

Exercice 1:

Ecrire un algorithme et son implémentation en Python d'un programme qui permet de :

- Saisir un tableau T1 par N chaines ($2 \leq N \leq 25$)
- Convertir toutes les chaines obtenues en majuscules puis les mettre dans un tableau T2
- Afficher le tableau T2

Exemple :Pour le tableau T1 suivant, avec $n=3$:

Ahmed	mahdi	la1
-------	-------	-----

Le programme affiche :

AHMED	MAHDI	LA1
-------	-------	-----

Exercice 2

Ecrire un algorithme et son implémentation en Python d'un programme qui permet de :

- Saisir un tableau T1 par N entiers de deux chiffres ($2 \leq N \leq 10$)
- Chercher tous les diviseurs de chaque élément du tableau puis les mettre dans un tableau T2
- Afficher le tableau T2

Exemple :Pour le tableau T1 suivant, avec $n=3$:

6	43	5
---	----	---

Le programme affiche :

12	44	6
----	----	---

Exercice 3:

Ecrire un algorithme et son implémentation en Python d'un programme qui permet de :

- Saisir un tableau T1 par N entiers ($2 \leq N \leq 10$)
- Chercher tous les entiers premiers de chaque élément du tableau puis les mettre dans un tableau T2
- Afficher le tableau T2

Exemple :Pour le tableau T1 suivant, avec $n=6$:

10	2	18	7	43	10
----	---	----	---	----	----

Le programme affiche :

2	7	43
---	---	----

Exercice 4:

Ecrire un script en Python qui permet de :

A/-Saisir aléatoirement un tableau T1 par N entiers de deux chiffres ($2 \leq N \leq 10$)

B/-Chercher la valeur maximale de ce tableau

Exemple :

Pour le tableau T1 suivant, avec $n=6$:

10	12	18	17	43	10
----	----	----	----	----	----

Le programme affiche :

La valeur maximale est : 43

C/-Saisir deux indices i et j puis permuter les deux éléments du tableau d'indice i et j

Exemple :

Donner i : 0

Donner j : 2

Le nouveau tableau après permutation de $t[0]$ et $t[2]$ est

18	12	10	17	43	10
----	----	----	----	----	----

D/-Vérifier l'existence d'un élément e dans le tableau

Exemple

Donner l'élément à rechercher : 23

False

Exercice 5

A/Ecrire un programme qui permet de saisir un entier N puis de déterminer la nature de cet entier pair ou impaire(Routner une chaîne de caractère '**pair**' ou '**impair**').

C/Répondre en A mais retourner **True** si pair ou **False** dans le cas contraire

D/Redéfinir la fonction **estnum()** (**verifier si une chaîne est numérique ou non**)

E/ Ecrire l'algorithme et l'implémentation en **Python** d'un programme qui permet de :

- Remplir un tableau T de n chaînes de caractères($n > 3$) de longueur 5 caractères maximum
- Afficher les chaînes composés uniquement par des chiffres

Exemple

T

'1014'	'A29'	'5L79'	'20'
--------	-------	--------	------

Alors on affiche

'1014'

'20'

F/Définir une fonction qui détermine si une chaîne se compose par des caractères minuscules ou non.

