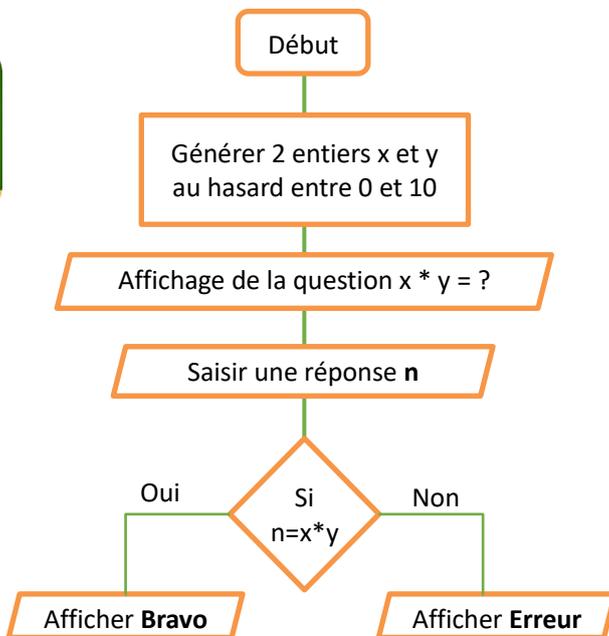
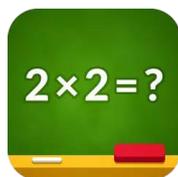


### Projet 1 :

On va réaliser un jeu pour apprendre la table de multiplication.  
 Référez-vous à l'algorithme ci-contre pour, Ecrire un algorithme puis une implémentation Python permettant de réaliser ce jeu.



On commence par générer 2 entiers (x et y) au hasard entre 0 et 10.

Point d'information

**Aléa** Permet de générer un entier au hasard.

Algorithme	Python
$x \leftarrow \text{Aléa}(0,10)$	<code>from random import * x=randint(0,10)</code>

Rq : Pour utiliser **randint** en Python, il faut commencer par importer la bibliothèque **random**

Point d'information

Pour affecter une valeur à une variable, on utilise le symbole d'affectation.

Algorithme	Python
$x \leftarrow 10$	<code>x=10</code>

Attention : ne pas confondre avec le test d'égalité

Algorithme	Python
Si $x=10$ alors	<code>if x==10 :</code>

On va avoir la solution suivante :

Algorithme	python
Algorithme multiplication Début $x \leftarrow \text{Aléa}(0,10)$ $y \leftarrow \text{Aléa}(0,10)$	<code>from random import * x=randint(0,10) y=randint(0,10)</code>

On veut maintenant afficher un message contenant la question  $x * y = ?$  et demander de l'utilisateur une réponse.

Point d'information

Pour afficher une variable ou un message :

Algorithme	Python
Ecrire("Message",x)	<code>print('Message',x)</code>

- En Python on peut délimiter un texte avec ' ' ou " "
- Exemple : `print("C'est un texte")`
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : `print('___',end='')`

Point d'information

Pour saisir une variable :

Algorithme	Python
lire(n)	<code>n=input()</code>

- En Python : saisir une chaîne (str) : `n=input()`  
saisir un entier (int) : `n=int(input())`  
saisir un réel (float) : `n=float(input())`
- Pour lire une variable en affichant un message : `n=input('Message=')`

On va avoir la solution suivante :

Algorithme	python
Algorithme multiplication Début $x \leftarrow \text{Aléa}(0,10)$ $y \leftarrow \text{Aléa}(0,10)$ Ecrire(x, " * ", y, " = ") Lire(n)	<code>from random import * x=randint(0,10) y=randint(0,10) print(x, ' * ', y, '=', end='') n=int(input())</code>

On veut afficher un message suivant la valeur entrée : « Bravo » si la réponse est juste et « erreur » sinon :

Algorithme	python™
Algorithme multiplication Début x ← Aléa(0,10) y ← Aléa(0,10) Ecrire(x, " * ", y, " = ") Lire(n) Si n=x*y alors Ecrire("Bravo") Sinon Ecrire("Erreur") FinSi Fin	<pre> from random import * x=randint(0,10) y=randint(0,10) print(x, ' * ', y, '=', end='') n=int(input()) if n==x*y :     print('Bravo') else :     print('Erreur')                     </pre>

**Structure conditionnelle si**

Algorithme	Python
<b>Si condition</b> alors Traitement1	<b>if condition :</b> ↔ traitement1
<b>Sinon</b> Traitement 2	<b>else :</b>
<b>FinSi</b>	↔ traitement2

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : print('\_\_\_',end='')

Pour bien s'entraîner sur la table de multiplication, on va répéter le jeu 5 fois.

**La boucle Pour**

Algorithme	Python
Pour i de 0 à n-1 faire Traitement FinPour	for i in range(n) : ↔ Traitement

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- range(4) donne 0 1 2 3    range(1,4) donne 1 2 3

Algorithme	python™
Algorithme multiplication Début Pour i de 0 à 4 faire x ← Aléa(0,10) y ← Aléa(0,10) Ecrire(x, " * ", y, " = ") Lire(n) Si n=x*y alors Ecrire("Bravo") Sinon Ecrire("Erreur") FinSi FinPour Fin	<pre> from random import * for i in range(5) :     x=randint(0,10)     y=randint(0,10)     print(x, ' * ', y, '=', end='')     n=int(input())     if n==x*y :         print('Bravo')     else :         print('Erreur')                     </pre>

On veut afficher le n° d'essai :

Algorithme	python™
Algorithme multiplication Début Pour i de 1 à 5 faire Ecrire("Essai",i) x ← Aléa(0,10) y ← Aléa(0,10) Ecrire(x, " * ", y, " = ") Lire(n) Si n=x*y alors Ecrire("Bravo") Sinon Ecrire("Erreur") FinSi FinPour Fin	<pre> from random import * for i in range(1,6) :     print('Essai : ',i)     x=randint(0,10)     y=randint(0,10)     print(x, ' * ', y, '=', end='')     n=int(input())     if n==x*y :         print('Bravo')     else :         print('Erreur')                     </pre>



Pour noter l'utilisateur on va ajouter un score (initialisé par 0, si bonne réponse le score s'incrémente par 1 sinon se décrémente par 1), afficher le score de chaque essai et enfin le score final :

Algorithme	python
Algorithme multiplication Début Score=0 Pour i de 1 à 5 faire Ecrire("Essai",i) x←Aléa(0,10) y←Aléa(0,10) Ecrire(x, " * ", y, " = ") Lire(n) Si n=x*y alors Ecrire("Bravo") Score←score+1 Sinon Ecrire("Erreur") Score←score-1 FinSi Ecrire("Score=",score) FinPour Ecrire("Score final=",score) Fin	<pre> from random import * score=0 for i in range(1,6) :     print('Essai : ',i)     x=randint(0,10)     y=randint(0,10)     print(x, ' * ', y, ' = ', end=' ')     n=int(input())     if n==x*y :         print('Bravo')         score=score+1     else :         print('Erreur')         score=score-1     print('score= ',score) print('score final= ',score)                     </pre>



L'utilisateur doit s'identifier au début du jeu à travers la saisie de son **nom** et **prénom** séparés par **espace**.

