 **République Algérienne Démocratique et Populaire**

**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**

**Université Akli Mohand Oulhadj de Bouira**

**Faculté des Sciences et des Sciences Appliquées**

**Département d'Informatique**

**Compte Rendu de MTI**

***Spécialité*** *: GSI*

|  |
| --- |
| **TP n°1** |

**Réalisé par**

- Mehennaoui Abdelghani

-boukazoula boulame

2019/2020

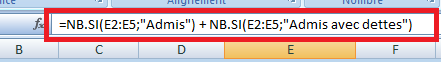
# Méthodes de travail

## partie 01 :

L’objectif de cette partie est d’apprendre à utiliser la programmation fonctionnelle

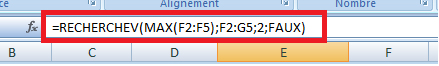
Pour réaliser notre travail, nous avons utilisés l’Excel.

1. Pour calculer le nombre des étudiants admis, en utilisant la fonction **NB.SI**

****

**Figure1.1 :la syntaxe de la fonction NB.SI(NB.SI (plage ; critère))**

1. Pour déterminer le nom du majeur de promotion utilisant la fonction **RechercheV**



**Figure1.2 :la syntaxe de la fonction RechercheV**

3 pour calcule la moyenne de deux nombre en utilisant la fonction MOYENNE



**Figure1.3 :la syntaxe de la fonction MOYENNE**

## partie 02 :

L’objectif de cette partie, c’est d’apprendre le langage Python, nous avons apprendre :

* Comment installer Python sous Windows.
* Pour affiché un variable ou bien une chaine de caractère, en utilise la fonction **print ()**.
* pas de déclaration explicite.
* Pour afficher le type de variable en utilise la fonction **type (ma\_variable)**.
* La syntaxe utilisé pour déclarer une liste est : **ma\_liste = [ele1, ele2, eleN],** pour ajouter un élément a cette liste en utilisant la fonction **append (elemntX)** .
* On peut parcourir une liste des éléments grâce à la syntaxe :

**for élément in ma\_liste :**

## travail a domicile :

# Introduction

il écrire un programme Python sert à lire les données des étudiants de 1ere année master à partir d’un fichier texte et on affiche le résultat sous forme d’une page web.

Pour chaque un étudiant en 1ere année master est admis en 2eme année master, il faux que les condition suivante selon réalisé :

1-pour chaque semestres le moyenne supérieure à 10 pour chaque semestre. Et la compensation n’est existé pas

2-Pour le passage par dettes, l’étudiant doit avoir 45 crédits. Et l’unité fondamentale acquise

afficher les résultats de l’année et préciser également si l’étudiant est admis, admis avec dettes ou ajournée, mentionner la meilleure et la mauvaise moyenne et organiser des statistiques qui concerne le nombre et le pourcentage de ce résultats.

2.Conception de la solution :

* 1. Lire les données a partir d’un fichier texte :

f=open("pro1.txt","r")

* 1. L’extraction des données depuis le fichier (pro1.txt) :

Le fichier (pro1.txt) est écrit sous forme d’une table (colonnes)

2.1 Les solution possible :

* Chaque ligne sera considérer comme (objet) et ajouter à une liste du même type.Autre solutions :[n°1]
* Chaque colonne sera considérer comme une table des chaines de caractères, alors il y aura 9 tables (Matricule [], Nom [], Prénom [] … etc.) [n°2]

Les Avantages de chaque solutions obtenu  n°1:

a/solutions obtenu  n°1

* + La bonne structure des données.
  + L’indépendance des objets.
  + La facilite de la manipulation.

b/solutions obtenu  n°2:

* Minimiser le nombre des opérations.
* L’accès rapide aux données.
* Lire une partie des données sans parcourir tous les données.
* La facilite de la manipulation.

**list=[]**

**f=f.read().replace('\t','\n')**

**list=f.split('\n')**

**matricule=list[9::9]**

**nom=list[10::9]**

**prenom=list[11::9]**

**moy\_s1=list[12::9]**

**moy\_s2=list[13::9]**

**moy\_annuel=list[14::9]**

**credit\_s1=list[15::9]**

**credits\_s2=list[16::9]**

**unite\_acuise=list[17::9]**

**Figure1.4 :le code source de lire les donne sans in 9 tabule**

c-Le calcule et l’affichage des résultats de l’année (admis, admis avec dettes, ajournée)

j=0

for i in moy\_s1:

if float(moy\_s1[j]) >= 10 and float(moy\_s2[j]) >= 10:

