

TD N° 4-5 : LES FONCTIONS PERSONNALISÉES

L'objectif de ces deux séances est de vous initier à l'emploi de fonctions personnalisées écrites en VBA sous Access. Il est possible de télécharger une présentation succincte du langage Visual Basic nommée *vb-reference.pdf* et stockée à l'adresse suivante : <http://eric.univ-lyon2.fr/~jdarmon/docs/>

En annexe de ce document de TD, il est proposé un **bref rappel** sur les structures de sélection en VBA.

I PRÉSENTATION DE LA BASE DE DONNÉES

L'UFR SEG vous demande de construire des requêtes sur des données concernant des étudiants de licence pour simuler plusieurs règlements de scolarité. Elle emploie une base Access nommée *etudes.accdb*.

Télécharger la base *etudes.accdb* stockée à l'adresse : <http://eric.univ-lyon2.fr/~jdarmon/docs/>

La base contient une seule table *Licence* qui possède la structure suivante :

LICENCE (Ch_NumEtudiant, Ch_Nom, Ch_Prenom, Ch_Math, Ch_Stat, Ch_Macro, Ch_Micro)

Remarque : Dans cet exercice, le nom de chaque champ comprend systématiquement un préfixe composé des 2 lettres « Ch » pour que soient bien distingués le nom du champ et le nom du paramètre.

II L'ENCHAINEMENT LINEAIRE : CALCUL DE MOYENNES PONDEREES

Le règlement de la Faculté impose la pondération suivante entre les quatre matières du cursus :

Mathématiques	Statistiques	Macroéconomie	Microéconomie
1,5	2,5	3	3

A Première requête et première fonction : Moyenne des notes en techniques quantitatives

L'objectif de la première requête est d'éditer à l'aide d'une fonction Visual Basic la moyenne pondérée des notes de chaque étudiant en matières quantitatives (mathématiques et statistiques).

1 Créer la requête

Créer une requête avec la table *Licence*, puis insérer dans la grille de création les champs suivants :

Ch_Nom, *Ch_Prenom*, *Ch_Math* et *Ch_Stat*. Les deux derniers champs contiennent les informations nécessaires au calcul. Enregistrer et nommer la requête *Méthodes quantitatives*.

2 Concevoir l'algorithme de la fonction de moyenne pondérée :

Attention ! Il ne s'agit pas d'écrire la fonction VBA dans l'éditeur adéquat d'Access, mais seulement de découvrir la logique du calcul.

a) Pour calculer dans une fonction la moyenne pondérée des matières quantitatives, il est nécessaire d'avoir

- Deux **paramètres** : On appelle **paramètres** les variables de la fonction dont les valeurs proviennent de la base Access. Pour notre calcul de la moyenne des notes en méthode quantitatives, nous avons besoin de deux **variables** dont les valeurs proviennent respectivement des **champs** *Ch_Math* et *Ch_Stat* de la table *Licence*. Les notes sont des données de **type Réel**. Nommons *Par_Stat* les valeurs en provenance du champ *Ch_Stat* et *Par_Math* celles en provenance de *Ch_Math*. Dans toutes les questions posées dans ce TD, les paramètres seront **préfixés** par les 3 lettres « Par ».
- Deux **constantes** : elles mémorisent les coefficients de pondération de chacune des matières. Nous les nommerons *PondMath* et *PondStat*.
- Une **variable locale** : *Somme_Ponderation*, de type Réel, pour mémoriser la somme des deux pondérations employées. C'est une variable locale : elle n'a d'existence qu'au sein de la fonction.
- Un **identificateur** de la fonction : *Fc_Moyenne_Quantitatif*. Il sert d'une part à **nommer** la fonction et d'autre part à **mémoriser** le résultat du calcul effectué dans la fonction ; il est de type **Réel**. Dans les questions de l'exercice, l'identificateur de fonction sera toujours **préfixé** par trois lettres « Fct ».

b) L'algorithme de la fonction (**et non le programme VBA**) peut être celui-ci :

```

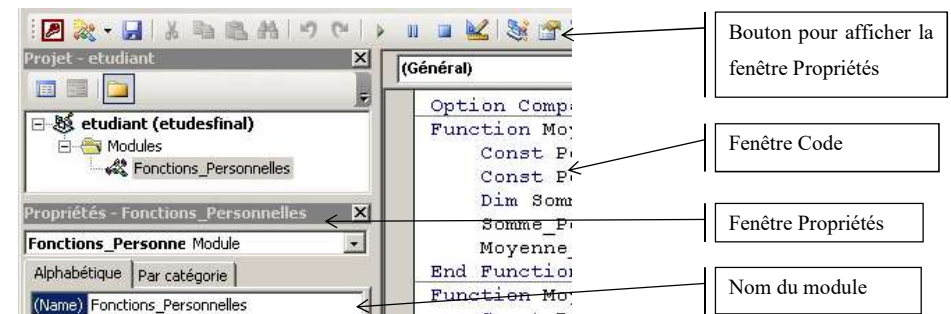
Fonction Fct_Moyenne_Quantitatif (Par_Math, Par_Stat: Réels) : Réels
    Constante
        Pond_Math = 1,5
        Pond_Stat = 2,5
    Variable
        Somme_Ponderation : Réels
    Début
        Somme_Ponderation := Pond_Math + Pond_Stat
        Fct_Moyenne_Quantitatif := (Par_Math * Pond_Math + Par_Stat * Pond_Stat) / Somme_Ponderation
    Fin Fonction
    
```