Affichons par la suite un message de bienvenu à l'utilisateur (en affichant uniquement son prénom).

Nom&prénom du joueur :fouleni **Foulen**  
 Bonjour **Foulen**

Point d'information

Pour une chaîne de caractère ch="Bonjour" ch[0]="B"

Algorithme	Python
Long(ch)=7	len(ch)
Pos("j",ch)=3      Pos("x",ch)=-1	ch.find('j')
Sous_chaine(ch,3,long(ch))="jour"	Ch[3:len(ch)] ch[3:]

Algorithme	python						
Algorithme multiplication Début Ecrire("nom et prénom=") lire(ch) p←pos(" ",ch) prenom=sous_chaine(ch,p+1,long(ch)) Ecrire("Bonjour", prenom) Score=0 Pour i de 1 à 5 faire Ecrire("Essai",i) x←Aléa(0,10) y←Aléa(0,10) Ecrire(x, " * ", y, " = "), Lire(n) Si n=x*y alors Ecrire("Bravo", prenom) Score←score+1 Sinon Ecrire("Erreur") Score←score-1 FinSi Ecrire("Score=",score) FinPour Ecrire("Score final=",score) Fin Tableau de déclaration des objets	<pre> from random import randint print('Table de multiplication') print('-----') ***** Identification ***** ch=input('nom et prénom(séparés par un espace)=') p=ch.find(' ') #position de l'espace prenom=ch[p+1:len(ch)]# ou prenom=ch[p+1:] print('Bonjour:',prenom) ***** commencer le jeu ***** score=0 for i in range(1,6) :     print('Essai : ',i)     x=randint(0,10)     y=randint(0,10)     print(x, ' * ', y, ' = ', end=' ')     n=int(input())     if n==x*y :         print('Bravo',prenom)         score=score+1     else :         print('Erreur')         score=score-1     print('score= ',score) print('*****') print('score final= ',score)                     </pre>						
<table border="1"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>Ch,prenom</td> <td>chaîne</td> </tr> <tr> <td>Score,p,i, x,y,n</td> <td>Entier</td> </tr> </tbody> </table>	Objets	Types/Nature	Ch,prenom	chaîne	Score,p,i, x,y,n	Entier	
Objets	Types/Nature						
Ch,prenom	chaîne						
Score,p,i, x,y,n	Entier						

**Jeu Chilaformi**

"Pair-Impair" est une version plus simple de jeu « mourore » appelé en Tunisie Chilaformi

- 1- L'ordinateur saisit aléatoirement un nombre x compris entre 0 et 5.
- 2- L'utilisateur saisit un nombre y compris entre 0 et 5.
- 3- Calculer la somme de x et y.
- 4- Si la somme est paire, alors afficher « Ordinateur gagnant ». Si la somme est impaire, alors afficher « Utilisateur gagnant ».

Attribuer un score à l'ordinateur (score1) et un score à l'utilisateur (score2) initialisés par 0, incrémentés à chaque partie gagnée.

Répéter les étapes 1-4 , 5 fois et afficher à la fin le score final : score utilisateur = .. score ordinateur =...

Exemple d'exécution:

Ordinateur: j'ai choisi un nombre entre 0 et 5= \*  
 Utilisateur :Saisir un nombre entre 0 et 5=2  
 Chilaformi  
 La somme de ordinateur( 3) et utilisateur (2)=5 est impaire  
 Utilisateur gagnant



Algorithme Chilaformi

Début

Score1 ← 0

Score2 ← 0

Pour i de 0 à 4 faire

X ← Aléa(0,5)

Ecrire("Ordinateur: J'ai choisi un nombre entre 0 et 5=")

Ecrire("Utilisateur: J'ai choisi un nombre entre 0 et 5=")

Lire(y)

S ← x + y

Ecrire("Chilaformi")

Si S mod 2=0 alors

Ecrire("La somme de ordinateur(",x,") et l'utilisateur(",y,")=",s," est paire")

Ecrire("Ordinateur gagnant")

Score1 ← Score1+1

Sinon

Ecrire("Utilisateur gagnant")

Ecrire("La somme de ordinateur(",x,") et l'utilisateur(",y,")=",s," est impaire")

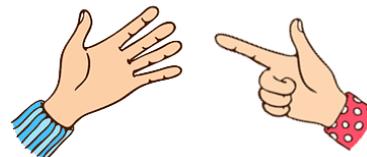
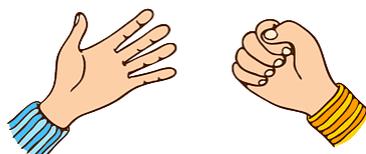
Score2 ← score2+1

FinSi

Finpour

Ecrire("score ordinateur =",score1, "Score utilisateur=",score2)

Fin



**Point d'information**

Division entière	
Algorithme	Python
a div b	a // b
A mod b	a % b

7	2	
	3	Quotient
7 mod 2=1		Reste 1
		7 div 2=3

Objets	Types/Nature
x,y,s,score1,score2, i	entier

<https://archeostudia.com/2020/04/14/idees-archeo-jeu-antique-pair-impair/>

Implémenter l'algorithme en Python et l'envoyer sur Google Classroom



Google Classroom

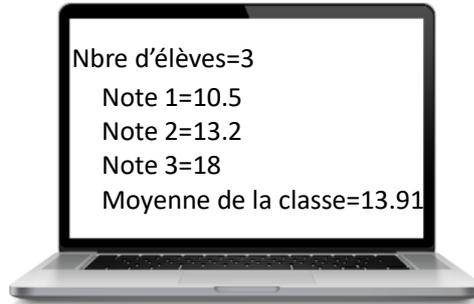
# Moyennes



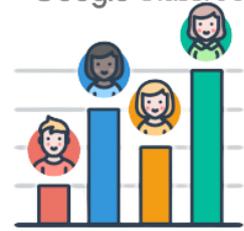
**Projet 3: On veut faire la moyenne des notes d'une classe.**



Exemple d'exécution :



Google Classroom



**Objectif 1 :** Saisir le nombre d'élèves, puis la note de chaque élève, enfin afficher la moyenne de la classe.

**Solution :**

Algorithme	
<b>SOLUTION</b>	Algorithme Moyenne
	<b>Début</b>
	Ecrire("Nbre d'élèves="), Lire(n)
	S ← 0
	Pour i de 0 à n-1 faire
	Ecrire("Note ", i+1, "="), Lire(x)
	S ← S+x
	Finpour
	Moy ← S/n
	Ecrire("moy classe=", Moy)
<b>Fin</b>	



Déclaration des objets

Objets	Types/Nature
i, n	Entier
x, S, moy	réel



**Objectif 2 :** Ajouter une appréciation :  
 « Bien » si la moyenne  $\geq 12$  ,  
 « Passable » si  $10 \leq \text{moyenne} \leq 12$   
 « faible » si moyenne  $< 10$ .