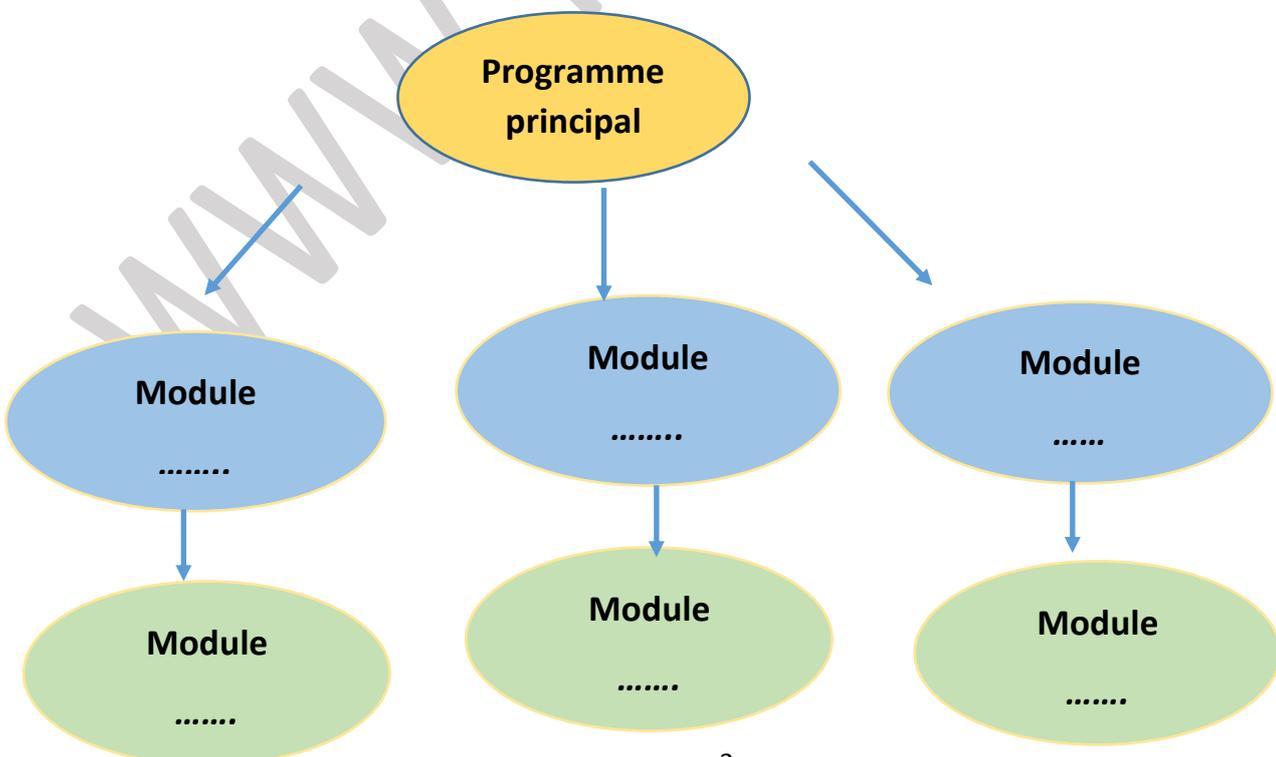
Exercice 6

Soit à saisir les moyennes de n élèves ($2 \leq n \leq 30$) dans un tableau. Il s'agit de trouver puis d'afficher le nombre d'élèves qui ont réussi.