3 Écrire la fonction en VBA dans un module de la base de données

Pour créer la fonction, vous devez activer, dans la zone « Macros et codes » de l'onglet « Créer » du ruban, le

bouton  :

- Dans la fenêtre *Propriétés* du module, inscrire dans la propriété *Name* le nom que vous voulez donner au module. Nous appellerons le module : « Fonctions_Personnelles » (voir graphique page suivante).



- Dans la fenêtre *Code* écrire au-dessous de la commande **Option Compare Database**, la commande **Option Explicit**. Elle vous permettra une mise au point des programmes plus aisée.
- Écrire ensuite le code de la fonction.

```
Option Compare Database
Option Explicit

Function Fct_Moyenne_Quantitatif(Par_Math As Single, Par_Stat As Single) As Single ' En Tête
' Déclaration des données locales
Const Pond_Math = 1.5 ' déclaration des constantes locales
Const Pond_Stat = 2.5
Dim Somme_Ponderation As Single ' déclaration d'une variable locale de type réel
' Calculs
Somme_Ponderation = Pond_Math + Pond_Stat
Fct_Moyenne_Quantitatif = (Par_Math * Pond_Math + Par_Stat * Pond_Stat) / Somme_Ponderation
End Function
```

- Enregistrer le module à l'aide de la Commande **Fichiers > Enregistrer**.
- Activer la commande **Fichier > Fermer et retourner à Microsoft Access** pour revenir au *Volet de navigation*.

4 Insérer la fonction dans la requête

- Sélectionner la requête *Méthodes Quantitatives* et activer le mode **Création**.
- Dans la première colonne vide (la cinquième), inscrire
 - o Le nom du champ calculé : Moy_Q
 - o Le symbole d'affectation « : »
 - o L'expression de calcul qui inclura l'appel à la fonction VBA que vous venez de construire. L'expression comprend :
 - Le nom de la fonction : *Fct_Moyenne_Quantitatif*, suivi d'une parenthèse ouvrante
 - Les **paramètres réels** de la fonction, c'est-à-dire le nom de chacun des champs contenant des informations nécessaires au calcul. Il faut les écrire entre crochets [Ch_Math] [Ch_Stat] et les séparer par le séparateur de liste « ; ». **Attention !** Il faut utiliser les champs (paramètres réels) **dans le même ordre** que celui des paramètres formels de la fonction VBA écrite dans le module. La valeur contenue dans le champ *Ch_Math* (premier paramètre réel) doit être mémorisée dans le premier paramètre formel *Par_Math*, celle du champ *Ch_Stat* (second paramètre réel) dans le deuxième paramètre formel *Par_Stat*.
 - Une parenthèse fermante.

```
Ch_Nom | Ch_Prenom | Ch_Math | Ch_Stat | Moy_Q: fct_moyenne_quantitatif([ch_math];[ch_stat])
```

- Exécuter la requête et l'enregistrer.
- Ne pas oublier, si l'exécution n'a pas lieu de cliquer sur le bouton **Activer le contenu**

 **Avertissement de sécurité** Du contenu actif a été désactivé. Cliquez pour plus d'informations. Activer le contenu

- En **mode création**, modifier la propriété **Légende** du cinquième champ pour que soit affiché comme étiquette du champ calculé l'expression « Moyenne Quantitatif » et non « Moy_Q ».
- Modifiez la propriété **Format** pour que soit affiché qu'un seul chiffre après la virgule. Le format à employer est celui-ci : 0,0. Exécutez la requête et enregistrez-la.

B Élaborer une fonction simple : Calcul de la moyenne en économie

1 Créer une deuxième fonction VBA dans le module

- Ouvrir le module *Fonctions Personnelles*.
- Au-dessous de la fonction *Fct_Moyenne_Quantitatif*, écrire une nouvelle fonction VBA que vous nommerez *Fct_Moyenne_Eco* et qui permet de calculer la moyenne pondérée en économie.
- La fonction suit la même logique que la précédente. Les pondérations en microéconomie et en macroéconomie sont toutes les deux égales à 3.

