

# TD 10 Corrigé : Python

ESSADDOUKI Mostafa (essaddouki@gmail.com), 13 mai 2018

## Exercice 1

Soit un fichier typé intitulé `concours.txt` qui comporte les enregistrements relatifs aux candidats d'un concours. Chaque enregistrement est composé de : **NCIN**, **NOM**, **PRENOM**, **AGE**, **DECISION** : (type contenant les identificateurs suivants : admis, refusé, ajourné), et séparé par point virgule (;).

### Travail demandé :

- Définir la fonction `saisir()` qui permet de remplir les données relatives aux candidats dans le fichier `concours.txt`

```

1 def saisir():
2     new="0" # 0 -> oui ; N -> non
3     fichier=open("concours.txt","a")
4     decision={"a":"admis(e)","r":"refuse(e)","aj":"ajourne(e)"}
5     while new=="0":
6         cin=input("Saisir le Numero CIN:_")
7         nom=input("Saisir le Nom:_")
8         prenom=input("Saisir le prenom:_")
9         age=input("saisir l'age_")
10        dec=input("saisir la decision_a_(admis(e))_;_r_(refuse(e))_";
11        .....aj_(ajourne(e))_:_")
12        ligne=cin+";"+nom+";"+prenom+";"+age+";"+decision[dec]+"\\n"
13        fichier.write(ligne)
14        new=input("Saisir un nouveau candidat,_(0_/_N)_?")
15    fichier.close()

```

- Définir la fonction `admis()` qui permet créer le fichier `admis.txt` comportant les données relatives aux candidat admis

```

1 def admis():
2     fichier=open("concours.txt")
3     dest=open("admis.txt","a")
4     for ligne in fichier:
5         L=ligne.split(";")
6         if L[4].strip()=="admis(e)":
7             dest.write(ligne)
8     fichier.close()
9     dest.close()

```

- Afin de sélectionner en priorité les candidats admis et âgés moins de 30 ans, créer la fonction `attente()` qui produira à partir du fichier `admis.txt`, un nouveau fichier intitulé `attente.txt` comportant les données relatives aux candidats admis et âgés plus que 30 ans. Une ligne du fichier `attente.txt` comprend le **NCIN**, le **NOM** et **PRENOM** d'un candidat séparés par point virgule (;).

```

1 def attente():
2     fichier=open("admis.txt")
3     dest=open("attente.txt","a")
4     for ligne in fichier:
5         L=ligne.split(";")
6         if int(L[3])>=30:
7             enreg=L[0]+";"+L[1]+";"+L[2)+"\n"
8             dest.write(enreg)
9     fichier.close()
10    dest.close()

```

4. Définir la fonction **statistiques(dec)** qui permet de retourner le pourcentage des candidats pour la décision **dec** (**admis, refusé et ajourné**).

**Exemple :** Le pourcentage des candidats admis = ( Nombre des candidats admis / Nombre des candidats)\*100

```

1 def statistiques(dec):
2     fichier=open("concours.txt")
3     L=fichier.readlines()
4     fichier.close()
5     L1=[] # candidats admis
6     L2=[] # candidats refuses
7     L3=[] # candidats ajournes
8     for ligne in L:
9         L=ligne.split(";")
10        if L[4].strip()=="admis(e)":
11            L1.append(ligne)
12        elif L[4].strip()=="refuse(e)":
13            L2.append(ligne)
14        else:
15            L3.append(ligne)
16    if dec=="admis":
17        return (len(L1)/len(L))*100
18    elif dec=="refuse":
19        return (len(L2)/len(L))*100
20    else:
21        return (len(L3)/len(L))*100

```

5. Définir la fonction **supprimer()** qui supprimera du fichier **admis.txt** les candidat âgés plus que 30

```

1 def supprimer():
2     fichier=open("admis.txt")
3     candidat=[] # contient les candidats restants
4     for ligne in fichier:
5         L=ligne.split(";")
6         if int(L[3])<30:
7             candidat.append(ligne)
8     fichier.close()
9     #reecrire la nouvelle liste
10    fichier=open("admis.txt","w")
11    fichier.writelines(candidat)
12    fichier.close()

```

## Exercice 2

Après la réussite au baccalauréat, les meilleurs bacheliers sont orientés vers les classes préparatoires aux grandes écoles. En effet, la priorité est à celui qui a le score (nombre de points) le plus élevé. Ce score est appelé formule globale.