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Admis </td></tr></table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ad=ele\_ad+1

elif ((int(credit\_s1[j]) == 30 and int(credits\_s2[j]) >= 15) or (int(credit\_s1[j]) == 15 and int(credits\_s2[j]) >= 30)) and unite\_acuise[j]=="Oui":

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Admis avec detts </td></tr> </table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ad\_det=ele\_ad\_det+1

else:

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Ajournee </td></tr></table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ajou=ele\_ajou+1

**j= j +1**

**Figure1.5 :le code source pure afficher les étudiant que admis/admis avec dets/ ajournée**

d-Le calcule et l’affichage de la meilleure/mauvaise moyenne :

**meilleure\_not= float(moy\_annuel[0])**

**mauvaise\_not= float(moy\_annuel[0])**

**j=0**

**#indise le mauvaise et meure not**

**indi\_meilleure=0**

**indi\_mauvaise=0**

**for i in moy\_annuel:**

**#rechercher le eleve que meure not**

**if meilleure\_not< float(moy\_annuel[j]):**

**meilleure\_not=float(moy\_annuel[j])**

**indi\_meilleure=j**

**# recherche le eleve de mauvaise not**

**if mauvaise\_not > float(moy\_annuel[j]):**

**mauvaise\_not=float(moy\_annuel[j])**

**indi\_mauvaise=j**

**j=j+1**

**Figure1.6 :le code source pour ajouter le meure et mauvaise note**

e-Le calcule et l’affichage des statistique.

**x=float((ele\_ad\*100)/j)**

**y=float((ele\_ad\_det\*100)/j)**

**z=float((ele\_ajou\*100)/j)**

**Figure1.7 :le code source pour les statistique**

# Annexe

***1-le code de source***

#le code source générale de set travaille

f=open("pro1.txt","r")

list=[]

f=f.read().replace('\t','\n')

list=f.split('\n')

matricule=list[9::9]

nom=list[10::9]

prenom=list[11::9]

moy\_s1=list[12::9]

moy\_s2=list[13::9]

moy\_annuel=list[14::9]

credit\_s1=list[15::9]

credits\_s2=list[16::9]

unite\_acuise=list[17::9]

ele\_ad=0

ele\_ad\_det=0

ele\_ajou=0

o=0

m=""

credit\_moyen=[]

#veri les not

m+= ("<table border='2'><tr><td width=80px>Matri</td><td width=120px>Nom</td><td width=120px>prennome</td><td width=70px> moyen\_s1</td><td width=70px>moyen\_s2</td><td width=120px>Etat</td></tr></table>")

j=0

for i in moy\_s1:

if float(moy\_s1[j]) >= 10 and float(moy\_s2[j]) >= 10:

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Admis </td></tr></table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ad=ele\_ad+1

elif ((int(credit\_s1[j]) == 30 and int(credits\_s2[j]) >= 15) or (int(credit\_s1[j]) == 15 and int(credits\_s2[j]) >= 30)) and unite\_acuise[j]=="Oui":

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Admis avec detts </td></tr> </table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ad\_det=ele\_ad\_det+1

else:

m += ("<table border='2'><tr><td width=80px> %s</td><td width=120px> %s</td><td width=120px> %s </td><td width=70px> %s</td><td width=70px> %s</td><td width=120px> Ajournee </td></tr></table>" % (matricule[j],nom[j], prenom[j],moy\_s1[j],moy\_s2[j]))

ele\_ajou=ele\_ajou+1

j= j +1

meilleure\_not= float(moy\_annuel[0])

mauvaise\_not= float(moy\_annuel[0])

j=0

#indise le mauvaise et meure not

indi\_meilleure=0

indi\_mauvaise=0

for i in moy\_annuel:

#rechercher le eleve que meure not

if meilleure\_not< float(moy\_annuel[j]):

meilleure\_not=float(moy\_annuel[j])

indi\_meilleure=j

# recherche le eleve de mauvaise not

if mauvaise\_not > float(moy\_annuel[j]):

mauvaise\_not=float(moy\_annuel[j])

indi\_mauvaise=j

j=j+1

me=nom[indi\_meilleure] + " " + prenom[indi\_meilleure]

m += ("<br><div id='donne'>la meilleur moyenne est : %s de l'etudiant : %s </div>" % (moy\_annuel[indi\_meilleure], me))

mi=nom[indi\_mauvaise]+" " + prenom[indi\_mauvaise]

m += ("<br><div id='donne'>la mauvaise moyenne est : %s de l'etudiant : %s </div>" % (moy\_annuel[indi\_mauvaise], mi))