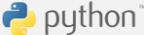
Algorithme	python
si moy $\geq 12$ alors	if moy $\geq 12$ :
Ecrire("Bien")	print('Bien')
Sinon si moy $\geq 10$ :	elif moy $\geq 10$ :
Ecrire("Passable")	print('passable')
Sinon	else :
Ecrire("Faible")	print('Faible')
FinSi	

Implémenter l'algorithme en Python et l'envoyer sur Google Classroom



Ajouter un contrôle de saisie sur les notes pour être entre 0 et 20 et sur le nombre des élèves pour être entre 2 et 25.

On va avoir la solution suivante :

Algorithmme	 python™										
<p>Algorithmme classe</p> <p><b>Début</b></p> <p><b>Répéter</b> Ecrire("Nbre d'élèves=") Lire(n) <b>Jusqu'à</b> <math>2 \leq n \leq 25</math> <math>S \leftarrow 0</math> Pour i de 0 à n-1 faire <b>Répéter</b> Ecrire("Note ", i+1, "=") Lire(x) <b>Jusqu'à</b> <math>0 \leq x \leq 20</math> <math>S \leftarrow S+x</math></p> <p>Finpour Moy <math>\leftarrow S/n</math></p> <p>Ecrire("moy classe=", Moy ) si Moy <math>\geq 12</math> alors Ecrire("Bien") Sinon si Moy <math>\geq 10</math> alors Ecrire("Passable") Sinon Ecrire("Faible") FinSi</p> <p><b>Fin</b> Déclaration des objets</p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>i, n</td> <td>Entier</td> </tr> <tr> <td>S, moy, x</td> <td>réel</td> </tr> </tbody> </table> <p><i>Point d'information</i></p> <div style="border: 2px solid blue; border-radius: 15px; padding: 10px; display: inline-block;"> <p><b>i</b> La boucle <b>Répéter jusqu'à</b></p> <table border="1" style="width: 100%;"> <thead> <tr> <th>Algorithmme</th> </tr> </thead> <tbody> <tr> <td><b>Répéter</b></td> </tr> <tr> <td>Traitement</td> </tr> <tr> <td><b>Jusqu'à</b> condition(s)</td> </tr> </tbody> </table> </div>	Objets	Types/Nature	i, n	Entier	S, moy, x	réel	Algorithmme	<b>Répéter</b>	Traitement	<b>Jusqu'à</b> condition(s)	<pre>n=0 while n not in range(2,26):     n=int(input('Nombres élèves='))  s=0 for i in range(n):     x=-1     while not(0&lt;=x&lt;=20) :         x=float(input('Note'+str( i+1)+ '='))     s=s+x moy=s/n print('moyenne de la classe=',moy) if moy&gt;=12 :     print('Bien') elif moy&gt;=10 :     print('passable') else :     print('Faible')</pre>
Objets	Types/Nature										
i, n	Entier										
S, moy, x	réel										
Algorithmme											
<b>Répéter</b>											
Traitement											
<b>Jusqu'à</b> condition(s)											

# jeu de devinette



## Projet 1:

Un jeu de devinette consiste à générer un nombre **x**(Aléatoire entre 1 et 100) puis essayer de le deviner .  
(Ajouter perdu, si le nombre n'est pas trouvé dans 10 essais)



**Objectif 1 :** Générer un entier *x* au hasard entre 1 et 100.

Point d'information

**Aléa** Permet de générer un entier au hasard.

Algorithme	Python
$x \leftarrow \text{Aléa}(0,10)$	from random import * x=randint(0,10)

Remarque : Pour utiliser **randint** en Python, il faut commencer par importer la bibliothèque **random**

Point d'information

Pour affecter une valeur à une variable, on utilise le symbole d'affectation.

Algorithme	Python
$x \leftarrow 10$	x=10

Attention : ne pas confondre avec le test d'égalité

Algorithme	Python
Si x=10 alors	if x==10 :

SOLUTION

On va avoir la solution suivante :



Algorithme	python™
Algorithme devinette Début $x \leftarrow \text{Aléa}(1,100)$	from random import * #ou from random import randint x=randint(1,100)



**Objectif 2 :** Afficher un message contenant la question « Donner un entier entre 1 et 100 = ? » et demander de l'utilisateur une réponse.

Point d'information

Pour afficher une variable ou un message :

Algorithme	Python
Ecrire("Message",x)	print('Message', x)

- En Python on peut délimiter un texte avec ' ' ou " "
- Exemple : print("C'est un texte")
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : print('\_\_\_',end=' ')

Point d'information

Pour saisir une variable :

Algorithme	Python
lire(n)	n=input()

- En Python : saisir une chaîne (str) : n=input()  
saisir un entier (int): n=int(input())  
saisir un réel (float) : n=float(input())
- Pour lire une variable en affichant un message:  
n=input('Message=')



On va avoir la solution suivante :



Algorithme	python™
Algorithme devinette Début $x \leftarrow \text{Aléa}(1,100)$ Ecrire("Donner un entier entre 1 et 100:"),Lire(n)	from random import randint x=randint(1,100) n=int(input('Donner un entier (entre 1-100) = '))

**Objectif 3 :** On veut afficher un message suivant la valeur entrée : « Bravo » si la réponse est juste et « plus grand » ou « plus petit » sinon .

Algorithme	python™
Algorithme multiplication Début x ← Aléa(1,100) Ecrire("Donner un entier entre 1 et 100:") Lire(y) Si x=y alors Ecrire("Bravo") Sinon si y>x alors Ecrire("Plus petit") Sinon Ecrire("Plus grand") Fin	from random import * x=randint(1,100) y=int(input('Donner un entier (entre 1-100) = ')) if y==x : print('Bravo') elif y>x: print('plus petit !') else: print('Plus grand :')

**Structure conditionnelle si**

Algorithme	Python
<b>Si</b> condition <b>alors</b> Traitement1 <b>Sinon</b> Traitement 2 <b>FinSi</b>	<b>if</b> condition : ↔ traitement1 <b>else</b> : ↔ traitement2

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- Pour rester sur la même ligne et empêcher un retour à la ligne suivante avec print, on pourra utiliser l'instruction suivante : print('\_\_\_',end=' ')

**Objectif 4 :** On veut **répéter** la saisie d'un nombre et l'affichage du message correspondant **jusqu'à** ce que l'utilisateur trouve la solution (x=n) .