2 Insérer la fonction dans une requête

- Par **Copier-Coller**, dupliquer la requête *Méthodes quantitatives* et nommer la copie *Économie*.
- Activer le mode Modification.
- Insérer en sixième et septième colonnes les champs *Ch_Macro* et *Ch_Micro*
- Insérer la nouvelle fonction en huitième colonne. Vous devez obtenir une requête comme ci-dessous :

```
Ch_Nom | Ch_Prenom | Ch_Math | Ch_Stat | MoyQ: Fct_Moyenne_quantitatif([Ch_Math];[Ch_Stat]) | Ch_Macro | Ch_Micro | MoyEco: Fct_Moyenne_Eco([Ch_Macro];[Ch_Micro])
```

- Modifier les propriétés *Légende* et *Format* de la requête pour obtenir le résultat suivant :

Nom	Prenom	Mathématiques	Statistiques	Moyenne Quantitatif	Macro économie	Micro économie	Moyenne Economie
Dupuit	Jules	9,8	8,5	9,0	12,1	10,5	11,3
Egdeworth	Ysidro	15	17	16,3	7,5	8	7,8
Marshall	Alfred	13	7	9,3	10	5,5	7,8
Quesnay	François	7	8,2	7,8	11	8	9,5
Ricardo	David	6	9	7,9	14	12	13,0
Smith	Adam	13	11	11,8	11	13	12,0
Walras	Léon	7	6	6,4	6	7	6,5
Keynes	John-Maynard	2	12	8,3	14	13	13,5
Keynes	John-Neville	4	5	4,6	3	5	4,0

- C Construire une fonction à partir d'autres fonctions : calculer la moyenne générale
- Pour construire la fonction VBA qui permettra de calculer la moyenne générale de chaque étudiant, nous allons créer une fonction personnelle qui appelle, pour ses calculs, les deux fonctions déjà créées.

1 Créer la fonction Moyenne_Generale

- Dans le module *Fonctions Personnelles*, créer une nouvelle fonction *Fct_Moyenne_Gene*.
- Il faut déclarer 1°) quatre **paramètres** que sont les 4 matières que l'étudiant passe lors des examens et qui sont nécessaires aux calculs ; 2°) deux **constantes** que sont les coefficients de pondération des matières quantitatives et de l'économie ; 3°) deux **variables locales**, la somme des pondérations et la somme des notes pondérées
- Pour faire appel aux fonctions déjà créées, il faut construire la nouvelle fonction comme suit :

```

Function Fct_Moyenne_Gene(Par_Math As Single, Par_Stat As Single, Par_Macro As Single, Par_Micro As Single) As Single
' Déclaration des données locales
Const Pond_Quant = 4 ' déclaration des constantes locales
Const Pond_Eco = 6
Dim Somme_Ponderation As Single ' déclaration de variables locales de type réel
Dim Total As Single
' Calculs
Somme_Ponderation = Pond_Quant + Pond_Eco
Total = Fct_Moyenne_Quantitatif(Par_Math, Par_Stat) * Pond_Quant ' Total des points obtenus en matières quantitatives
Total = Total + Fct_Moyenne_Eco(Par_Macro, Par_Micro) * Pond_Eco ' Total des points obtenus
Fct_Moyenne_Gene = Total / Somme_Ponderation
End Function

```

2 Insérer la fonction dans la requête

- Copier-Coller la requête *Économie* et nommer la nouvelle requête *Générale*.
- Insérer la nouvelle fonction en dernière colonne de la requête. Vous devez obtenir le résultat suivant :

Mathématiques	Statistiques	Moyenne Quantitatif	Macro économie	Micro économie	Moyenne Economie	Moyenne générale
9,8	8,5	9,0	12,1	10,5	11,3	10,4
15	17	16,3	7,5	8	7,8	11,1
13	7	9,3	10	5,5	7,8	8,4
7	8,2	7,8	11	8	9,5	8,8
6	9	7,9	14	12	13,0	10,9
13	11	11,8	11	13	12,0	11,9
7	6	6,4	6	7	6,5	6,4
2	12	8,3	14	13	13,5	11,4
4	5	4,6	3	5	4,0	4,3

III LES STRUCTURES ALTERNATIVES

A Premier règlement : Structure alternative simple

1 Objectif

Le premier règlement proposé par la Faculté stipule qu'une personne qui a une moyenne supérieure ou égale à 10 a réussi ; sinon elle a échoué.

