

Algorithmes usuels+ implémentation en Python

Saisir un nombre entre 2 et 15 : Procédure saisir(@n : entier) Début Répéter Ecrire ("Donner un entier") Lire (n) Jusqu'à $2 \leq n \leq 15$ Fin	def saisir () : global n n= int (input ("Donner un entier")) while not (2<=n<=15) : n= int (input ("Donner un entier"))
Saisir une chaine numérique : Procédure saisir(@ch : chaine) Début Répéter Ecrire ("Donner une chaine de caractères") Lire (ch) Jusqu'à Estnum(ch)=vrai Fin	def saisir () : global ch ch= input ("Donner une chaine de caractères ") while not ch.isdecimal() : ch= input ("Donner une chaine de caractères ")
Saisir une chaine alphabétique Procédure saisir(@ch : chaine) Début Répéter Ecrire ("Donner une chaine de caractères") Lire (ch) Jusqu'à Alpha (ch)=vrai Fin Fonction Alpha (ch : chaine) : booléen Début i ← 0 test ← vrai Tantque (i < long (ch)) et (test=vrai) faire test← Majus(ch[i]) ∈ ["A".."Z"] i←i+1 FinTantque Retourner test Fin	def Alpha(ch) : i=0 test=True while (i<len(ch)) and (test ==True): test= "A"<= ch.upper() <= "Z" i=i+1 return test def saisir () : global ch ch= input ("Donner une chaine de caractères ") while not Alpha(ch): ch= input ("Donner une chaine de caractères ")
Remplir un tableau par n valeurs entre 1 et 100 : Procédure Remplir(@T : Tab, n : entier) Début Pour i de 0 a n-1 faire Répéter Ecrire ("Donner l'élément n°", i+1, " :") Lire (T[i]) Jusqu'à $1 \leq T[i] \leq 100$ FinPour Fin	from numpy import array T=array([int()]*n) def Remplir(T, n) : for i in range(0, n) : T[i] = int(input("Donner l'élément n°"+ str(i+1)+ " :")) while not (1<=T[i]<=100) : T[i]=int(input("Donner l'élément n°"+ str(i+1)+ " :"))

Remplir un tableau par n nombres aléatoires entre 1 et 100 : Procédure Remplir(@T : Tab, n : entier) Début Pour i de 0 a n-1 faire T[i]←Aléa(1, 100) FinPour Fin	<pre>from random import randint from numpy import array T=array([int()]*n) def Remplir(T, n) : for i in range(0, n) : T[i] = randint(1, 100)</pre>
Remplir un tableau par n valeurs distincts : Procédure Remplir(@T : Tab, n : entier) Début Pour i de 0 a n-1 faire Répéter Ecrire ("Donner l'élément n°", i+1, " :") Lire (T[i]) Jusqu'à Existe (T[i], T, i) = faux FinPour Fin Fonction Existe (x :entier,T :tab,i :entier) :booléen Début test←faux j←0 Tantque (test=faux) et (j<i) faire Si T[j]=x alors test ←vrai Sinon j←j+1 FinSi FinTantque Retourner test Fin	<pre>from numpy import array T=array([int()]*n) def Existe(x, T, i) : test=False j=0 while (test== False) and (j<i): if T[j]== x: test= True else: j=j+1 return test def Remplir(T, n) : for i in range(0, n) : T[i] = int(input("Donner l'élément n°"+ str(i+1)+ " :")) while not Existe(T[i], T, i) : T[i]=int(input("Donner l'élément n°"+ str(i+1)+ " :"))</pre>
Remplir un tableau dans l'ordre croissant : Procédure Remplir(@T : Tab, n : entier) Début Ecrire ("Donner l'élément n° 1 :") Lire (T[0]) Pour i de1 a n-1 faire Répéter Ecrire ("Donner l'élément n°", i+1, " :") Lire (T[i]) Jusqu'à T[i] > T[i-1] FinPour Fin	<pre>from numpy import array T=array([int()]*n) def Remplir(T, n) : T[0] = int(input("Donner l'élément n° 1 :")) for i in range(1, n) : T[i] = int(input("Donner l'élément n°"+ str(i+1)+ " :")) while not (T[i]>T[i-1]) : T[i]=int(input("Donner l'élément n°"+ str(i+1)+ " :"))</pre>

Remplir un tableau par n chaîne de caractères numériques : Procédure Remplir(@T : Tab, n : entier) Début Pour i de 0 a n-1 faire Répéter Ecrire ("Donner l'élément n°", i+1, " :") Lire (T[i]) Jusqu'à Estnum(T[i]) FinPour Fin	<pre>from numpy import array T=array([str]*n) def Remplir(T, n) : for i in range(0, n) : T[i] = input("Donner l'élément n°"+ str(i+1)+ " :") while not T[i].isdecimal() : T[i] = input("Donner l'élément n°"+ str(i+1)+ " :")</pre>
Remplir un tableau par n chaîne de caractères alphabétiques : Procédure Remplir(@T : Tab, n : entier) Début Pour i de 0 a n-1 faire Répéter Ecrire ("Donner l'élément n°", i+1, " :") Lire (T[i]) Jusqu'à Alpha(T[i]) FinPour Fin Fonction Alpha (ch : chaîne) : booléen Début i ← 0 test ← vrai Tantque (i < long (ch)) et (test=vrai) faire test← Majus(ch[i]) ∈["A".."Z"] i←i+1 FinTantque Retourner test Fin	<pre>from numpy import array T=array([str]*n) def Alpha(ch) : i=0 test=True while (i<len(ch)) and (test ==True): test= "A"<= ch.upper() <= "Z" i=i+1 return test def Remplir(T, n) : for i in range(0, n) : T[i] = input("Donner l'élément n°"+ str(i+1)+ " :") while not Alpha(T[i]): T[i] = input("Donner l'élément n°"+ str(i+1)+ " :")</pre>

Calculer le pgcd de deux entiers : Fonction pgcd (n, m :entier) :entier Début Tantque n≠m faire Si n >m alors n ← n-m Sinon m ← m-n FinSi FinTantque Retourner n Fin	def pgcd (n, m): while n!=m: if n>m: n=n-m else: m=m-n return n
Calculer le ppcm de deux entiers : Fonction ppcm (n, m : entier) :entier Début c←n d←m Tantque n≠m faire Si n < m alors n ← n+c Sinon m ← m+d FinSi FinTantque Retourner n Fin	def ppcm (n, m): c, d=n, m while n!=m: if n<m: n=n+c else: m=m+d return n
Vérifier si un nombre est parfait ou non : Fonction parfait (n: entier) : booléen Début s←0 Pour i de 1 à n div 2 faire Si n mod i = 0 alors s ← s+i FinSi FinPour Retourner n=s Fin	def parfait(n) : s=0 for i in range(1, n//2 +1): if n%i ==0 : s=s+i return n==s
Verifier si un nombre est premier ou non : Fonction premier(n:entier): booléen Début nb←0 Pour i de 1 à n faire Si n mod i = 0 alors nb ← nb+1 FinSi FinPour Retourner nb=2 Fin	def premier(n) : nb=0 for i in range(1, n+1): if n%i ==0 : nb=nb+1 return nb==2