

<b>Lycée Ras Jebel</b> ***** <b>DEVOIR DE SYNTHESE N° 2</b>	<b>Matière : Informatique</b>	
	<b>Classes : 4<sup>ème</sup> Math. &amp; 4<sup>ème</sup> Sc.Exp2</b>	<b>Coefficient : 1</b>
	<b>Date : 16/03/2024</b>	<b>Durée : 1 h 30 mn</b>
	<b>Enseignant : M<sup>me</sup> Nejla Mahjoub Abdellatif</b>	
<b>Nom &amp; prénom : .....</b>		<b>Note : ..... / 20</b>

**Exercice n°1 : (3 points)**

Soient les deux algorithmes **PP1** et **PP2** appelant respectivement les deux modules **tri1** et **tri2** permettant de **mettre en ordre** les caractères d'une chaîne dans la **même** chaîne :

<b>Algorithme PP1</b> <b>Début</b> Lire (ch) Ecrire(Tri1(ch)) Ecrire(ch) <b>Fin</b>	Module <b>Tri1</b> (s : chaîne) : chaîne <b>Début</b> <b>Pour</b> i de 1 à long (s)-1 <b>faire</b> v←s[i] j←i <b>Tant que</b> s[j-1] >v <b>faire</b> s[j]←s[j-1] j←j-1 <b>Fin tant que</b> s[j]←v <b>Fin pour</b> <b>Retourner</b> s <b>Fin</b>	<b>Algorithme PP2</b> <b>Début</b> Lire (ch) Tri2(ch) Ecrire(ch) <b>Fin</b>	Module <b>Tri2</b> (@s : chaîne) <b>Début</b> <b>Pour</b> i de 1 à long (s)-1 <b>faire</b> v←s[i] j←i <b>Tant que</b> s[j-1] >v <b>faire</b> s[j]←s[j-1] j←j-1 <b>Fin tant que</b> s[j]←v <b>Fin pour</b> <b>Fin</b>
--	---	--	---

**Questions :**

- Qu'appelle-t-on la méthode de tri utilisée ?  
**Le tri par insertion**
- Déduire la nature des modules utilisés (fonction ou procédure)  
**Tri1 : fonction**  
**Tri2 : procédure**
- Étant donnée ch= "module " :
  - Donner les résultats d'exécution des programmes PP1 et PP2,
  - Est-ce que les modules appelés modifient la valeur de ch ? (Répondre par **Oui** ou **Non**)
  - Si **Non**, apporter les modifications nécessaires sur l'un ou les deux programmes PP1 et PP2 afin de **modifier** le contenu de ch **après** l'appel.

	PP1	PP2
a)	<b>edlmou</b> <b>module</b>	<b>edlmou</b>
b)	<b>Non</b>	<b>Oui</b>
c)	<b>ch←Tri1(ch)</b> <b>Ecrire(ch)</b>	

## Exercice n°2 : (6 points)

Afin de pouvoir **insérer** un entier **a** dans un tableau **T** de **n** entiers **triés** par ordre croissant tout en gardant les éléments **en ordre**, on vous propose l'algorithme **insertion** suivant :

**Algorithme** insertion

**Début**

Saisie\_n(n)  
Remplir (T,n)  
Tri(T,n)  
Saisie\_a(a)  
Insérer (T,n,a)  
Afficher(T,n)

**Fin**

**TDNT**

Type
Tab=tableau de 50 entier

**TDO Globaux**

Objet	Type/nature
T	Tab
N	Entier
Saisie_n	Procédure
Remplir	Procédure
Tri	Procédure
Saisie_a	Procédure
Insérer	Procédure
Afficher	Procédure

Procédure	Rôle
Saisie_n	Saisir n entre 5 et 50
Remplir	Remplir T par n entiers
Tri	Trier le tableau T par ordre croissant
Saisie_a	Saisir un entier a
Insérer	Insérer l'entier a à sa bonne position dans le tableau T
Afficher	Afficher les entiers de T après insertion de a

**Questions :**

Soient **n=8, a=15**

Et T=

12	32	23	-6	12	45	5	12
1	2	3	4	5	6	7	8

a) Donner le résultat d'exécution du programme insertion :

