 lors de l'exécution de chacune des instructions suivantes.

```
Dim a As Byte, b As Byte
Dim c1 As String, c2 As String, c3 As String

a = 5
b = a + 8
a = a * 2
b = a - 3

c1 = "12"
c2 = "345"
c3 = c1 & c2 & "c1"
c3 = c3 & a
```

Q2.2 Donner les valeurs des variables a, b, c, d et e lors de l'exécution de chacune des instructions suivantes.

```
Dim a As Integer, b As Integer
Dim c As Long, d As Single, e As Single
a = 50: b = 100: c = 10
a = a * b
a = 3 * a + 3
c = c ^ 2
```

```
b = b - 2 * c
c = c - 52
d = a / c
e = a \ c
b = a / c
a = a Mod 10
a = a + 3 * b - 6 * (c Mod b)
```

Q2.3 Que valent les variables b1 à b7 après exécution des instructions suivantes ?

```
Dim p As Boolean, q As Boolean
Dim b1 As Boolean, b2 As Boolean, b3 As Boolean, b4 As Boolean
Dim b5 As Boolean, b6 As Boolean, b7 As Boolean
p = True: q = False

b1 = p And q
b2 = p Or q
b3 = p And (q Or p)
b4 = p And q And p
b5 = p Or (q And q)
b6 = (p Or q) And q
b7 = Not q And (q Or p)
```

Q2.4 Que valent les variables b1 à b4 après l'exécution des instructions suivantes ?

```
Dim b1 As Boolean, b2 As Boolean, b3 As Boolean, b4 As Boolean

b1 = 3 > 5 : b2 = 3 >= 3
b3 = 5 < (10-2)
b4 = b3 And (2 > 0)
```

Exercice 3 : Échange de variables

Q3.1 Soient a et b deux variables de même type (que l'on ne connaît pas) que l'on suppose déjà déclarées, proposer un algorithme échangeant leur contenu avec le moins d'instructions possible.

Q3.2 Soient a, b et c trois variables de type Integer, que l'on suppose déjà déclarées, écrire un algorithme transférant à b la valeur de a, à c la valeur de b et à a la valeur de c avec le moins d'instructions possible.

TD 2 - Entrées/sorties et conditionnelles

Exercice 1 : Entrées/Sorties

Q1.1 Écrire une macro VBA qui demande un nombre à l'utilisateur, puis qui calcule et affiche le carré de ce nombre. Cette procédure doit faire en sorte que la valeur saisie par l'utilisateur soit bien un nombre (et non une chaîne de caractères par exemple).

Q1.2 Écrire une macro VBA qui demande à l'utilisateur de saisir un nombre ou une chaîne de caractères et qui affiche la valeur saisie par l'utilisateur précédée du texte "Vous avez saisi : ".

Q1.3 Écrire une macro VBA qui demande à l'utilisateur de saisir un entier relatif et qui affiche la valeur absolue de ce nombre.

Exercice 2 : Comparaisons

Q2.1 Écrire une macro VBA qui demande à l'utilisateur de saisir deux entiers et qui affiche la valeur du plus grand des deux.

Q2.2 Écrire une macro VBA qui demande à l'utilisateur de saisir deux mots (chaînes de caractères) et affiche le premier des deux mots dans l'ordre alphabétique.

Q2.3 Écrire une macro VBA qui demande à l'utilisateur de saisir trois mots et qui affiche le dernier des trois mots selon l'ordre alphabétique.

Exercice 3 : Impôt

Le but de cet exercice est de calculer le montant de l'impôt de l'utilisateur. Le calcul de l'impôt est progressif. Le revenu de la personne est considéré en trois tranches : de 0 à 1000€, de 1001 à 5000€ et plus de 5000€. La personne va donc payer :

- 10% de la tranche de 0 à 1000€
- 30% de la tranche de 1001 à 5000€
- 50% de la tranche au-dessus de 5000€.