Pour pouvoir tester une des valeurs limites du règlement (l'étudiant a une moyenne égale précisément à 10), entrer dans la table *Licence* les informations concernant ce nouvel étudiant :

Ch NumEtudiant	Ch Nom	Ch Prénom	Ch Math	Ch Stat	Ch Macro	Ch Micro
10	Mirabeau	Victor	10	10	10	10

2 Créer la fonction

Dans le module *Fonctions_Personnelles*, construire une nouvelle fonction *Fct_Reglement_1* permettant d'éditer pour chaque étudiant un message de réussite ou d'échec.

```

Function Fct_Reglement_1(Par_Moy_Gen As Single) As String ' Le type String : Texte
If Par_Moy_Gen >= 10 Then ' Par_Moy_Gen mémorise la moyenne générale de l'étudiant
Fct_Reglement_1 = "Réussite"
Else
Fct_Reglement_1 = "Echec"
End If
End Function

```

À partir de la **requête Generale** et de la **fonction Fct_Reglement_1**, créer une nouvelle requête que vous nommerez *Règle 1* afin d'obtenir le résultat suivant :

	Nom	Prénom	Moyenne générale	Resultat
	Dupuit	Jules	10,4	Réussite
	Egdeworth	Ysidro	11,1	Réussite
►	Marshall	Alfred	8,4	Echec
	Quesnay	François	8,8	Echec
	Ricardo	David	10,9	Réussite
	Smith	Adam	11,9	Réussite
	Walras	Léon	6,4	Echec
	Keynes	John-Maynard	11,4	Réussite
	Keynes	John-Neville	4,3	Echec
	Mirabeau	Victor	10,0	Réussite

B Deuxième règlement : Condition complexe

1 Objectifs

Une condition complexe est formée de plusieurs conditions simples liées les unes aux autres par un opérateur logique *And*, *Or*, *Eqv* ou *X_Or*.

Le deuxième règlement stipule que, pour réussir, un étudiant doit avoir non seulement la moyenne générale, mais aussi 8 ou plus en économie **ET** 8 ou plus en matières quantitatives.

Pour tester une des valeurs limites, entrer dans la table *Licence* des informations concernant un nouvel étudiant qui a une moyenne générale élevée, mais pas une moyenne supérieure ou égale à 8 en économie :

Ch NumEtudiant	Ch Nom	Ch Prénom	Ch Math	Ch Stat	Ch Macro	Ch Micro
11	Morgenstern	Oskar	18	19	7	8,5

2 L'emploi de l'opérateur AND

Créer dans le module *Fonctions_Personnelles* une fonction *Reglement_2* qui emploie l'opérateur **AND** pour construire la condition complexe et qui permet d'éditer les résultats des examens en fonction des nouvelles règles.

```

Function Fct_Reglement_2(Par_Moy_Q As Single, Par_Moy_Ec As Single, Par_Moy_Gen As Single) As String
If (Par_Moy_Gen >= 10) And (Par_Moy_Ec >= 8) And (Par_Moy_Q >= 8) Then
Fct_Reglement_2 = "Réussite"
Else
Fct_Reglement_2 = "Echec"
End If
End Function

```

Insérer cette fonction dans une nouvelle requête que vous nommerez *Règle 2*. L'exécution de la nouvelle requête doit donner le résultat suivant :

Nom	Prénom	Moyenne Quantitatif	Moyenne Economie	Moyenne générale	Resultat
Dupuit	Jules	9,0	11,3	10,4	Réussite
Egdeworth	Ysidro	16,3	7,8	11,1	Echec
Marshall	Alfred	9,3	7,8	8,4	Echec
Quesnay	François	7,8	9,5	8,8	Echec
Ricardo	David	7,9	13,0	10,9	Echec
Smith	Adam	11,8	12,0	11,9	Réussite
Walras	Léon	6,4	6,5	6,4	Echec
Keynes	John-Maynard	8,3	13,5	11,4	Réussite
Keynes	John-Neville	4,6	4,0	4,3	Echec
Mirabeau	Victor	10,0	10,0	10,0	Réussite
Morgenstern	Oskar	18,6	7,75	12,1	Echec

3 L'emploi de l'opérateur OR

Créer une fonction *Fct_Reglement_2_Bis* qui n'emploie que l'opérateur **OR** pour construire la condition complexe et qui permet de tester le **même règlement** de scolarité que celui qui a été traité précédemment. Une nouvelle requête, que vous nommerez *Règle 2_Bis* et qui est fondée sur la fonction *Fct_Reglement_2_Bis*, doit donner le même résultat que précédemment.

Comment qualifier la différence entre les deux fonctions *Fct_Reglement_2* et *Fct_Reglement_2_Bis* ?

C Troisième règlement : L'imbrication

1 Objectifs

Il est possible d'écrire une structure alternative au sein d'une autre structure alternative.