T=

-6	5	12	12	12	15	23	32	45
1	2	3	4	5	6	7	8	9

b) Ecrire l'algorithme de la procédure **insérer** ainsi que son TDO Locaux

**Procédure** inserer (@ V : Tab, @ x:entier,y:entier)

**Début**

p←0, i←0  
**Tant que** (i<x et V[i]<y) **faire**  
    i←i+1  
**Fin tant que**  
p←i  
x←x+1  
**Pour** i de x à p+1 (pas=-1) **faire**  
    V[i]←V[i-1]  
**Fin Pour**  
V[p]←y

**Fin**

**TDO locaux**

Objet	Type/nature
P,i	Entier/variable

## Exercice n°3 : (11 points)

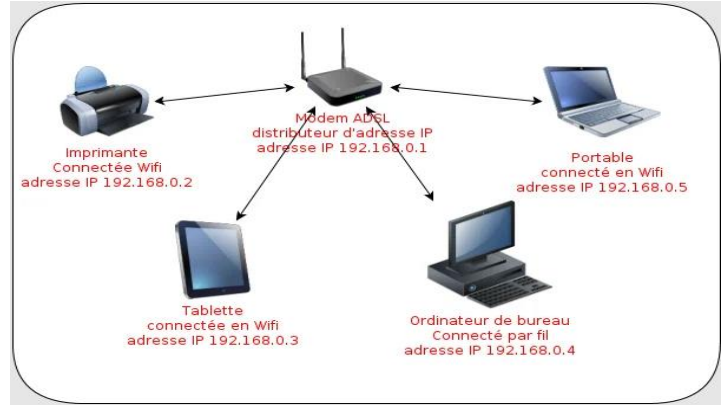
Dans les réseaux informatiques, l'**adresse IP** (Internet Protocol) d'une machine est un numéro d'identification (dans le réseau) qui permet à la machine de communiquer avec les autres machines du réseau. L'adresse IP d'une machine est une séquence de **4 nombres**, chaque nombre étant un entier entre **0 et 255**. Elle a généralement la forme suivante : **W.X.Y.Z**

Pour des raisons de lisibilité, les entiers de l'adresse IP sont séparés par des **points**.

Par exemple **86.215.169.62** est une adresse IP.

Selon le premier nombre (**W**) on peut classer les adresses IP en cinq classes (**A, B, C, D, E**) :

Classe	Valeur de W
Classe A	Entre 0 et 127 inclus
Classe B	Entre 128 et 191 inclus
Classe C	Entre 192 et 223 inclus
Classe D	Entre 224 et 239 inclus
Classe E	Entre 240 et 255 inclus



### Travail demandé :

On désire écrire un programme **adresseIP** qui permet de :

- ✚ Saisir un entier **n** (avec  $4 \leq n \leq 20$ ),
- ✚ Remplir un tableau T par des adresses IP **valides** ayant le format **W.X.Y.Z** (avec **W, X, Y, Z** sont des entiers compris entre **0 et 255**),
- ✚ Afficher pour **chaque** classe ses adresses IP correspondantes (on affiche **seulement** les classes ayant des adresses IP dans le tableau T)

**Exemple :** Pour  $n=7$

T :	122.4.32.156	155.200.250.3	245.12.100.3	240.3.1.0	225.13.12.10	244.1.2.3	32.0.0.1
	1	2	3	4	5	6	7

L'ordinateur affiche :

```

Classe A: 122.4.32.156|32.0.0.1|
Classe B: 155.200.250.3|
Classe D: 225.13.12.10|
Classe E: 245.12.100.3|240.3.1.0|244.1.2.3|
    
```

### Questions :

1. Décomposer le problème en modules,
2. Elaborer l'algorithme du programme principal ainsi que les algorithmes des modules envisagés,
3. Dresser le TDO globaux ainsi que les TDO locaux

1) Algorithme de programme principal :

**Algorithme** adresseIP

**Début**

n ← Saisir()  
Remplir (T,n)  
Afficher(T,n)

**Fin**

TDNT

Type
Tab=tableau de 50 entier

TDO Globaux

Objet	Type/nature
T	Tab
N	Entier
Saisir	Fonction
Remplir	Procédure
Afficher	Procédure

2) Les algorithmes des modules :

Algorithme de la fonction saisir

**fonction** Saisir ( ) :entier

**Début**

**Répéter**

Ecrire ("donner un entier compris entre 4 et 20 :")

Lire (n)

**Jusqu'à n** ∈ [4..20]

**Retourner n**

**Fin**

TDO locaux

Objet	Type/nature
n	Entier/variable

Algorithme de la procédure afficher :