#print(j,ele\_ad,ele\_ad\_det,ele\_ajou)

x=float((ele\_ad\*100)/j)

y=float((ele\_ad\_det\*100)/j)

z=float((ele\_ajou\*100)/j)

m += ("<br><div id='donne'>le nombre des eleves qui admis= %s etidiant :les percentage : %.2f %%</div>" %(ele\_ad,x))

m += ("<br><div id='donne'>le nombre des eleves qui dmis avec dettes= %s etidiant :le percentage : %.2f %%</div>" %(ele\_ad\_det,y))

m += ("<br><div id='donne'>le nombre des eleves qui ajournée=%s etidiant:le percentage : %.2f %%</div>" %(ele\_ajou,z))

fi = open("abdelghani.html","w")

message = ("""<html>

<head>

<style type="text/css">

body{

background:#808080;

}

#donne{

color:blue;

font-size:20px;

}

</style>

</head>

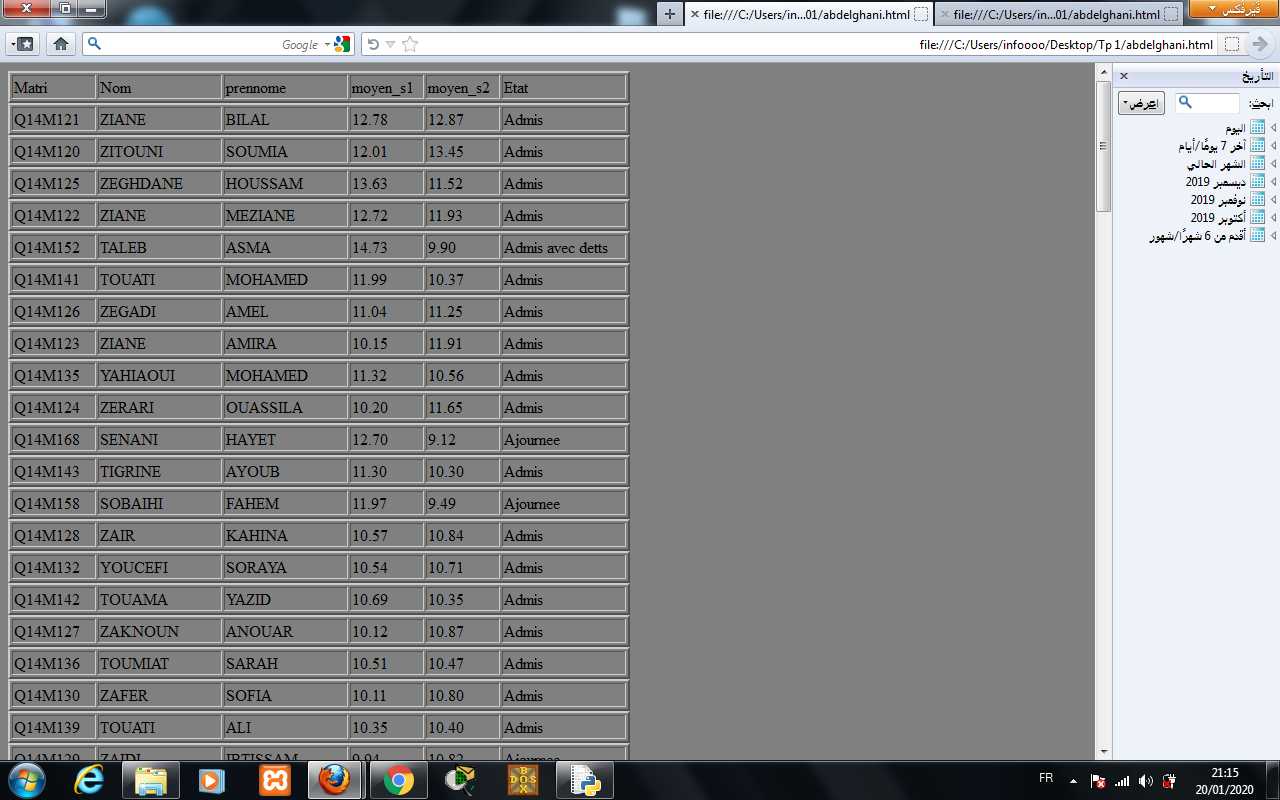
<body>%s</body>

</html>""" % m)

fi.write(message)

fi.close

***2-le résultat final :***

******

**Figure1.7 :le résultat final affiche sous forme HTML**

******

**Figure1.8 :le résultat final affiche sous forme HTML**