Il y a un traitement qui se répète, quelle est la boucle à choisir ?

**La boucle Pour**

Algorithme	Python
Pour i de 0 à n-1 faire Traitement FinPour	for i in range(n) : ↔ Traitement

- En Python on doit utiliser la tabulation ↔ pour exprimer un bloc d'instructions
- range(4) donne 0 1 2 3    range(1,4) donne 1 2 3

**La boucle Répéter jusqu'à**

Algorithme	Python
<b>Répéter</b> Traitement <b>Jusqu'à</b> condition(s)	valide=False <b>while</b> valide==False: Traitement valide=(cond(s))

**La boucle Tant que faire**

Algorithme	Python
<b>Tant que</b> Cond(s) <b>Faire</b> Traitement <b>Fin Tantque</b>	<b>while</b> condition : Traitement

On va avoir la solution suivante :

Algorithme	python™
Algorithme nombre <b>Début</b> X ← Aléa(1,100) <b>Répéter</b> Ecrire("Donner un entiere entre 1 et 100:"),Lire(y) Si x=y alors Ecrire("Bravo") Sinon si y>x alors Ecrire("Plus petit") Sinon Ecrire("Plus grand") Finsi <b>Jusqu'à</b> (x=y) <b>Fin</b>	from random import * x=randint(1,100) valide=False <b>while</b> valide==False: y=int(input('Donner un entier (entre 1-100) = ')) if y==x : print('Bravo') elif y>x: print('plus petit !') else: print('Plus grand :') <b>valide= (x==y)</b>

On pourra aussi implémenter l'algorithme en utilisant While :

```
python™
from random import *
x=randint(1,100)
y=-1
while x !=y:
    y=int(input('Donner un entier (entre 1-100) = '))
    if y==x :
        print('Bravo')
    elif y>x:
        print('plus petit !')
    else:
        print('Plus grand :')
```



**Objectif 5 :** Ajouter le nombre d'essai nb, et le jeu se termine si le joueur trouve le nombre caché et un message « Bravo » est affiché ou si le joueur fait 10 essais sans trouver le nombre et un message « Perdu ! » et affiché, ainsi que le nombre à trouver.

On va avoir la solution suivante :



Algorithme	python™				
<p>Algorithme nombre</p> <p><b>Début</b>  <math>X \leftarrow \text{Aléa}(1,100)</math>  <math>Nb \leftarrow 0</math></p> <p><b>Répéter</b>  <math>Nb \leftarrow nb+1</math>                      Ecrire("Donner un entiere entre 1 et 100:"),Lire(y)                      Si <math>x=y</math> alors Ecrire("Bravo")                      Sinon si <math>y&gt;x</math> alors Ecrire("Plus petit")                                          Sinon Ecrire("Plus grand")                      Finsi</p> <p><b>Jusqu'à</b> (<math>x=y</math>) ou (<math>nb=10</math>)                      Si <math>x \neq y</math> alors Ecrire("Perdu !, le nombre à trouver=", x)                      Finsi</p> <p><b>Fin</b> Déclaration des objets</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>x ,y, nb</td> <td>entier</td> </tr> </tbody> </table>	Objets	Types/Nature	x ,y, nb	entier	<pre> from random import * x=randint(1,100) nb=0 valide=False while valide==False:     nb=nb+1     y=int(input("Donner un entier (entre 1-100) = "))     if y==x :         print('Bravo')     elif y&gt;x:         print('plus petit !')     else:         print('Plus grand :')     valide= (x==y)or (nb==10)  if x!=y:     print('perdu !, le nombre à trouver=',x)                     </pre>
Objets	Types/Nature				
x ,y, nb	entier				

**Exercice 2:**

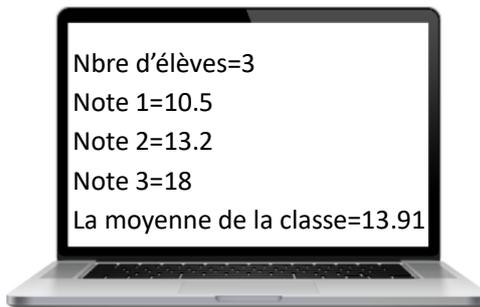
<p>Soit l'algorithme permettant de compter le nombre des voyelles dans une chaîne CH.</p> <p>Algorithme Lettres</p> <p><b>Début</b>                      Ecrire("Donner ch="),lire(ch)  <math>Nb \leftarrow 0</math>                      Pour i de 0 à long(ch)-1 faire                      Si Majus(ch[i]) ∈ {"A","E","I","O","U","Y"} alors  <math>nb \leftarrow nb+1</math>                      Finsi                      FinPour                      Ecrire('Le nombre de voyelles=',nb)</p> <p><b>Fin</b></p> <p style="text-align: center;">Déclaration des objets</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>Ch</td> <td>chaîne</td> </tr> <tr> <td>i, nb</td> <td>entier</td> </tr> </tbody> </table> <p><u>Donner une implimentation en Python:</u>                      ch=input('Donner ch=')                      nb=0                      for i in range(len(ch)):                          if ch[i].upper() in {'A','E','I','O','U','Y'}:                              nb=nb+1                      print('Le nombre de voyelles=',nb)</p>	Objets	Types/Nature	Ch	chaîne	i, nb	entier	<p><u>Ajouter le cas de consonnes :</u></p> <p>Algorithme Lettres2</p> <p><b>Début</b>                      Ecrire("Donner ch="),lire(ch)  <math>Nb \leftarrow 0</math>  <math>nbc \leftarrow 0</math>                      Pour i de 0 à long(ch)-1 faire                      Si Majus(ch[i]) ∈ {"A","E","I","O","U","Y"} alors <math>nb \leftarrow nb+1</math>                          Sinon si Majus(ch[i]) ∈ {"A".."Z"} alors <math>nbc \leftarrow nbc+1</math>                      Finsi                      FinPour                      Ecrire('Le nombre de voyelles=',nb)                      Ecrire('Le nombre de consonnes=',nbc)</p> <p><b>Fin</b></p> <p style="text-align: center;">Déclaration des objets</p> <table border="1" style="margin-left: 40px;"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>Ch</td> <td>chaîne</td> </tr> <tr> <td>i, nb,nbc</td> <td>entier</td> </tr> </tbody> </table> <p><u>en Python:</u>                      ch=input('Donner ch=')                      nb=0                      nbc=0                      for i in range(len(ch)):                          if ch[i].upper() in {'A','E','I','O','U','Y'}:                              nb=nb+1                          elif "A"&lt;=ch[i].upper()&lt;="Z":                              nbc=nbc+1                      print('Le nombre de voyelles=',nb)                      print('Le nombre de consonnes=',nbc)</p>	Objets	Types/Nature	Ch	chaîne	i, nb,nbc	entier
Objets	Types/Nature												
Ch	chaîne												
i, nb	entier												
Objets	Types/Nature												
Ch	chaîne												
i, nb,nbc	entier												

# Moyennes

## Projet 3:

On veut faire la moyenne des notes d'une classe.