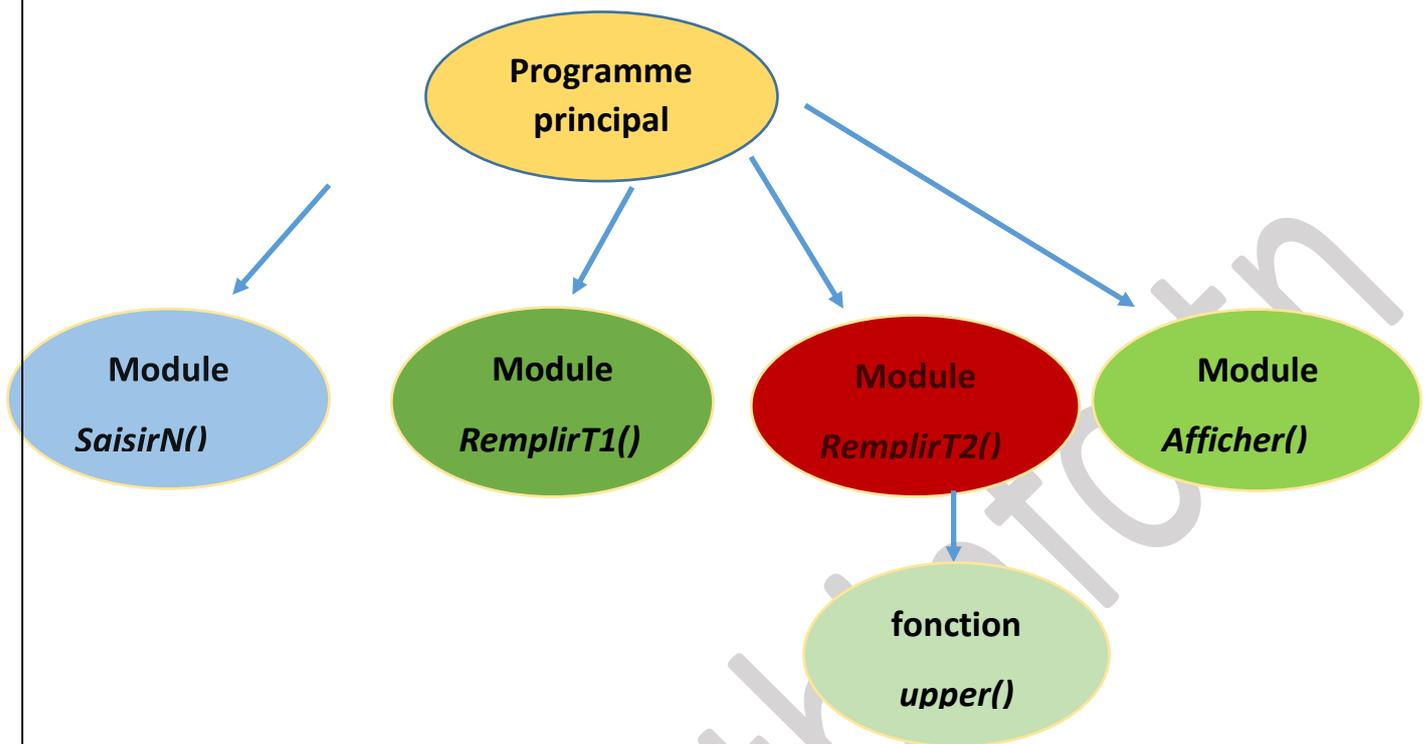
Module saisir n :

Module remplir le tableau :

Module Nombre :



Correction exercice 1



```
from numpy import array
def saisirN():
    valide=False
    while valide==False:
        n=int(input('Taille tableau '))
        valide=2<=n<=25
    return n
def remplirT1(n):
    T1=array([str]*n)
    for i in range(n):
        T1[i]=str(input('Donner element du tableau '))
    return T1
```

```
def remplirT2(T1,n):
    T2=array([str]*n)
    for i in range(n):
        T2[i]=T1[i].upper()
    return T2
def afficher(t,n):
    for i in range(n):
        print(t[i])
#programme principale
n=saisirN()
T1=remplirT1(n)
afficher(T1,n)
T2=remplirT2(T1,n)
afficher(T2,n)
```

Algorithmique	Implémentation en python										
<p>1. Programme principal Algorithme Ex1 Debut saisirN(n) RemplirT1(T1,n) afficher(T1,n) remplirT2(T2,T1,n) afficher(T2,n) Fin</p> <p>2. Tableau de declarations des nouveaux types (T.D.N.T)</p> <table border="1"> <tr> <td>Type</td> </tr> <tr> <td>Tab = tableau de 25 chaines</td> </tr> </table> <p>3. Tableau de declarations des objets globaux (T.D.O.G)</p> <table border="1"> <thead> <tr> <th><u>Objet</u></th> <th><u>Type / Nature</u></th> </tr> </thead> <tbody> <tr> <td>T1</td> <td>Tab</td> </tr> <tr> <td>T2</td> <td>Tab</td> </tr> <tr> <td><u>n</u></td> <td><u>Entier</u></td> </tr> </tbody> </table>	Type	Tab = tableau de 25 chaines	<u>Objet</u>	<u>Type / Nature</u>	T1	Tab	T2	Tab	<u>n</u>	<u>Entier</u>	<pre>#programme principale n=saisirN() T1=remplirT1(n) afficher(T1,n) T2=remplirT2(T1,n) afficher(T2,n)</pre>
Type											
Tab = tableau de 25 chaines											
<u>Objet</u>	<u>Type / Nature</u>										
T1	Tab										
T2	Tab										
<u>n</u>	<u>Entier</u>										