Le troisième règlement stipule qu'un étudiant a réussi s'il a une moyenne supérieure ou égale à 10. Si l'étudiant n'a pas réussi, il peut cependant acquérir le module d'économie s'il obtient une moyenne supérieure ou égale à 10 dans ce module. Si l'étudiant n'obtient ni une moyenne générale ni une moyenne au module d'économie supérieure ou égale à 10, il a échoué.

Pour tester certaines valeurs, entrer dans la table *Licence* des informations sur de nouveaux étudiants. Aucun de ces étudiants n'a réussi son année ; en revanche, le deuxième a obtenu son module d'économie.

12	Mill	James	10	11	7	6
13	Turgot	Anne	4	2	11	12
14	Scott	Dun	10	11	5	3

2 Créer une fonction

Créer une fonction *Reglement_3* qui permet la gestion d'un tel règlement.

```
Function Fct_Reglement_3(Par_Moy_Ec As Single, Par_Moy_Gen As Single) As String
    If (Par_Moy_Gen < 10) Then ' L'étudiant n'a pas la moyenne générale
        If (Par_Moy_Ec >= 10) Then ' L'étudiant a la moyenne au module d'économie
            Fct_Reglement_3 = "Module d'économie acquis"
        Else ' L'étudiant n'a pas la moyenne au module d'économie
            Fct_Reglement_3 = "Échec"
        End If
    Else ' L'étudiant a la moyenne générale à l'examen
        Fct_Reglement_3 = "Réussite"
    End If
End Function
```

3 Insérer la fonction dans une requête

Insérer cette fonction dans une requête que vous nommerez *Règle 3*. La requête éditera le résultat suivant :

	Nom	Prénom	Moyenne Quantitatif	Moyenne Economie	Moyenne générale	Resultat
▶	Dupuit	Jules	9,0	11,3	10,4	Réussite
	Egdeworth	Ysidro	16,3	7,8	11,1	Réussite
	Marshall	Alfred	9,3	7,8	8,4	Echec
	Quesnay	François	7,8	9,5	8,8	Echec
	Ricardo	David	7,9	13,0	10,9	Réussite
	Smith	Adam	11,8	12,0	11,9	Réussite
	Walras	Léon	6,4	6,5	6,4	Echec
	Keynes	John-Maynard	8,3	13,5	11,4	Réussite
	Keynes	John-Neville	4,6	4,0	4,3	Echec
	Mirabeau	Victor	10,0	10,0	10,0	Réussite
	Morgenstern	Oskar	18,6	7,8	12,1	Réussite
	Mill	James	10,6	6,5	8,1	Echec
	Turgot	Anne	2,8	11,5	8,0	Module économie acquis
	Scott	Dun	10,6	4,0	6,7	Echec

4 Règlement 3 bis

Le troisième règlement est modifié. Sa nouvelle version stipule qu'un étudiant a réussi s'il a une moyenne générale supérieure ou égale à 10. Sinon, il acquiert le module d'économie s'il a la moyenne supérieure ou égale à 10 dans ce module ; il acquiert le module de méthodes quantitatives s'il a une moyenne supérieure ou égale à 10 dans ce module. Si la moyenne générale, la moyenne au module d'économie et la moyenne au module de méthodes quantitatives sont toutes les trois inférieures à 10, il a échoué.

Créer une nouvelle fonction que vous nommerez *Fct_Reglement_3_Bis* capable de gérer cette nouvelle règle et insérer la dans une requête *Règle 3 bis*. Vous devez trouver le résultat suivant :

	Nom	Prénom	Moyenne Quantitatif	Moyenne Economie	Moyenne générale	Resultat
	Dupuit	Jules	9,0	11,3	10,4	Réussite
	Egdeworth	Ysidro	16,3	7,8	11,1	Réussite
	Marshall	Alfred	9,3	7,8	8,4	Echec
	Quesnay	François	7,8	9,5	8,8	Echec
	Ricardo	David	7,9	13,0	10,9	Réussite
	Smith	Adam	11,8	12,0	11,9	Réussite
	Walras	Léon	6,4	6,5	6,4	Echec
	Keynes	John-Maynard	8,3	13,5	11,4	Réussite
	Keynes	John-Neville	4,6	4,0	4,3	Echec
	Mirabeau	Victor	10,0	10,0	10,0	Réussite
	Morgenstern	Oskar	18,6	7,8	12,1	Réussite
	Mill	James	10,6	6,5	8,1	Module quantitatif acquis
▶	Turgot	Anne	2,8	11,5	8,0	Module économie acquis
	Scott	Dun	10,6	4,0	6,7	Module quantitatif acquis

IV LE CHOIX MULTIPLE

Le quatrième règlement de scolarité admet le règlement 3 bis. Il stipule en plus que le jury accorde des mentions selon les règles suivantes fondées sur la moyenne générale (MG) :