Exemple d'exécution :



**Objectif 1** : Saisir le nombre d'élèves, puis la note de chaque élève, enfin afficher la moyenne de la classe.

### 1<sup>ère</sup> solution :

Algorithme	
Algorithme Moyenne	
<b>Début</b>	
Ecrire("Nbre d'élèves="), Lire(n)	
S ← 0	
Pour i de 0 à n-1 faire	
Ecrire("Note ", i+1, "="), Lire(x)	
S ← S+x	
Finpour	
Moy ← S/n	
Ecrire("moy classe=", Moy)	
<b>Fin</b>	

Déclaration des objets

Objets	Types/Nature
i, n	Entier
x, S, moy	réel

### 2<sup>ème</sup> solution : (en utilisant les tableaux)

On va avoir la solution suivante :

Algorithme		python
Algorithme classe		
<b>Début</b>		
Ecrire("Nbre d'élèves=")		from <b>numpy</b> import *
Lire(n)		n=int(input('Nombres élèves='))
Pour i de 0 à n-1 faire		T=array([float()]*n)
Ecrire("Note ", i+1, "=")		for i in range(n):
Lire(T[i])		print('Note', i+1, '=')
Finpour		T[i]=float(input())
S ← 0		s=0
Pour i de 0 à n-1 faire		for i in range(n):
S ← S+T[i]		s=s+T[i]
FinPour		moy=s/n
Moy ← S/n		print(' moy de la classe=', moy)
Ecrire("moy classe=", Moy)		
<b>Fin</b>		
Déclaration des objets		
Objets	Types/Nature	
i, n	Entier	
T	Tableau de n réels	
S, moy	réel	



On utilisera la bibliothèque numpy pour implémenter les tableaux  
Un tableau numpy est :

- Homogène : constitué d'éléments de même type,
- Statique : sa taille est fixée lors de la création

```

Importation
from numpy import *
ou
from numpy import array
ou
import numpy as alias
    
```

```

Déclaration
T = array ([Type_élément] * N)
ou bien
T = array ([valeur_initiale] * N)
    
```

Exemple :  
from numpy import array  
T=array([float()]\*10)



**Objectif 2 :** Ajouter le calcul puis l'affichage de la meilleure note.

On va avoir la solution suivante :



Algorithme		python™								
Algorithme Classe <b>Début</b> Ecrire("Nbre d'élèves="), Lire(n) Pour i de 0 à n-1 faire Ecrire("Note ",i+1,"="), Lire(T[i]) Finpour S←0 Pour i de 0 à n-1 faire S←S+T[i] FinPour Moy←S/n M←T[0] Pour i de 1 à n-1 faire Si T[i]>M alors M←T[i] Finsi FinPour Ecrire("moy classe=",Moy, "Meilleure note=",M) Fin Déclaration des objets		<pre> from numpy import * n=int(input('Nombres élèves=')) T=array([float()]*n) for i in range(n):     print("Nbre d'élèves=")     T[i]=float(input()) s=0 for i in range(n):     s=s+T[i] moy=s/n M=T[0] for i in range(1,n) :     if T[i]&gt;M :         M=T[i] print(' moy de la classe=',moy,'\nMeilleure note=',M)                     </pre>								
<table border="1"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>i,n</td> <td>Entier</td> </tr> <tr> <td>T</td> <td>Tableau de n réels</td> </tr> <tr> <td>S, moy, M</td> <td>réel</td> </tr> </tbody> </table>	Objets	Types/Nature	i,n	Entier	T	Tableau de n réels	S, moy, M	réel	<p><b>Point d'information</b></p> <p>Dans un print "\n" permet d'ajouter un retour à la ligne.</p>	
Objets	Types/Nature									
i,n	Entier									
T	Tableau de n réels									
S, moy, M	réel									



**Objectif 3 :** Ajouter un contrôle de saisie sur les notes pour être entre 0 et 20 et sur le nombre des élèves pour être entre 2 et 25.

On va avoir la solution suivante :



Algorithme		python™												
Algorithme classe <b>Début</b> Répéter Ecrire("Nbre d'élèves="), Lire(n) Jusqu'à 2<=n<=25 Pour i de 0 à n-1 faire Répéter Ecrire("Note ",i+1,"="), Lire(T[i]) Jusqu'à 0<=T[i]<=20 Finpour S←0 Pour i de 0 à n-1 faire S←S+T[i] FinPour Moy←S/n M←T[0] Pour i de 1 à n-1 faire Si T[i]>M alors M←T[i] Finsi FinPour Ecrire("moy classe=",Moy, "Meilleure note=",M) Fin Déclaration des objets		<pre> from numpy import * T=array([float()]*25) valide=False while valide==False :     n=int(input('Nombres élèves='))     valide=(2&lt;=n&lt;=25) for i in range(n):     valide=False     while valide==False :         print("Note", i+1, '=',end=' ')         T[i]=float(input())         valide=(0&lt;=T[i]&lt;=20) s=0 for i in range(n):     s=s+T[i] moy=s/n M=T[0] for i in range(1,n) :     if T[i]&gt;M :         M=T[i] print(' moy de la classe=',moy,'\nMeilleure note=',M)                     </pre>												
<table border="1"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>i,n</td> <td>Entier</td> </tr> <tr> <td>T</td> <td>Tableau de n réels</td> </tr> <tr> <td>S, moy, M</td> <td>réel</td> </tr> </tbody> </table>	Objets	Types/Nature	i,n	Entier	T	Tableau de n réels	S, moy, M	réel	<p><b>Point d'information</b></p> <p>La boucle Répéter jusqu'à</p> <table border="1"> <thead> <tr> <th>Algorithme</th> </tr> </thead> <tbody> <tr> <td>Répéter Traitement Jusqu'à condition(s)</td> </tr> </tbody> </table> <p><b>Point d'information</b></p> <p>La boucle Répéter jusqu'à</p> <table border="1"> <thead> <tr> <th>Python</th> </tr> </thead> <tbody> <tr> <td>valide=False while valide==False:     Traitement     valide=(cond(s))</td> </tr> </tbody> </table>		Algorithme	Répéter Traitement Jusqu'à condition(s)	Python	valide=False while valide==False: Traitement valide=(cond(s))
Objets	Types/Nature													
i,n	Entier													
T	Tableau de n réels													
S, moy, M	réel													
Algorithme														
Répéter Traitement Jusqu'à condition(s)														
Python														
valide=False while valide==False: Traitement valide=(cond(s))														



**Objectif 4** : Décomposer le problème en modules.

**Point d'information**

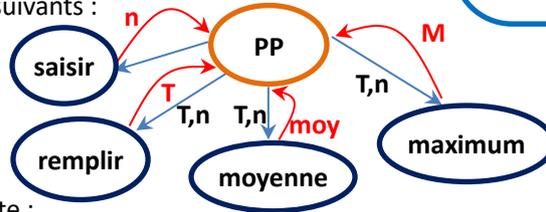
La **décomposition modulaire** consiste à diviser un problème en sous problème de difficultés moindres. Ces derniers sont aussi soumis à cette division jusqu'à ce qu'on arrive à un niveau abordable de difficulté.