Algorithmique	Implémentation en python
<p>4. Procedure « saisirN » : Procedure saisirN(@n :entier) Debut Repete Ecrire ("Taille tableau ") Lire (n) Jusqu'à 2<=n<=25) Fin</p>	<pre>def saisirN(): valide=False while valide==False: n=int(input('Taille tableau ')) valide=2<=n<=25 return n</pre>

Algorithmique	Implémentation en python						
<p>5. Procédure « remplirT1 » : Procédure remplirT1 (@t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Ecrire (" 'Donner element du tableau :") Lire (t[i]) Fin Pour Fin</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def remplirT1(n): T1=array([str]*n) for i in range(n): t[i]=input("Donner element du tableau :") return T1</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Algorithmique	Implémentation en python						
<p>6. Procédure « remplirT2 » : Procédure remplirT2 (@T2 : Tab,t1 :tab,n :entier) Debut Pour i de 0 à n-1 faire T2[i]←majus(t1[i]) Fin Pour Fin</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def remplirT2(T2,n): T2=array([str]*n) for i in range(n): T2[i]=T1[i].upper() return T2</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Algorithmique	Implémentation en python						
<p>7. Procédure « afficher » : Procédure afficher (t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Ecrire (t[i]) Fin Pour Fin</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def afficher(t,n): for i in range(n): print(t[i])</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Exercice 2

Algorithmique	Implémentation en python										
<p>1 Programme principal</p> <p>Algorithme Ex1 Debut saisirN(n) RemplirT1(T1,n) remplirT2(T2,T1,n) afficher(T2,n) Fin</p> <p>Tableau de declarations des nouveaux types (T.D.N.T)</p> <table border="1"> <tr> <td>Type</td> </tr> <tr> <td>Tab = tableau de 25 chaines</td> </tr> </table> <p>2. Tableau de declarations des objets globaux (T.D.O.G)</p> <table border="1"> <thead> <tr> <th><u>Objet</u></th> <th><u>Type / Nature</u></th> </tr> </thead> <tbody> <tr> <td>T1</td> <td>Tab</td> </tr> <tr> <td>T2</td> <td>Tab</td> </tr> <tr> <td><u>n</u></td> <td><u>Entier</u></td> </tr> </tbody> </table>	Type	Tab = tableau de 25 chaines	<u>Objet</u>	<u>Type / Nature</u>	T1	Tab	T2	Tab	<u>n</u>	<u>Entier</u>	<pre>#programme principale n=saisirN() T1=remplirT1(n) T2=remplirT2(T1,n) afficher(T2,n)</pre>
Type											
Tab = tableau de 25 chaines											
<u>Objet</u>	<u>Type / Nature</u>										
T1	Tab										
T2	Tab										
<u>n</u>	<u>Entier</u>										

Algorithmique	Implémentation en python
<p>3. Procédure « saisirN » :</p> <p>Procédure saisirN(@n :entier) Debut Repete Ecrire ("Taille tableau ") Lire (n) Jusqu'à 2<=n<=10) Fin</p>	<pre>def saisirN(): valide=False while valide==False: n=int(input('Taille tableau ')) valide=2<=n<=10 return n</pre>

Algorithmique	Implémentation en python						
<p>4. Procédure « remplirT1 » : Procédure remplirT1 (@t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Repete Ecrire (" Donner element du tableau : ") Lire (t[i]) Jusqu'à 10<=t[i]<=99 Fin Pour Fin</p> <table border="1" data-bbox="453 568 796 730"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def remplirT1(n): T1=array([str]*n) for i in range(n): while valide==False: t[i]=int(input("Donner element du tableau : ")) valide=10<=t[i]<=99 return T1</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Algorithmique	Implémentation en python								
<p>5. Fonction « diviseur » : Fonction diviseur (n :entier) Debut s←0 for i in range (1,N+1): if n%i=0: s←s+i retourner s Fin</p> <table border="1" data-bbox="592 1072 935 1274"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> <tr> <td>s</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	s	Entier	<pre>def diviseur(n): s=0 for i in range (1,N+1): if n%i==0: s=s+i return s</pre>
TDO Locaux									
Objet	Type/Nature								
i	Entier								
s	Entier								