Par exemple l'impôt correspondant au revenu 3000 est $(0,10 \times 1000 + 0,3 \times (3000 - 1000))$.

Écrire une macro VBA qui demande à l'utilisateur de saisir son revenu puis qui affiche le montant de l'impôt à payer.

Exercice 4 : Année bissextile

Une année est bissextile si elle est divisible par 4 et non divisible par 100, ou si elle est divisible par 400. Écrire une macro VBA qui demande à l'utilisateur de saisir une année et qui affiche le message "C'est une année bissextile" si l'année saisie est bissextile, et "Ce n'est pas une année bissextile" sinon.

Note : en VBA, $n \text{ Mod } m$ donne le reste de la division euclidienne de n par m .

TD 3 - Boucles

Exercice 1 : Saisie d'un entier

Écrire en VBA une macro qui demande à l'utilisateur de saisir un entier compris entre 1 et 10 inclus jusqu'à ce que la valeur saisie convienne (i.e. elle est bien comprise entre 1 et 10), puis affiche cette valeur.

Exercice 2 : Suite de Héron d'Alexandrie

La suite de Héron d'Alexandrie (permettant d'approcher la valeur de $\sqrt{2}$) est définie de la façon suivante :

$$u_n = \begin{cases} 1 & \text{si } n = 0 \\ u_{n-1}/2 + 1/u_{n-1} & \text{sinon} \end{cases}$$

Écrire en VBA une macro qui demande à l'utilisateur de saisir un entier positif n , puis affiche la valeur de u_n .

Exercice 3 : Minimum

Écrire en VBA une macro qui demande à l'utilisateur de saisir successivement 10 nombres puis qui affiche le plus petit de ces nombres.

Exercice 4 : Somme de nombres

Q4.1 Écrire en VBA une macro qui affiche la somme de nombres positifs saisis par l'utilisateur. L'utilisateur signale qu'il n'a plus de nombre à saisir en saisissant la valeur -1. Par exemple, si l'utilisateur saisit 1, 10, 5, 8 puis -1, la macro affiche 24.

Q4.2 Écrire en VBA une macro qui affiche la somme de nombres positifs saisis par l'utilisateur jusqu'à la saisie d'un nombre plus petit que le précédent. Le dernier nombre saisi n'est pas compté dans la somme. Par exemple, si les nombres saisis sont 1, 5, 8 puis 2, la macro affiche 14.

TD 4 - Procédures, fonctions et macros

Exercice 1 : Paramètres et portée des variables

Soit le programme VBA suivant :

```
Dim c As Single

Sub testAB ()
    Dim a As Single , b As Single
    a = 1: b = 2: c = 10
    MsgBox "Avant, a = " & a & ", b = " & b & " et c = " & c
    procA
    MsgBox "Après procA, a = " & a & ", b = " & b & " et c = " & c
    procB a, b
    MsgBox "Après procB, a = " & a & ", b = " & b & " et c = " & c
    b = funA(a)
    c = funA(30)
    MsgBox "Après funA, a = " & a & ", b = " & b & " et c = " & c
End Sub

Sub procA ()
    Dim a As Single , b As Single
    a = 5: b = c
End Sub

Sub procB(b As Single , a As Single)
    MsgBox "Dans procB, a = " & a & " et b = " & b
    c = 50
End Sub

Function funA(x As Single) As Single
    funA = 10 * x + 5
End Function
```

Quels sont les différents affichages effectués par **MsgBox** lors de l'exécution de la macro **testAB** ?

Exercice 2 : Aire d'un triangle

La formule suivante permet de calculer l'aire A d'un triangle dont on ne connaît pas la hauteur, à partir de la longueur de ses trois côtés a , b et c :

$$A = \sqrt{p(p-a)(p-b)(p-c)}$$

où le demi-périmètre p vaut :

$$p = \frac{a + b + c}{2}$$

Q2.1 Écrire une fonction VBA qui prend en paramètre trois réels a , b et c et retourne la valeur de l'aire d'un triangle de côtés de longueurs a , b et c . On pourra utiliser la fonction `Sqr` de VBA qui retourne la racine carrée du nombre réel passé en paramètre.