**Point d'information**

**Intérêts de la décomposition modulaire :**

- Plus d'organisation en séparant les difficultés et les tâches.
- S'occuper d'un seul problème à la fois.
- En cas d'erreur la division en module permet de savoir quel module à corriger
- Plus facile à faire évoluer.
- Permet d'éviter la répétition d'un même traitement dans un programme.

On peut utiliser les modules suivants :



**Point d'information**

La décomposition modulaire permet de diviser un programme en plusieurs sous-programmes. (procédures ou une fonctions).

On va avoir la solution suivante :

**Algorithmes**

**Algorithme du programme principal :**

```

Algorithme Classe:
DEBUT
    Saisir(n)
    Remplir(T1, n)
    Moy ← moyenne(T1,n)
    M ← maximum(T1,n)
    Ecrire("moy classe=", Moy, "Meilleure note=", M)
FIN
    
```

Appel d'une fonction

Appel d'une procédure

Paramètres effectifs



Nouveaux Types	
TAB= tableau de 25 réels	
T.D.O.globaux	
Objet	Type/Nature
n	Entier
moy, M	réel
T1	TAB
Saisir, remplir	Procédures
Moyenne, maximum	fonctions

**Algorithme de la fonction moyenne :**

```

fonction moyenne(T:tab,n :entier) :réel
Début
    S ← 0
    Pour i de 0 à n-1 faire
        S ← S+T[i]
    FinPour
    Moy ← S/n
    Retourner Moy
Fin
    
```

Type de retour de la fonction

T.D.O.Locaux

Objets	Types/Nature
i	Entier
s, moy	réel

**Algorithme de la fonction maximum :**

```

fonction maximum(T:tab,n :entier) :réel
Début
    X ← T[0]
    Pour i de 1 à n-1 faire
        Si T[i]>M alors X ← T[i]    Finsi
    FinPour
    Retourner X
Fin
    
```

Paramètres formels

T.D.O.Locaux

Objets	Types/Nature
i	Entier
X	réel

**Algorithme de la procédure saisir :**

```

Procédure saisir(@n :entier)
DEBUT
    Répéter
        Ecrire("Nbre d'élèves="), lire(n)
    Jusqu'à 2<=n<=25
FIN
    
```

**Algorithme de la procédure remplir :**

```

Procédure Remplir(@T:tab,n :entier)
Début
    Pour i de 0 à n-1 faire
        Répéter
            Ecrire("Note ",i+1,"="), Lire(T[i])
        Jusqu'à 0<=T[i]<=20
    Finpour
Fin
    
```

T.D.O.Locaux

Objets	Types/Nature
i	Entier

**Point d'information**

On distingue deux types de paramètres :

- 1-**Les paramètres formels** : qui figurent dans la définition de la procédure ou la fonction.
- 2-**Les paramètres effectifs** : qui figures dans l'appel de la procédure ou la fonction et qui sont manipulés par celle-ci.

**Point d'information**

Il y a deux modes de passage par (**valeur/adresse**). Si le mode de passage est par référence (par adresse), on ajoutera le symbole **@** avant le nom du paramètre.

**Point d'information**

Remarque : Les paramètres **formels** et les paramètres **effectifs** doivent s'accorder de point de vue nombre et ordre et leurs types doivent être identiques ou compatibles.

**Point d'information**

En Algorithmique :

Les **procédures** sont des sous-programmes qui peuvent avoir plusieurs résultats.

Une **fonction** est un sous-programme qui renvoie une valeur de type simple, ce type sera le type de la fonction..

**Implémentations en python**

 python <sup>™</sup> Méthode 1	 python <sup>™</sup> Méthode 2
<pre>from numpy import * T1=array([float()]*25)  def saisir():     valide=False     while valide==False :         n=int(input('Nombres élèves='))         valide=(2&lt;=n&lt;=25)     return n  def remplir(T,n):     for i in range(n):         valide=False         while valide==False :             print("Note", i+1, '=', end=' ')             T[i]=float(input())             valide=(0&lt;=T[i]&lt;=20)  def moyenne(T,n):     s=0     for i in range(n):         s=s+T[i]     moy=s/n     return moy  def maximum(T,n):     X=T[0]     for i in range(1,n) :         if T[i]&gt;X :             X=T[i]     return X  #programme principal n=saisir() remplir(T1,n) moy=moyenne(T1,n) M=maximum(T1,n) print('moy de la classe=',moy,'\nMeilleure note=',M)</pre>	<pre>from numpy import * T1=array([float()]*25)  def saisir():     global n     valide=False     while valide==False :         n=int(input('Nombres élèves='))         valide=(2&lt;=n&lt;=25)  def remplir(T,n):     for i in range(n):         valide=False         while valide==False :             print("Note", i+1, '=', end=' ')             T[i]=float(input())             valide=(0&lt;=T[i]&lt;=20)  def moyenne(T,n):     s=0     for i in range(n):         s=s+T[i]     moy=s/n     return moy  def maximum(T,n):     X=T[0]     for i in range(1,n) :         if T[i]&gt;X :             X=T[i]     return X  #programme principal saisir() remplir(T1,n) moy=moyenne(T1,n) M=maximum(T1,n) print('moy de la classe=',moy,'\nMeilleure note=',M)</pre>

**Point d'information**

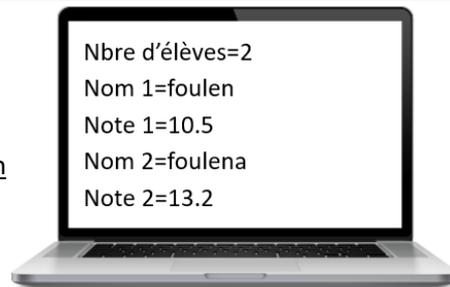
En Python : Un module (fonction ou procédure) se définit en utilisant le mot clé **def**

**Point d'information****La portée des variables en python :**

- Toute variable déclarée au sein d'un module a une portée locale.
- Toute variable déclarée au sein d'un module précédée par le mot clé **global** a une portée globale. Par conséquent, elle ne devra pas figurer parmi les paramètres de ce module.
- Il est possible de définir un sous-programme sans paramètres. La communication avec l'appelant se produit grâce aux ressources (objets) communes partagées entre l'appelé et l'appelant.