Les élèves admissibles seront classés par ordre décroissant selon la formule globale, Puis, une fois classés, ces bacheliers seront divisés en 4 groupes de la façon suivante : Dans le répertoire **C :/bachelier**, on

Groupe	description	Nb d'étudiants	Répartition
<b>Principale</b>	Liste principale	30	40% (SMA) ; 30% (SMB) ; 20% (PC) ; 10% (SVT)
<b>Tranche 1</b>	Liste d'attente N°1	40	50% (SMA) ; 25% (SMB) ; 20% (PC) ; 5% (SVT)
<b>Tranche 2</b>	Liste d'attente N°2	60	40% (SMA) ; 30% (SMB) ; 20% (PC) ; 10% (SVT)
<b>Tranche 3</b>	Liste d'attente N°3	50	55% (SMA) ; 25% (SMB) ; 20% (PC) ;

dispose d'un fichier nommé '**PSI.txt**' contenant la liste des bacheliers admissibles de la section Physique et sciences industrielles. Dans ce fichier, chaque bachelier est défini par :

- **Num\_insc** : le numéro d'inscription
- **NP** : le nom et prénom
- **FILIERE** : le nom de la filière (SMA, SMB, PC ou SVT)
- **MG** : moyenne générale.
- **FS** (formule spécifique) : un réel déjà calculé à partir des notes obtenues dans les diverses matières.
- **i** : un réel=1 si l'élève est redoublant en BAC et 1.10 sinon

On souhaite réaliser les fonctions suivantes :

1. **formule\_gen()** qui permet de créer un autre fichier '**PSI\_FG.txt**', à partir du fichier '**PSI.txt**', et y stocker, pour les mêmes bacheliers, les informations suivantes : **Num\_insc**, **NP**, **FILIERE** et **FG**. Sachant que la formule globale (FG) de chaque élève est calculée par l'équation :  

$$FG = ((5 * MG + FS) * i)$$

```

1 def formule_gen():

```

```

2   source=open("PSI.txt")
3   dest=open("PSI_FG.txt","a")
4   for ligne in source:
5       s=ligne.strip()
6       tab=s.split(":")
7       fg=((5*float(tab[3])+float(tab[4]))*float(tab[5]))
8       etd=str(tab[0])+":"+str(tab[1])+":"+str(tab[2])+":"+str(fg)+"\n"
9       dest.write(etd)
10  source.close()
11  dest.close()

```

2. **classer()** qui permet de classer les bacheliers du fichiers 'PSI\_FG.txt' par ordre décroissant selon la formule globale(FG).

*N.B : pour faire cette tache, on doit transmettre les informations dans un tableau, les trier puis les transmettre ordonnées dans le fichier, implémenter un algorithme de tri que vous voulez pour trier le tableau*

```

1  def classer():
2      source=open("PSI_FG.txt")
3      L=source.readlines()
4      for i in range(len(L)-1):
5          m=i
6          for j in range(i+1,len(L)):
7              t1=L[j].strip().split(":")
8              t2=L[m].strip().split(":")
9              if float(t1[3])>float(t2[3]):
10                 m=j
11             L[i],L[m]=L[m],L[i]
12         source.close()
13         source=open("PSI_FG.txt","w")
14         source.writelines(L)
15         source.close()

```

3. **generer()** qui permet d'extraire, dans le même répertoire, 4 autres fichiers ('PSI\_princ.txt', 'PSI\_t1.txt', 'PSI\_t2.txt', 'PSI\_t3.txt') contenant respectivement les bacheliers appartenant au groupe liste principale, tranche 1, tranche2 et tranche3.

```

1  def generer():
2      l1=[]#SMA
3      l2=[]#SMB
4      l3=[]#PC
5      l4=[]#SVT
6      #classer()
7      #classer dans 3 listes les etudiants selon la filiere
8      source=open("PSI_FG.txt")
9      for ligne in source:
10         t=ligne.split(":")
11         if t[2]=="SMA":