Q2.2 Écrire en VBA une macro qui demande à l'utilisateur de saisir les valeurs des trois côtés d'un triangle a , b et c puis qui affiche « Plus grande » si l'aire du triangle de côtés de longueurs a , b et c est plus grande que l'aire du triangle de côtés de longueurs 10, 15 et 20, et « Plus petite » sinon. La macro vérifie que les données saisies par l'utilisateur sont bien numériques.

Exercice 3 : Nombre parfait

Un nombre parfait est un entier naturel égal à la somme de ses diviseurs stricts (distincts de lui-même). Par exemple $6 = 1 + 2 + 3$ est un nombre parfait.

Q3.1 Écrire en VBA une fonction `parfait` qui prend en paramètre un entier n et retourne vrai si n est parfait, et faux sinon.

Q3.2 Écrire en VBA une procédure `nbParfaits` qui prend un entier n en paramètre et qui affiche successivement tous les nombres parfaits entre 1 et n .

Q3.3 Écrire en VBA une macro qui demande à l'utilisateur de saisir un entier n et qui affiche tous les nombres parfaits entre 1 et n .

Exercice 4 : Palindrome

Q4.1 Écrire en VBA une fonction `palin` qui prend en paramètre un mot et retourne vrai si ce mot est un palindrome (une séquence de lettres qui se lit indifféremment dans les deux sens, comme « laval »), et faux sinon.

On pourra utiliser les fonctions VBA sur les chaînes de caractère suivantes :

- `Len(s)` : retourne la taille de la chaîne de caractères s
- `Mid(s, i, 1)` : retourne le i ème caractère de la chaîne s

Par exemple, `Len("bonjour")` retourne 7 et `Mid("bonjour", 2, 1)` retourne le caractère `o`.

Q4.2 Écrire un macro de test de `palin`. Cette macro commence par demander à l'utilisateur de saisir un mot puis affiche le message « C'est un palindrome. » si le mot saisi est un palindrome, et « Ce n'est pas un palindrome. » sinon. La macro vérifie aussi que l'utilisateur saisit bien une chaîne de caractères.

TD 5 - Cellules Excel

Exercice 1 : Mot de passe

Q1.1 On suppose que la colonne 1 de la feuille Excel contient deux mots de passe : un sur la première ligne et l'autre sur la seconde. Écrire une macro VBA qui demande un mot de passe à l'utilisateur et vérifie que le mot de passe saisi par l'utilisateur corresponde à l'un des deux mots contenus dans la feuille Excel. Tant que ce n'est pas le cas, la procédure demande un nouveau mot de passe à l'utilisateur.

Q1.2 Modifier la macro précédente pour qu'elle demande au maximum trois fois un mot de passe à l'utilisateur. Si au bout de trois essais l'utilisateur n'a pas saisi le mot de passe, la procédure affiche le message « Vous ne connaissez pas le mot de passe, au revoir! »

Exercice 2 : Emploi du temps

Dans cet exercice, on considère que les rendez-vous professionnels de la semaine courante de l'utilisateur (appelons-le M. X) sont représentés dans une feuille Excel comme dans la figure 1.

