

En algorithme on a deux types de modules :

**Les fonctions :**

Algorithmique	Implémentation en python
Fonction cube (n :entier) Debut S←n*n*n retourner s Fin	<pre>def cube(n):     s=n*n*n     return s</pre>

**Les procédures :**

Algorithmique	Implémentation en python
Procedure afficher (t : Tab,n :entier) Debut Pour i de 0 à n-1 faire Ecrire (t[i]) Fin Pour Fin	<pre>def afficher(t,n):     for i in range(n):         print(t[i])</pre>

**Activité**

Donner le résultat d'exécution des scripts suivant :

A/

Python	Algorithmique	Résultat
<pre>def ajouterUn(a):     a=a+1  a=4 ajouterUn(a) print(a)</pre>	procedure ajouterUn( a :entier): Debut     a←a+1 Fin      a←4     ajouterUn(a)     ecrire(a)	<b>On affiche toujours 4</b>

**Solution 1 :**

Python	Algorithme	Résultat
<pre>def ajouterUn():     global a     a=a+1      a=4     ajouterUn(a)     print(a)</pre>	procedure ajouterUn(@ a :entier): Debut     a←a+1 Fin      a←4     ajouterUn(a)     ecrire(a)	<b>On affiche 5</b>

**B/**

Python	Algorithme	Résultat
<pre>def saisir():     a=int(input())      a=3     saisir(a)     print(a)</pre>	Procedure saisir(a :entier) : Debut     Lire(a) Fin      a←3     saisir(a)     ecrire(a)	<b>On affiche toujours 3</b>

**Solution 2 :**

Python	Algorithme	Résultat
<pre>def saisir():     global a     a=int(input())      a=3     saisir(a)     print(a)</pre>	Procedure saisir( @ a :entier) : Debut     Lire(a) Fin      a←3     saisir(a)     ecrire(a)	<b>On afficher la valeur saisie</b>

Si on modifie un objet global dans un module alors utiliser @

## Exemples d'utilisations:

### A-Module de saisir :

#### Saisir une variable

Procedure saisirN(@ n :entier)

Debut

Repeter

Ecrire ("Taille tableau ")

Lire (n)

Jusqu'à 2<=n<=25)

Fin

#### Remplir un tableau

Procedure remplirT1 (@ t : Tab,n :entier)

Debut

Pour i de 0 à n-1 faire

Lire (t[i])

Fin Pour

Fin

#### B/Modifier une ou plusieurs valeur(s)

Procedure permuter (@ a : entier, @ b :entier)

Debut

X←a

a←b

b←X

Fin

#### C/ Afficher un tableau

Exemple :

Procedure afficher (t : Tab,.n :entier)

Debut

Pour i de 0 à n-1 faire

Ecrire (t[i])

Fin Pour

Fin

```
a←5
b←3
Permuter(a,b)
Ecrire('a=',a,'b=',b)
```

On affiche a=3 b=5

**Exercice 1****Soit le script suivant**M $\leftarrow$ T[0]

Pour i de 1 à 3 faire

Si T[i]&gt;M alors

        M $\leftarrow$ T[i]

Fin Si

Fin Pour

Ecrire(M)

a) Exécuter le code suivant sur le tableau ci contre:

3	7	2	10
---	---	---	----

**On Affiche 10**

b) Donner le rôle de cet algorithme

**Afficher la valeur maximale**

c) Modifier le programme pour afficher l'élément minimum

d) Modifier le programme pour afficher l'indice (position) de l'élément maximum

e) Ecrire un programme qui permet de saisir un tableau de n entiers puis d'afficher la valeur maximale (Utiliser des modules).

f) Modifier le programme pour afficher le nombre des valeurs maximales

**Exercice 2**

Soir l'algorithme de la fonction 'Recherche' suivante :

fonction Recherche (t :tab ;n :entier ;c :entier) :booleen

    Trouve $\leftarrow$ faux

Pour i de 0 à n-1 faire

si t[i]=c alors

            trouve $\leftarrow$ vrai

Finsi

finPour

**Questions**

a)Exécuter le code suivant sur le tableau T ci dessous et c=3 :

6	3	2	13
---	---	---	----

b)remplacer la boucle pour par la boucle répéter