Algorithmique	Implémentation en python						
<p>6. Procédure « remplirT2 » : Procédure remplirT2 (@T2 : Tab, t1 : tab, n : entier) Debut Pour i de 0 à n-1 faire T2[i] ← diviseur(t1[i]) Fin Pour Fin</p> <table border="1" data-bbox="592 322 936 479"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def remplirT2(T2,n): T2=array([str]*n) for i in range(n): T2[i]=diviseur(T1[i]) return T2</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Algorithmique	Implémentation en python						
<p>7. Procédure « afficher » : Procédure afficher (t : Tab, n : entier) Debut Pour i de 0 à n-1 faire Ecrire (t[i]) Fin Pour Fin</p> <table border="1" data-bbox="571 931 895 1088"> <thead> <tr> <th colspan="2">TDO Locaux</th> </tr> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def afficher(t,n): for i in range(n): print(t[i])</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Exercice 3

Algorithmique	Implémentation en python
<p>Procédure « saisirN » : Procédure saisirN(var n : entier) Debut Repete Ecrire ("Taille tableau ") Lire (n) Jusqu'à 2 ≤ n ≤ 10 Fin</p>	<pre>def saisirN(): valide=False while valide==False: n=int(input('Taille tableau ')) valide=2<=n<=10 return n</pre>

Algorithmique	Implémentation en python						
<p>Procédure « remplirT » : Procédure remplirT (@t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Ecrire (" Donner element du tableau : ") Lire (t[i]) Fin Pour Fin</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th colspan="2" style="text-align: center;">TDO Locaux</th> </tr> <tr> <th style="text-align: left;">Objet</th> <th style="text-align: left;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	<pre>def remplirT1(n): T1=array([str]*n) for i in range(n): t[i]=input("Donner element du tableau :") return T1</pre>
TDO Locaux							
Objet	Type/Nature						
i	Entier						

Algorithmique	Implémentation en python														
<p>8. Programme principal Algorithme principal Debut saisirN(n) remplirT(T,n) afficher(T,n) Fin</p> <p>2. Tableau de declarations des nouveaux types (T.D.N.T)</p> <table border="1" style="margin-left: 20px;"> <tr> <td>Type</td> </tr> <tr> <td>Tab = tableau de n entiers</td> </tr> </table> <p>8. Tableau de declarations des objets globaux (T.D.O.G)</p> <table border="1" style="margin-left: 20px;"> <thead> <tr> <th style="text-align: center;"><u>Objet</u></th> <th style="text-align: center;"><u>Type / Nature</u></th> </tr> </thead> <tbody> <tr> <td><u>n</u></td> <td><u>entier</u></td> </tr> <tr> <td><u>T</u></td> <td><u>Tab</u></td> </tr> <tr> <td><u>saisir</u></td> <td><u>procedure</u></td> </tr> <tr> <td><u>remplir</u></td> <td><u>procedure</u></td> </tr> <tr> <td><u>afficher</u></td> <td><u>procedure</u></td> </tr> </tbody> </table>	Type	Tab = tableau de n entiers	<u>Objet</u>	<u>Type / Nature</u>	<u>n</u>	<u>entier</u>	<u>T</u>	<u>Tab</u>	<u>saisir</u>	<u>procedure</u>	<u>remplir</u>	<u>procedure</u>	<u>afficher</u>	<u>procedure</u>	<pre><i>#programme principale</i> n=saisirN() T1=remplirT1(n) afficher(T1,n) T2=remplirT2(T1,n) afficher(T2,n)</pre>
Type															
Tab = tableau de n entiers															
<u>Objet</u>	<u>Type / Nature</u>														
<u>n</u>	<u>entier</u>														
<u>T</u>	<u>Tab</u>														
<u>saisir</u>	<u>procedure</u>														
<u>remplir</u>	<u>procedure</u>														
<u>afficher</u>	<u>procedure</u>														