**Objectif 5** : Modifier le programme principal et le module Remplir pour saisir le nom et chaque élève et sa note.



Point d'information

Déclarer un tableau de caractère :

```
Déclaration
T1 = array ([str( )] * N)
```

Déclarer un tableau de chaînes de caractère :

```
Déclaration
T2 = array ([str] * N)
```

Exemple d'exécution

Algorithme	python™																		
<p><u>Algorithme du programme principal :</u></p> <p>Algorithme Classe:</p> <p>DEBUT</p> <p style="padding-left: 20px;">Saisir(n)</p> <p style="padding-left: 20px;">Remplir(T1,T2 n)</p> <p style="padding-left: 20px;">Moy ← moyenne(T1,n)</p> <p style="padding-left: 20px;">M ← maximum(T1,n)</p> <p style="padding-left: 20px;">Ecrire("moy classe=",Moy, "Meilleure note=",M)</p> <p>FIN</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;">Nouveaux Types</p> <p>TAB= tableau de 25 réels</p> <p>TAB2=Tableau de 25 chaînes</p> </div> <p style="text-align: center;">T.D.O.globaux</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 50%;">Objet</th> <th style="width: 50%;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td>n</td> <td>Entier</td> </tr> <tr> <td>moy, M</td> <td>réel</td> </tr> <tr> <td>T1</td> <td>TAB</td> </tr> <tr> <td><b>T2</b></td> <td><b>TAB2</b></td> </tr> <tr> <td>Saisir, remplir</td> <td>Procédures</td> </tr> <tr> <td>Moyenne,maximum</td> <td>fonctions</td> </tr> </tbody> </table> <p><u>Algorithme de la procédure remplir :</u></p> <p><b>Procédure Remplir</b>(@T:tab,@T2:tab2,n :entier)</p> <p>Début</p> <p style="padding-left: 20px;">Pour i de 0 à n-1 faire</p> <p style="padding-left: 40px;">Ecrire("Nom ",i+1,"="), Lire(T2[i])</p> <p style="padding-left: 20px;">Répéter</p> <p style="padding-left: 40px;">Ecrire("Note ",i+1,"="), Lire(T[i])</p> <p style="padding-left: 20px;">Jusqu'à 0&lt;=T[i]&lt;=20</p> <p>Finpour</p> <p>Fin</p> <p>T.D.O.Locaux</p> <table border="1" style="width: 100%; border-collapse: collapse; margin: 10px 0;"> <thead> <tr> <th style="width: 50%;">Objets</th> <th style="width: 50%;">Types/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> </tbody> </table>	Objet	Type/Nature	n	Entier	moy, M	réel	T1	TAB	<b>T2</b>	<b>TAB2</b>	Saisir, remplir	Procédures	Moyenne,maximum	fonctions	Objets	Types/Nature	i	Entier	<pre> from numpy import * T1=array([float()*25) T2=array([str]*25) def saisir():     valide=False     while valide==False :         n=int(input('Nombres élèves='))         valide=(2&lt;=n&lt;=25)     return n  def remplir(T,T2,n):     for i in range(n):         print("Nom:", i+1, '=', end=' ')         T2[i]=input()         valide=False         while valide==False :             print("Note", i+1, '=', end=' ')             T[i]=float(input())             valide=(0&lt;=T[i]&lt;=20)  def moyenne(T,n):     s=0     for i in range(n):         s=s+T[i]     moy=s/n     return moy  def maximum(T,n):     X=T[0]     for i in range(1,n) :         if T[i]&gt;X :             X=T[i]     return X  #programme principal n=saisir() remplir(T1,T2,n) moy=moyenne(T1,n) M=maximum(T1,n) print('moy de la classe=',moy,'\nMeilleure note=',M)                     </pre>
Objet	Type/Nature																		
n	Entier																		
moy, M	réel																		
T1	TAB																		
<b>T2</b>	<b>TAB2</b>																		
Saisir, remplir	Procédures																		
Moyenne,maximum	fonctions																		
Objets	Types/Nature																		
i	Entier																		



**Objectif 6 : Modifier le programme principal et ajouter un module Afficher pour afficher les élèves et leurs notes**



Algorithme	python														
<p><u>Algorithme du programme principal :</u>                      Algorithme Classe:                      DEBUT                          Saisir(n)                          Remplir(T1,T2, n)                          Moy ← moyenne(T1,n)                          M ← maximum(T1,n)                          <b>Afficher(T1,T2,n)</b>                          Ecrire("moy classe=",Moy, "Meilleure note=",M)                      FIN</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p style="text-align: center;">Nouveaux Types</p> <p>TAB= tableau de 25 réels                      TAB2=Tableau de 25 chaînes</p> </div> <p style="text-align: center;">T.D.O.globaux</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Objet</th> <th style="width: 50%;">Type/Nature</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">n</td> <td style="text-align: center;">Entier</td> </tr> <tr> <td style="text-align: center;">moy, M</td> <td style="text-align: center;">réel</td> </tr> <tr> <td style="text-align: center;">T1</td> <td style="text-align: center;">TAB</td> </tr> <tr> <td style="text-align: center;"><b>T2</b></td> <td style="text-align: center;"><b>TAB2</b></td> </tr> <tr> <td style="text-align: center;">Saisir, remplir, <b>afficher</b></td> <td style="text-align: center;">Procédures</td> </tr> <tr> <td style="text-align: center;">Moyenne,maximum</td> <td style="text-align: center;">fonctions</td> </tr> </tbody> </table>	Objet	Type/Nature	n	Entier	moy, M	réel	T1	TAB	<b>T2</b>	<b>TAB2</b>	Saisir, remplir, <b>afficher</b>	Procédures	Moyenne,maximum	fonctions	<pre> from numpy import * T1=array([float()]*25) T2=array([str]*25) def saisir():     valide=False     while valide==False :         n=int(input('Nombres élèves='))         valide=(2&lt;=n&lt;=25)     return n def remplir(T,T2,n):     for i in range(n):         print("Nom:", i+1, '=', end=' ')         T2[i]=input()         valide=False         while valide==False :             print("Note", i+1, '=', end=' ')             T[i]=float(input())             valide=(0&lt;=T[i]&lt;=20) def moyenne(T,n):     s=0     for i in range(n):         s=s+T[i]     moy=s/n     return moy def maximum(T,n):     X=T[0]     for i in range(1,n) :         if T[i]&gt;X :             X=T[i]     return X def afficher(T1,T2,n):     for i in range(n):         print(T2[i], ' : ', T1[i])  #programme principal n=saisir() remplir(T1,T2,n) moy=moyenne(T1,n) M=maximum(T1,n) <b>afficher(T1,T2,n)</b> print('moy de la classe=',moy,'\nMeilleure                     note=',M)                 </pre>
Objet	Type/Nature														
n	Entier														
moy, M	réel														
T1	TAB														
<b>T2</b>	<b>TAB2</b>														
Saisir, remplir, <b>afficher</b>	Procédures														
Moyenne,maximum	fonctions														
<p><u>Algorithme de la procédure Afficher :</u>  <b>Procédure Afficher(T1:tab,T2:tab2,n :entier)</b>                      Début                          Pour i de 0 à n-1 faire                              Ecrire(T2[i] , " : ", T1[i] )                          Finpour                      Fin</p> <p>T.D.O.Locaux</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 50%;">Objets</th> <th style="width: 50%;">Types/Nature</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">i</td> <td style="text-align: center;">Entier</td> </tr> </tbody> </table>	Objets	Types/Nature	i	Entier											
Objets	Types/Nature														
i	Entier														