	1	2	3	4	5	6	7	8	9
1		lundi	mardi	mercredi	jeudi	vendredi	samedi	dimanche	
2	8h-9h			Sacha					
3	9h-10h			Sacha		Mr Z			
4	10h-11h		Mme Y	Sacha					
5	11h-12h								
6	12h-13h								
7	13h-14h								
8	14h-15h		Réunion pôle		Réunion d'équipe				
9	15h-16h				Réunion d'équipe	Mme P			
10	16h-17h	Charles				Mme P			
11	17h-18h	Charles	Mr V						
12	18h-19h								
13	19h-20h								
14									
15									

FIGURE 1 – Emploi du temps de M. X

Les cellules grisées correspondent à des créneaux auxquels M. X ne travaille pas. Un rendez-vous est signalé par un nom figurant dans les cellules correspondantes. Ainsi, "Mme Y" signifie que M. X a rendez-vous avec Mme Y mardi de 10h à 11h, et "Réunion d'équipe" signifie que M. X aura une réunion avec son équipe mercredi de 14h à 16h. Un rendez-vous en italique (comme celui de Mme Y) signifie qu'il peut être déplacé si nécessaire.

Q2.1 Écrire la macro `init` qui initialise l'emploi du temps de la semaine de M. X en effaçant d'éventuelles données déjà présentes, et en grisant (RGB(200,200,200)) les heures du déjeuner et les jours du week-end

comme dans la figure 1.

Q2.2 Écrire la fonction `col` qui prend en paramètre un jour de la semaine sous forme de chaîne de caractères ("lundi", "mardi", etc.) et retourne le numéro de la colonne correspondante dans le planning de M. X. Par exemple `col("mardi")` retourne 3.

Q2.3 Écrire la fonction `libre` qui prend en paramètre un créneau horaire `c` et un jour `j`, et retourne vrai si le créneau `c` est libre au jour `j` et faux sinon. Le créneau est passé en paramètre sous la forme du numéro de ligne lui correspondant dans la feuille Excel, et le jour est passé sous la forme d'une chaîne de caractères ("lundi", "mardi", etc.). Ainsi, `libre(4, "mercredi")` doit retourner faux dans l'exemple de la figure 1 car M. X n'est pas disponible de 10h à 11h mercredi puisqu'il sera en rendez-vous avec Sacha.

Q2.4 Écrire une macro `PremDispo` qui demande à l'utilisateur de saisir un jour de la semaine ("lundi", "mardi", etc.) et affiche le premier créneau libre ce jour-là ou le message "Pas de créneau libre!" s'il n'en existe pas. L'affichage du créneau doit correspondre aux créneaux de la première colonne ("8h-9h", "9h-10h", etc.).

Q2.5 Écrire une fonction `nbRDVModifiables` qui retourne le nombre de rendez-vous modifiables dans une journée passée en paramètre. On rappelle que la classe `Font` a une propriété `Italic` qui est un booléen valant vrai si la police de caractères est en italique. On considère qu'un rendez-vous sur plusieurs créneaux consécutifs du même jour compte pour un seul rendez-vous. L'application de la fonction `nbRDVModifiables` à "vendredi" sur l'exemple de la figure 1 retournerait donc 1 rendez-vous (celui de Mme P).

Q2.6 Écrire une macro `jourPlusModif` qui détermine et affiche le jour de la semaine (entre lundi et vendredi) qui contient le nombre de rendez-vous modifiables le plus élevé.

TD 6 - Traitement de données sous Excel

Exercice 1 : Liste d'employés

Soit une feuille Excel contenant des informations sur les employés d'une société, comme dans la figure 1. Ces employés sont affectés à deux projets A et/ou B. Les informations sur les employés se trouvent à partir de la ligne 3 et de la colonne 1, selon la structure suivante : numéro de l'employé, nom de l'employé, numéro d'employé du supérieur hiérarchique, fonction, coût et nom du (des) projet(s) au(x)quel(s) il appartient. On ne connaît pas le nombre total d'employés dans cette liste.