**Procédure** Afficher ( T :Tab, n :entier)

**Début**

**Pour j** de "A" à "E" faire

V[j] ← ""

**Fin Pour**

**Pour i** de 1 à n faire

w ← Valeur(sous-chaine(T[i],0,Pos(".",T[i])))  
selon w

0..127: V["A"] ← V["A"]+T[i]+"|"

128..191: V["B"] ← V["B"]+T[i]+"|"

192..223: V["C"] ← V["C"]+T[i]+"|"

224..239: V["D"] ← V["D"]+T[i]+"|"

240.. 255: V["E"] ← V["E"]+T[i]+"|"

**Fin selon**

**Fin Pour**

**Pour i** de "A" à "E" faire

**Si** V[i] ≠ "" alors

Ecrire ("Classe ",i,":",V[i])

**Fin si**

**Fin Pour**

**Fin**

Algorithme de la procédure Remplir

**Procédure** Remplir (@T :Tab, n :entier)

**Début**

**Pour i** de 1 à n faire

T[i] ← ""

**Pour j** de 1 à 4 faire

Ecrire ("donner l'entier n°",j,":")

nb ← saisie\_nb()

T[i] ← T[i]+convch(nb)+"."

**Fin Pour**

T[i] ← sous-chaine(T[i], 0, long(T[i])-1)

**Fin Pour**

**Fin**

TDO locaux

Objet	Type/nature
I,j,nb	Entier/variable
Saisie_nb	fonction

Algorithme de la fonction saisie\_nb :

**fonction** saisie\_nb ( ) :entier

**Début**

**Répéter**

Ecrire ("donner un entier compris entre 0 et 255 :")

Lire (nb)

**Jusqu'à nb** ∈ [0 .. 255]

**Retourner nb**

**Fin**

TDO locaux

Objet	Type/Nature
nb	Entier/variable

TDO locaux de la procédure afficher

Objet	Type/Nature
I	Entier/variable
J	Caractère/variable
V	Tableau de 5 chaîne

Exercice n°3 :

## Les programmes Python

### Exercice n°1 :

```
1 def tri(s):
2     for i in range(1,len(s)):
3         v=s[i]
4         j=i
5         while j>0 and s[j-1]>v:
6             s=s[:j]+s[j-1]+s[j+1:]
7             j=j-1
8         s=s[:j]+v+s[j+1:]
9     return s
10 ch=input("donner une chaine:")
11 ch=tri(ch)
12 print(ch)
```

### Exercice n°2 :

```
22 def inserer(T,a):
23     global n
24     i=0
25     while T[i]<a and i<n:
26         i=i+1
27     p=i
28     n=n+1
29     for i in range(n,p,-1):
30         T[i]=T[i-1]
31     T[p]=a
```

```
1 from numpy import array
2 T=array([str]*20)
3 def saisie_n():
4     n=int(input("donner un entier entre 5 et 50:"))
5     while n<4 or n>20:
6         n=int(input("donner un entier entre 5 et 50:"))
7     return n
8 def saisie_nb():
9     nb=int(input("donner un entier entre 0 et 255:"))
10    while nb<0 or nb>255:
11        nb=int(input("donner un entier entre 0 et 255:"))
12    return nb
13 def remplir(T,n):
14    for i in range(n):
15        T[i]=" "
16        for j in range (4):
17            print("donner l'entier n°",j+1,":")
18            x=saisie_nb()
19            T[i]=T[i]+str(x)+ "."
20            T[i]=T[i][:len(T[i])-1]
21 def afficher(T,n):
22    V=array([str]*5)
23    for i in range(5):
24        V[i]=" "
25    for i in range(n):
26        w=int(T[i][:T[i].find(".")])
27        match w:
28            case w if 0<=w<=127:
29                V[0]=V[0]+T[i]+"|"
30            case w if 128<=w<=191:
31                V[1]=V[1]+T[i]+"|"
32            case w if 192<=w<=223:
33                V[2]=V[2]+T[i]+"|"
34            case w if 224<=w<=239:
35                V[3]=V[3]+T[i]+"|"
36            case w if 240<=w<=255:
37                V[4]=V[4]+T[i]+"|"
38    for i in range(5):
39        if V[i]!="":
40            print("Classe ",chr(i+65),":",V[i])
41 #Programme Principal
42 n=saisie_n()
43 remplir(T,n)
44 afficher(T,n)
```