**Objectif 7** : Ajouter un contrôle de saisie sur les noms des élèves pour être composés par des lettres et des espaces seulement.

Algorithme	python														
<p><u>Algorithme du programme principal :</u></p> <p>Algorithme Classe:</p> <p>DEBUT</p> <p>Saisir(n)</p> <p>Remplir(T1,T2, n)</p> <p>Moy ← moyenne(T1,n)</p> <p>M ← maximum(T1,n)</p> <p>Afficher(T1,T2,n)</p> <p>Ecrire("moy classe=",Moy, "Meilleure note=",M)</p> <p>FIN</p>	<pre>from numpy import * T1=array([float()]*25) T2=array([str]*25) def saisir():     valide=False     while valide==False :         n=int(input('Nombres élèves='))         valide=(2&lt;=n&lt;=25)     return n def verif(ch):     ok=True     i=-1     while (ok==True) and (i&lt;len(ch)-1):         i=i+1         if not(('A'&lt;=ch[i].upper()&lt;='Z') or (ch[i]==' ')):             ok=False     return ok def remplir(T,T2,n):     for i in range(n):         valide=False         while valide==False:             print("Nom:", i+1, '=', end=' ')             T2[i]=input()             valide=verif(T2[i])      valide=False     while valide==False :         print("Note", i+1, '=', end=' ')         T[i]=float(input())         valide=(0&lt;=T[i]&lt;=20) def moyenne(T,n):     s=0     for i in range(n):         s=s+T[i]     moy=s/n     return moy def maximum(T,n):     X=T[0]     for i in range(1,n) :         if T[i]&gt;X :             X=T[i]     return X def afficher(T1,T2,n):     for i in range(n):         print(T2[i], ' : ', T1[i]) #programme principal n=saisir() remplir(T1,T2,n) moy=moyenne(T1,n) M=maximum(T1,n) afficher(T1,T2,n) print('moy de la classe=',moy,'\nMeilleure note=',M)</pre>														
<p><b>SOLUTION</b></p> <p>Nouveaux Types</p> <p>TAB= tableau de 25 réels TAB2=Tableau de 25 chaînes</p> <p>T.D.O.globaux</p> <table border="1"> <thead> <tr> <th>Objet</th> <th>Type/Nature</th> </tr> </thead> <tbody> <tr> <td>n</td> <td>Entier</td> </tr> <tr> <td>moy, M</td> <td>réel</td> </tr> <tr> <td>T1</td> <td>TAB</td> </tr> <tr> <td><b>T2</b></td> <td><b>TAB2</b></td> </tr> <tr> <td>Saisir, remplir</td> <td>Procédures</td> </tr> <tr> <td>Moyenne,maximum</td> <td>fonctions</td> </tr> </tbody> </table>	Objet	Type/Nature	n	Entier	moy, M	réel	T1	TAB	<b>T2</b>	<b>TAB2</b>	Saisir, remplir	Procédures	Moyenne,maximum	fonctions	
Objet	Type/Nature														
n	Entier														
moy, M	réel														
T1	TAB														
<b>T2</b>	<b>TAB2</b>														
Saisir, remplir	Procédures														
Moyenne,maximum	fonctions														
<p><u>Algorithme de la fonction verif:</u></p> <p>Fonction verif(ch:chaîne):booléen</p> <p>Début</p> <p>i ← -1, ok ← vrai</p> <p>Répéter</p> <p>i ← i+1</p> <p>si NON(Majus(ch[i]) ∈ ["A".."Z"] ou ch[i]=" ")</p> <p>alors ok ← Faux</p> <p>Finsi</p> <p>Jusqu'a (i=long(ch)-1) ou (ok=faux)</p> <p>Retourner ok</p> <p>Fin</p> <p>T.D.O.Locaux</p> <table border="1"> <thead> <tr> <th>Objet</th> <th>T/N</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> <tr> <td>ok</td> <td>booléen</td> </tr> </tbody> </table>	Objet	T/N	i	Entier	ok	booléen									
Objet	T/N														
i	Entier														
ok	booléen														
<p><u>Algorithme de la procédure remplir :</u></p> <p><b>Procédure Remplir(@T:tab,@T2:tab2,n :entier)</b></p> <p>Début</p> <p>Pour i de 0 à n-1 faire</p> <p><b>Répéter</b></p> <p>Ecrire("Nom ",i+1,"="), Lire(T2[i])</p> <p><b>Jusqu'à verif(T2[i])=vrai</b></p> <p>Répéter</p> <p>Ecrire("Note ",i+1,"="), Lire(T[i])</p> <p>Jusqu'à 0&lt;=T[i]&lt;=20</p> <p>Finpour</p> <p>Fin</p> <p>T.D.O.Locaux</p> <table border="1"> <thead> <tr> <th>Objets</th> <th>Types/Nature</th> </tr> </thead> <tbody> <tr> <td>i</td> <td>Entier</td> </tr> <tr> <td>verif</td> <td>fonction</td> </tr> </tbody> </table>	Objets	Types/Nature	i	Entier	verif	fonction									
Objets	Types/Nature														
i	Entier														
verif	fonction														

**Rq :Afficher le nom de l'élève ayant la meilleure note.**



**Objectif 8 :** Ajouter un contrôle de saisie sur les noms des élèves pour être distincts .



**Objectif 9 :** Ajouter un module Résultat pour afficher le rang , le nom et la moyenne de chaque élève en ordre décroissant suivant la moyenne.

Exemple d'affichage :