	1	2	3	4	5	6	7
1							
2							
3	NumEmployé	NomEmployé	NumSupérieur	Fonction	Coût	Projet	
4	7369	Alain	7902	Vendeur	1050	A	
5	7499	Bernard	7521	Manager	1300	A	
6	7521	Charlotte	7902	Manager	1500	A	
7	7566	Didier	7216	Vendeur	1100	A et B	
8	7698	Edouard	7200	Directeur	2850	B	
9	7644	Farid	7902	Vendeur	1100	A	
10	7216	Gaëlle	7698	Manager	1600	A et B	
11	7123	Hassène	7698	Directeur	2500	B	
12	7857	Isabelle	7644	Vendeur	950	B	
13	7902	Jacky	7200	Directeur	2900	A	

FIGURE 1 – Employés

Q1.1 Écrire une fonction VBA `dernLigne` qui retourne le numéro de ligne du dernier employé de la liste des employés.

Q1.2 Écrire en VBA une fonction `calculCoût` qui prend en paramètre le nom d'un projet (chaîne de caractères) et retourne le coût total du projet en question.

Q1.3 On suppose qu'une fonction `nbEmp` a été définie. Elle prend en paramètre le nom d'un projet (de type `String`) et retourne le nombre d'employés qui travaillent sur ce projet.

Écrire en VBA une macro `bilan` qui détermine le coût moyen de chacun des deux projets et met en gras le nom du ou des directeurs du projet de coût moyen le plus élevé. Le coût moyen d'un projet est son coût total divisé par le nombre d'employés impliqués dans le projet.

Q1.4 Écrire en VBA une procédure `copierLigne` qui prend deux entiers `i` et `j` en paramètre et recopie la ligne `i` correspondant aux informations d'un employé à la ligne `j`. Écrire la même procédure de deux autres manières.

Q1.5 Écrire en VBA une procédure `insérerLigne` qui prend en paramètre une plage de données `r` (représentant les informations d'un employé) et un numéro de ligne `i` et insère à la ligne `i` les éléments de la plage `r` tout en décalant les autres lignes vers le bas. On suppose que `r` est une plage de données d'une seule ligne et 6 colonnes. Écrire une macro de test de cette procédure.

Q1.6 Écrire une macro VBA `subord` qui demande à l'utilisateur de saisir le nom d'un employé, puis met en rouge (`RGB(200,0,0)`) la police de caractère du nom de cet employé et grise (`RGB(210,210,210)`) l'intérieur des cellules des noms des subordonnés de l'employé. Si l'employé n'existe pas, la macro redemande à l'utilisateur un nom d'employé jusqu'à la saisie d'un nom valide. On suppose que tous les noms des employés sont différents.

TD7 - Représentation binaire des nombres

Exercice 1 : Représentation binaire

On considère la représentation en binaire des nombres entiers naturels (on n'envisage pas les entiers négatifs).

Q1.1 Quels sont les nombres entiers naturels que l'on peut coder sur 4 bits ? Sur n bits ?

Q1.2 Quel est le nombre représenté par 1011_2 ?

Q1.3 Quelle est la représentation binaire de 43 ?

Exercice 2 : Conversion d'un entier positif en base 2

Q2.1 Écrire en VBA une fonction `nbBits` qui prend en paramètre un nombre décimal n et retourne le nombre de bits nécessaires à la représentation en base 2 du nombre n .

Dans la suite de l'exercice, on suppose qu'un nombre binaire est stocké sur la première ligne de la feuille Excel en mettant un chiffre par cellule. Ainsi le nombre binaire 10010 par exemple est stocké dans la feuille Excel en mettant 1 dans les cellules L1C1 et L1C4 et 0 dans les cellules L1C2, L1C3 et L1C5.

Q2.2 Écrire en VBA une procédure `decToBin` convertissant un décimal en binaire. Cette procédure prend en paramètre le nombre entier à convertir n . Elle détermine dans un premier temps le nombre de bits nb nécessaires à la représentation binaire de n puis écrit le nombre binaire obtenu sur la première ligne des colonnes 1 à nb de la feuille Excel.

Q2.3 Écrire une fonction `BinToDec` qui, étant donnée la taille de la représentation du nombre binaire stocké dans la feuille de calcul, retourne la valeur décimale de ce nombre.