10 >= MG < 12	12 >= MG < 14	14 >= MG < 16	MG > 18
Mention Passable	Mention Assez bien	Mention Bien	Mention Très Bien

Pour tester certaines valeurs, entrer dans la table *Licence* les informations suivantes :

Ch NumEtudiant	Ch Nom	Ch Prénom	Ch Math	Ch Stat	Ch Macro	Ch Micro
15	Mill	John Stuart	18	16	17	15
16	Robinson	Joan	13	15	15	13
17	Sraffa	Piero	12	12,5	11	14

Pour construire la fonction « Reglement_4 », il faut employer la structure Select Case pour déterminer les différentes mentions.

```
Function Fct_Reglement_4(Par_Moy_Q As Single, Par_Moy_Ec As Single, Par_Moy_Gen As Single) As String
    Dim Tronc_MoyGen As Byte
    Tronc_MoyGen = Fix(Par_Moy_Gen) ' Fix renvoie la partie entière d'un nombre
    Select Case Tronc_MoyGen ' Structure de choix multiples
        Case 0 To 9 ' Test sur les valeurs inférieures à la moyenne
            If Par_Moy_Q >= 10 Then ' Test sur le module des matières quantitatives
                Fct_Reglement_4 = "Module quantitatif acquis"
            Else
                If Par_Moy_Ec >= 10 Then ' Test sur le module des matières d'économie
                    Fct_Reglement_4 = "Module économie acquis"
                Else
                    Fct_Reglement_4 = "Echec"
                End If
            End If
        End If
        Case 10 To 11 ' Test sur les mentions
            Fct_Reglement_4 = "Mention Passable"
        Case 12 To 13
            Fct_Reglement_4 = "Mention Assez bien"
        Case 14 To 15
            Fct_Reglement_4 = "Mention Bien"
        Case Else
            Fct_Reglement_4 = "Mention Très bien"
    End Select
End Function
```

Vous devez obtenir le résultat suivant :

Nom	Prénom	Moyenne Quantitatif	Moyenne Economie	Moyenne générale	Resultat
Dupuit	Jules	9,0	11,3	10,4	Mention Passable
Egdeworth	Ysidro	16,3	7,8	11,1	Mention Passable
Marshall	Alfred	9,3	7,8	8,4	Echec
Quesnay	François	7,8	9,5	8,8	Echec
Ricardo	David	7,9	13,0	10,9	Mention Passable
Smith	Adam	11,8	12,0	11,9	Mention Passable
Walras	Léon	6,4	6,5	6,4	Echec
Keynes	John-Maynard	8,3	13,5	11,4	Mention Passable
Keynes	John-Neville	4,6	4,0	4,3	Echec
Mirabeau	Victor	10,0	10,0	10,0	Mention Passable
Morgenstern	Oskar	18,6	7,8	12,1	Mention Assez bien
Mill	James	10,6	6,5	8,1	Module quantitatif acquis
Turgot	Anne	2,8	11,5	8,0	Module économie acquis
Scott	Dun	10,6	4,0	6,7	Module quantitatif acquis
Mill	John Stuart	16,8	16,0	16,3	Mention Très bien
Robinson	Joan	14,3	14,0	14,1	Mention Bien
Sraffa	Piero	12,3	12,5	12,4	Mention Assez bien

V STRUCTURES REPETITIVES CONDITIONNELLES

Tant qu’une condition est vraie, une structure répétitive conditionnelle répète une action. Elle s’écrit en suivant la syntaxe suivante :

```
While Condition
    Action
Wend
```

Le jury décide de remonter les notes de microéconomie de tous les étudiants qui ont une moyenne générale comprise entre 9,7 et 10, de manière à ce que cette moyenne soit juste égale à 10.

Entrer dans la table *Licence* les informations concernant ce nouvel étudiant :

Ch_NumEtudiant	Ch_Nom	Ch_Prénom	Ch_Math	Ch_Stat	Ch_Macro	Ch_Micro
18	Bentham	Samuel	9	11	11	8

A) Créer une requête à partir de la table LICENCE et sélectionner tous les champs de la table.

En colonne 8, calculer la moyenne générale de tous les étudiants à l’aide de la fonction Fct_Moyenne_Gene créée en début de séance.

Dans la ligne *Critères*, écrire un critère qui permet de ne retenir que les étudiants dont la moyenne générale est comprise entre 9,7 et 10.

Enregistrer la requête sous le nom *Réévaluation* et tester la requête. Vérifier que seul Bentham doit voir sa note de microéconomie remontée, car il a 9,8 de moyenne générale.

Créer une fonction que vous nommerez *NouvelleNote*. Elle a pour paramètres les quatre notes : mathématiques, statistiques, macroéconomie et microéconomie.