```

```
12     l1.append(ligne)
13     elif t[2]=="SMB":
14         l2.append(ligne)
15     elif t[2]=="PC":
16         l3.append(ligne)
17     else:
18         l4.append(ligne)
19     source.close()
20     # construire la liste principale
21     nb1=12 # nombre des etudiants SMA
22     nb2=9 # nombre des etudiants SMB
23     nb3=6 # nombre des etudiants PC
24     nb4=3 # nombre des etudiants SVT
25     isma=0 # indice de debut pour SMA
26     ismb=0 # indice de debut pour SMB
27     ipc=0 # indice de debut pour PC
28     isvt=0 # indice de debut pour SVT
29     princ=open("PSI_princ.txt","a")
30     for i in range(nb1):
31         if i<len(l1):
32             princ.write(l1[i])
33     for i in range(nb2):
34         if i<len(l2):
35             princ.write(l2[i])
36     for i in range(nb3):
37         if i<len(l3):
38             princ.write(l3[i])
39     for i in range(nb4):
40         if i<len(l4):
41             princ.write(l4[i])
42     princ.close()
43     # construire la tranche 1
44     isma=nb1
45     ismb=nb2
46     ipc=nb3
47     isvt=nb4
48     nb1+=20
49     nb2+=10
50     nb3+=8
51     nb4+=2
52     princ=open("PSI_t1.txt","a")
53     for i in range(isma,nb1):
54         if i<len(l1):
55             princ.write(l1[i])
56     for i in range(ismb,nb2):
57         if i<len(l2):
58             princ.write(l2[i])
```

```
59     for i in range(ipc, nb3):
60         if i < len(l3):
61             princ.write(l3[i])
62     for i in range(isvt, nb4):
63         if i < len(l4):
64             princ.write(l4[i])
65     princ.close()
66     # construire la tranche 2
67     isma=nb1
68     ismb=nb2
69     ipc=nb3
70     isvt=nb4
71     nb1+=24
72     nb2+=18
73     nb3+=12
74     nb4+=6
75     princ=open("PSI_t2.txt", "a")
76     for i in range(isma, nb1):
77         if i < len(l1):
78             princ.write(l1[i])
79     for i in range(ismb, nb2):
80         if i < len(l2):
81             princ.write(l2[i])
82     for i in range(ipc, nb3):
83         if i < len(l3):
84             princ.write(l3[i])
85     for i in range(isvt, nb4):
86         if i < len(l4):
87             princ.write(l4[i])
88     princ.close()
89     # construire la tranche 3
90     isma=nb1
91     ismb=nb2
92     ipc=nb3
93     nb1+=27
94     nb2+=12
95     nb3+=11
96     princ=open("PSI_t3.txt", "a")
97     for i in range(isma, nb1):
98         if i < len(l1):
99             princ.write(l1[i])
100    for i in range(ismb, nb2):
101        if i < len(l2):
102            princ.write(l2[i])
103    for i in range(ipc, nb3):
104        if i < len(l3):
105            princ.write(l3[i])
```

```

106     for i in range(isvt,nb4):
107         if i<len(l4):
108             princ.write(l4[i])
109     princ.close()

```

4. **afficher(Num\_insc)** qui permet d'afficher, pour un candidat donné, en fonction de son numéro d'inscription (donnée fournie en argument), le groupe auquel il appartient

```

1  def afficher(Num\_insc):
2      princ=open("PSI\_princ.txt")
3      L1=princ.readlines()
4      tr1=open("PSI\_t1.txt")
5      L2=tr1.readlines()
6      tr2=open("PSI\_t2.txt")
7      L3=tr2.readlines()
8      tr3=open("PSI\_t3.txt")
9      L4=tr3.readlines()
10     etat=False
11     tranche=""
12     # chercher dans la liste principale
13     for ligne in L1:
14         if str(Num\_insc)==ligne[:len(str(Num\_insc))]:
15             etat=True
16             tranche="Liste_principale"
17             break
18     # si l etudiant n appartient pas a liste principale rechercher dans la t
19     if etat==False:
20         for ligne in L2:
21             if str(Num\_insc)==ligne[:len(str(Num\_insc))]:
22                 etat=True
23                 tranche="Tranche_1"
24                 break
25     # si l etudiant n appartient pas a tranche 1 rechercher dans la tranche 2
26     if etat==False:
27         for ligne in L3:
28             if str(Num\_insc)==ligne[:len(str(Num\_insc))]:
29                 etat=True
30                 tranche="Tranche_2"
31                 break
32     # si l etudiant n appartient pas a tranche 2 rechercher dans la tranche 3
33     if etat==False:
34         for ligne in L4:
35             if str(Num\_insc)==ligne[:len(str(Num\_insc))]:
36                 etat=True
37                 tranche="Tranche_3"
38                 break
39     return (etat, tranche)

```