Exercice 3 : Addition binaire sur des entiers positifs

On considère dans cet exercice que la feuille de calcul Excel contient deux nombres binaires N_1 et N_2 de taille n , N_1 est sur la première ligne, de la colonne 1 à la colonne n , et N_2 est sur la deuxième ligne, sur les mêmes colonnes.

Écrire la procédure `additionne` qui, étant donnée la taille n de N_1 et N_2 , additionne les deux nombres binaires N_1 et N_2 et stocke le résultat sur la troisième ligne de la feuille Excel. Si le résultat nécessite plus de n bits, alors `additionne` affiche un message d'erreur.

TD8 - Représentation binaire des nombres relatifs

Exercice 1 : Représentation binaire de nombres relatifs

Q1.1 Quels sont les nombres entiers relatifs que l'on peut coder sur 8 bits avec une représentation en valeur absolue et avec une représentation en complément à 2? Mêmes questions sur n bits.

Q1.2 Donner la représentation sur 8 bits du nombre -41 avec la représentation en valeur absolue et avec la représentation en complément à deux.

Exercice 2 : Conversion d'un entier relatif en complément à deux

Q2.1 Écrire une procédure `compl1` qui calcule le complément à 1 du nombre binaire représenté sur la première ligne de la feuille de calcul Excel sur 8 bits et le place dans la première ligne de la feuille de calcul Excel.

Q2.2 Écrire une procédure `plus1` qui ajoute 1 au nombre binaire représenté sur la première ligne sur 8 bits. La procédure affiche le message "Dépassement de capacité" si l'ajout de 1 à nb provoque un dépassement de capacité.

Q2.3 Écrire une procédure `decToBinR`, qui étant donné un nombre relatif n , détermine la représentation binaire sur 8 bits de n si c'est possible ou affiche le message "Dépassement de capacité" sinon.

Q2.4 Écrire une fonction `binToDecR` qui calcule la valeur décimale d'une représentation binaire de l'entier relatif stocké sur la première ligne de la feuille de calcul.

Exercice 3 : Conversion

Dans cet exercice, on considère un nombre binaire sur n bits auquel on applique l'algorithme suivant : en allant de droite à gauche, on recopie chaque chiffre jusqu'au premier 1, on recopie ce 1, puis on remplace les 1 par des 0 et les 0 par des 1.

Appliquer cet algorithme aux nombres 00011010, 00010100 et 11010111 sur 8 bits. Quels nombres obtient-on? Quelles sont les valeurs de ces nombres en représentation en complément à 2? Que peut-on en déduire?

TD9 - Recherche d'un élément

Exercice 1 : Recherche séquentielle dans une plage de données

Q1.1 Écrire une fonction `RechSeq` qui, étant donné une plage de cellules et un élément `x`, retourne vrai si `x` est présent dans la plage, et faux sinon. On cherche à minimiser le nombre d'opérations effectuées par `RechSeq`.

Q1.2 Écrire la macro de test de `RechSeq` qui recherche le mot "Hello" dans la plage L1C3:L45C10.

Exercice 2 : Recherche dans une liste triée

On suppose dans cet exercice que des entiers sont stockés en ordre croissant dans la feuille Excel de la ligne 1 à la ligne 15000 dans la colonne 1.

Q2.1 Écrire en VBA une fonction `chercheSeq` qui prend en paramètre un entier `x` et retourne vrai si `x` appartient à la plage L1C1:L15000C1 et faux sinon. La fonction `chercheSeq` doit effectuer une recherche séquentielle.

Q2.2 Écrire en VBA une fonction `chercheDicho` qui prend en paramètre un entier `x` et retourne vrai si `x` appartient à la plage L1C1:L15000C1 et faux sinon. La fonction `chercheDicho` doit effectuer une recherche dichotomique.

Q2.3 Quel est le nombre de valeurs testées (comparées à `x`) par `chercheSeq` dans le meilleur-cas et dans le pire-cas ? Même question pour `chercheDicho`.