Algorithmique	Python								
<p>Procédure « afficher » :</p> <p>Procédure afficher (t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Si (premier(t[i])==vrai)alors Ecrire (t[i]) Fin Si Fin Pour Fin</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2" style="text-align: center;">TDO Locaux</th> </tr> <tr> <th style="text-align: left;">Objet</th> <th style="text-align: left;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> <tr> <td>premier</td> <td>fonction</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	premier	fonction	<pre>def afficher (t,n): for i in range(n): if (premier(t[i])): print(t[i])</pre>
TDO Locaux									
Objet	Type/Nature								
i	Entier								
premier	fonction								

Algorithmique	Implémentation en python										
<p>Fonction « premier » :</p> <p>Fonction premier (n :entier):booleen Debut valide ← Faux Somme ← 0 Pour i de 1 a n faire Si n mod i==0 alors somme ← somme+1 FinSi Si somme==2 alors valide ← Vrai FinSi retourner valide Fin</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th colspan="2" style="text-align: center;">TDO Locaux</th> </tr> <tr> <th style="text-align: left;">Objet</th> <th style="text-align: left;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> <tr> <td>Somme</td> <td>Entier</td> </tr> <tr> <td>valide</td> <td>Booleen</td> </tr> </tbody> </table>	TDO Locaux		Objet	Type/Nature	i	Entier	Somme	Entier	valide	Booleen	<pre>def premier(n): valide= False somme=0 for i in range(1,n+1): if n % i==0 : somme=somme+1 if somme==2 : valide=True return valide</pre>
TDO Locaux											
Objet	Type/Nature										
i	Entier										
Somme	Entier										
valide	Booleen										

Exercice 4 :

```
from numpy import *
from random import randint

def saisir(): # fonction saisir n et remplir t1
    valide = False
    while valide == False:
        n = int(input('Donner n'))
        valide = 2 <= n <= 10
    return n

def remplir(n): # fonction afficher tableau
    t= array([int()] * n)
    for i in range(n):
        t[i] = randint(10, 99)
    return t

def afficher(t, n): # procedure afficher tableau
    for i in range(n):
        print('t[' , i, '] :', t[i])

def maximal(t, n): # fonction chercher la valeur maximale
    maxi = t[0]
    for i in range(n):
        if t[i] > maxi:
            maxi = t[i]
    return maxi

def permuter(t, i, j): # Module permuter deux éléments

    a = t[i]
    t[i] = t[j]
    t[j] = a

def verif(t1, e): # Module vérifier existence de e dans le tableau
    valide = False
    for i in range(n):
        if t1[i] == e:
            valide = True
    return valide
```

```

n = saisir() # appel du module saisir
t1 = remplir(n)
afficher(t1, n) # appel du module afficher
print(' max :',maximal(t1, n))# appel du module maximale

valide = False
while valide == False:
    i = int(input(' Donner indice i'))
    j = int(input(' Donner indice j'))
    valide = (i < n) and (j < n)

permuter(t1, i, j) # appel du module permuter
afficher(t1, n) ##appel du module afficher
e = int(input(' Donner element à chercher'))
print(verif(t1, e)) # appel du module verif

```

Exercice 6 :

```

from numpy import array

def saisir(): # module saisir n
    valide = False
    while valide == False:
        n = int(input('Donner n'))
        valide = 2 <= n <= 10
    return n

def remplir(n): # module saisir un tableau
    t= array([float()] * n)
    for i in range(n):
        t[i] = float(input('Donner la moyenne'))
    return t

def afficher(t, n): # procedure afficher tableau
    for i in range(n):
        print('t[' , i, '] :', t[i])

def nombre(t, n): # fonction le nombre des eleves admis
    nb=0
    for i in range(n):
        if t[i] >= 10:
            nb = nb+1
    return nb

n = saisir() # appel du module saisir
t1 = remplir(n)
afficher(t1, n) # appel du module afficher
print(' Nombre :',nombre(t1, n))# appel du module nombre

```