Avec ces notes, la fonction calcule la moyenne générale à l’aide de la fonction Fct_Moyenne_Generale. Cette moyenne est testée. Tant qu’elle n’est pas égale à 10, il faut ajouter 0.1 à la note de microéconomie, puis recalculer la moyenne générale et la tester à nouveau.

```
Function Fct_Nouvelle_Note(Par_Math As Single, Par_Stat As Single, Par_Macro As Single, Par_Micro As Single) As Single
    Dim MoyenneGenerale As Single
    MoyenneGenerale = Fct_Moyenne_Gene(Par_Math, Par_Stat, Par_Macro, Par_Micro)
    While MoyenneGenerale < 10
        Par_Micro = Par_Micro + 0.1
        MoyenneGenerale = Fct_Moyenne_Gene(Par_Math, Par_Stat, Par_Macro, Par_Micro)
    Wend
    Fct_Nouvelle_Note = Par_Micro
End Function
```

Insérer la fonction dans la neuvième colonne de la requête *Réévaluation*. Modifier la légende, puis le format pour ne faire apparaître qu’une décimale.

Insérer de nouveau la fonction de moyenne générale pour bien vérifier que la nouvelle note permet à l’étudiant d’obtenir la moyenne générale.

Número étudiant	Nom	Prénom	Mathématiques	Statistiques	Macro économie	Micro économie	Moyenne générale départ	Nouvelle Note de Micro économie	Nouvelle moyenne
18	Bentham	Samuel	9	11	11	8	9,8	8,7	10,0

B) Ajouter dans la table Licence :

Nom	Prénom	Math	Stat	Macro	Micro
Tobin	James	9,2	9,7	10	10,3

C) Créer une requête à partir de la table LICENCE et sélectionner tous les champs de la table.

Comme précédemment, calculer la moyenne générale de tous les étudiants, limiter aux étudiants dont la moyenne générale est comprise entre 9,8 et 10. [

Créer une fonction qui ajuste la note de statistique pour atteindre 10 de moyenne générale. Vous la nommerez *NouvelleNoteStat* Utilisez « jusqu’à » (Do ... Loop Until) Voir Memento pages suivantes

Enregistrer la requête sous le nom *RéévaluationStat* et tester la. Vérifier que seul Tobin doit voir sa note de statistique remontée, car il a 9,9 de moyenne générale. (avant réévaluation de sa note de stat)

VI MEMENTO SUR LES STRUCTURES DE SELECTION

A Rappels sur les structures alternatives

1 Rappels d'algorithmique

Une structure de contrôle alternative permet, en fonction d'une condition, soit de faire ou non une action, soit de faire une action ou une autre.

Le choix entre agir ou ne pas agir	Le choix entre deux actions
Si Condition <u>Alors</u> Action <u>Fin Si</u>	Si Condition <u>Alors</u> Action 1 <u>Sinon</u> Action 2 <u>Fin Si</u>

Une condition simple est formée de deux variables liées par un opérateur de comparaison ou d'une variable et d'une valeur liées par un opérateur de comparaison (=, >, <, <>, >=, <=).

Une condition complexe est formée de deux ou plusieurs conditions simples liées par un opérateur logique (Et, Ou, Eqv, Ou_X).

Ou_inclusif	Condition 1			Et	Condition 1	
Condition 2 Vrai Faux	Vrai	Faux		Condition 2 Vrai Faux	Vrai	Faux
	Vrai	Vrai			Vrai	Faux
	Vrai	Faux			Faux	Faux
Ou_exclusif	Condition 1			Condition 1		
Condition 2 Vrai Faux	Vrai	Faux		Condition 2 Vrai Faux	Vrai	Faux
	Vrai	Faux			Vrai	Faux
	Vrai	Vrai			Faux	Faux

2 Les structures alternatives en Visual Basic

Une fonction contenant une condition complexe construite avec l'opérateur AND	Une fonction contenant une condition complexe construite avec l'opérateur OR
<u>Function</u> Test1 (Note <u>As Single</u>) <u>As String</u> <u>If</u> (Note <= 20) <u>And</u> (Note >= 0) <u>Then</u> Test1 = "Note acceptée" <u>Else</u> Test1 = "Note impossible" <u>End If</u> <u>End Function</u>	<u>Function</u> Test2 (Note <u>As Single</u>) <u>As String</u> <u>If</u> (Note > 20) <u>Or</u> (Note < 0) <u>Then</u> Test2 = "Note impossible" <u>Else</u> Test2 = "Note acceptée" <u>End If</u> <u>End Function</u>