TD10 - Tri d'éléments

Exercice 1 : Tri par sélection

Q1.1 Soit la collection de nombres $X = [3, 7, 1, 8, 12, 6, 20]$. Appliquer un algorithme de tri par sélection pour trier les éléments de X dans l'ordre décroissant.

Q1.2 Écrire une macro VBA qui applique un algorithme de tri par sélection pour trier en ordre décroissant les nombres situés dans la ligne 1 de la feuille de calcul Excel, de la colonne 1 à la colonne 100.

Q1.3 Quel est le nombre d'opérations réalisées par un algorithme de tri par sélection dans le pire-cas ? Et dans le meilleur des cas ?

Exercice 2 : Tri à bulles

Q2.1 Appliquer un algorithme de tri à bulles pour trier les éléments de l'ensemble X de l'exercice précédent.

Q2.2 Écrire une macro VBA qui applique un algorithme de tri à bulles pour trier en ordre décroissant les nombres situés dans la ligne 1 de la feuille de calcul Excel, de la colonne 1 à la colonne 100.

Q2.3 Améliorer l'algorithme de tri à bulles écrit précédemment en autorisant de stopper son exécution lorsque les éléments sont triés, même si toutes les itérations de boucles n'ont pas encore été exécutées. On sait que les éléments sont triés lorsqu'un parcours des éléments restant à trier n'a entraîné aucun échange.

Q2.4 Quel est le nombre d'opérations réalisées par un algorithme de tri à bulles dans le pire-cas ? Et dans le meilleur des cas ?

TD11 - Récursivité

Exercice 1 : Factorielle

Q1.1 Écrire en VBA une fonction récursive `facto` qui prend un entier n en paramètre et qui retourne $n!$. On suppose que n est positif ou nul.

Q1.2 Écrire une macro VBA qui demande à l'utilisateur de saisir un entier positif (type numérique) n , puis qui affiche la valeur $n!$. Tant que n n'est pas positif ou nul, la macro redemande à l'utilisateur de saisir n .

Exercice 2 : Suite de Héron d'Alexandrie

La suite de Héron d'Alexandrie (permettant d'approcher la valeur de $\sqrt{2}$) est définie de la façon suivante :

$$u_n = \begin{cases} 1 & \text{si } n = 0 \\ u_{n-1}/2 + 1/u_{n-1} & \text{sinon} \end{cases}$$

Q2.1 Écrire en VBA une fonction récursive qui prend en paramètre un entier n ($n \geq 0$) et qui retourne la valeur u_n . On veillera à minimiser le nombre d'opérations.

Exercice 3 : Suite de Fibonacci

La suite de Fibonacci est définie de la façon suivante :

$$F_n = \begin{cases} n & \text{si } n = 0 \text{ ou } n = 1 \\ F_{n-1} + F_{n-2} & \text{sinon} \end{cases}$$

Q3.1 Écrire en VBA une fonction récursive `fiboRec` qui prend en paramètre un entier n ($n \geq 0$) et qui retourne la valeur F_n .

Q3.2 Écrire en VBA une fonction `fiboIt` qui prend en paramètre un entier n ($n \geq 0$) et effectue le même calcul que `fiboRec` mais de manière itérative.

Exercice 4 : Recherche dichotomique

On considère que des entiers sont stockés dans la première colonne de la Feuille Excel en ordre croissant.

Q4.1 Écrire en VBA une fonction récursive `rechDichoRec` qui prend en paramètre un entier n , et deux entiers `li1` et `li2`, et qui effectue une recherche dichotomique de n dans la première colonne de la feuille Excel de la ligne `li1` à la ligne `li2` (incluses). La fonction `rechDichoRec` retourne la ligne, ou une des lignes, à laquelle n est présent si n appartient à la plage examinée, ou -1 sinon.

Q4.2 Écrire en VBA une macro de test qui recherche un nombre saisi par l'utilisateur sur la plage L1C1:L2000C1.