B Rappel sur la structure de choix multiples

1 Rappels d'algorithmique

La structure de contrôle de choix permet en fonction d'une valeur d'exécuter une parmi plusieurs actions possibles. Si une valeur entrée par l'utilisateur ne correspond à aucun des choix proposés, c'est l'action proposée à la suite du mot clé Sinon qui est exécutée.

```
Selon Variable
Début
    Valeur 1 : action 1
    Valeur 2 : action 2
    .....
    Valeur n : action n
    Sinon : action n+1
Fin
```

2 La structure de choix multiple en Visual Basic

La structure **Select Case** de Visual Basic Application admet le test non seulement sur une valeur, mais aussi sur un ensemble de valeurs ou une inégalité :

```
Function Mention (Note As Single) As String
    Dim Selecteur As Byte ' Le choix s'opère à partir d'une valeur entière
    Selecteur = Fix(Note) ' Prendre la partie entière de la note
    Select Case Selecteur
        Case 0 ' Test sur une valeur
            Mention = "Copies blanches"
        Case 1 To 9 ' Test sur un ensemble de valeurs
            Mention = "Échec"
        Case 10 To 11
            Mention = "Mention passable"
        Case 12 To 13
            Mention = "Mention Assez Bien"
        Case 14 To 16
            Mention = "Mention Bien"
        Case Else ' Si la valeur n'est pas parmi celles testées
            Mention = "Félicitations du jury"
    End Select
End Function
```

C La structure répétitive conditionnelle

1 Rappel d'algorithmique

Une structure de contrôle répétitive conditionnelle permet de répéter une action un nombre de fois indéterminé à l'avance,

- soit jusqu'à ce qu'une condition soit satisfaite,
- soit tant qu'une condition est satisfaite.

Structure Répéter ... Jusqu'à Condition vraie	Structure Tant que Condition vraie Faire
<u>Répéter</u> Action <u>Jusqu'à</u> Condition	<u>Tant que</u> Condition <u>faire</u> Action <u>Fin Tant que</u>

2 Structure en Visual Basic

Structure Répéter ... Jusqu'à Condition vraie	Structure Tant que Condition vraie Faire
<u>Do</u> Action <u>Loop Until</u> Condition	<u>While</u> Condition Action <u>Wend</u>

Exemple :

A = 0 <u>Do</u> A = A + 1 <u>Loop Until</u> A > 10	A = 0 <u>While</u> A <= 10 A = A + 1 <u>Wend</u>
---	---

Remarque : Les deux structures permettent d'aboutir au même résultat (A mémorise la valeur 11), mais la condition écrite pour la première structure est l'exact inverse de celle qui a été écrite pour la seconde.

Option Explicit

Function Fct_Moyenne_Eco(Par_Macro As Single, Par_Micro As Single) As Single

' Déclaration des constantes et variables

Const PondMacro = 3

Const PondMicro = 3

Dim SommePonderation As Single

' Calculs

SommePonderation = PondMacro + PondMicro

Fct_Moyenne_Eco = (Par_Macro * PondMacro + Par_Micro * PondMicro)
/ SommePonderation

End Function

**Function Fct_Reglement_2_Bis(Par_MoyGen As Single, Par_MoyQuant As Single,
Par_MoyEco As Single) As String**

If Par_MoyGen < 10 Or Par_MoyQuant < 8 Or Par_MoyEco < 8 Then

Fct_Reglement_2_Bis = "Echec"

Else

Fct_Reglement_2_Bis = "Réussite"

End If

End Function

**Function Fct_Reglement_3_Bis(Par_MoyGen As Single, Par_MoyQuant As Single,
Par_MoyEco As Single) As String**

If Par_MoyGen >= 10 Then

Fct_Reglement_3_Bis = "Réussite"

ElseIf Par_MoyEco >= 10 Then

Fct_Reglement_3_Bis = "Module économie acquis"

ElseIf Par_MoyQuant >= 10 Then

Fct_Reglement_3_Bis = "Module quantitatif acquis"

Else

Fct_Reglement_3_Bis = "Echec"

End If

End Function

```

Function Fct_NouvelleNoteStat(Par_math As Single, Par_stat As Single,
                               Par_macro As Single, Par_micro As Single) As Single

    ' Déclaration de variable
    Dim MoyenneGenerale As Single

    ' Modification de la note de stat

    ' Moyenne générale de départ
    MoyenneGenerale = Fct_Moyenne_Gen(Par_math, Par_stat, Par_macro, Par_micro)
    Do
        Par_stat = Par_stat + 0.1
        ' Nouvelle moyenne générale
        MoyenneGenerale = Fct_Moyenne_Gen(Par_math, Par_stat, Par_macro,
                                           Par_micro)
    Loop Until MoyenneGenerale >= 10

    ' Résultat
    Fct_NouvelleNoteStat = Par_stat

End Function

```