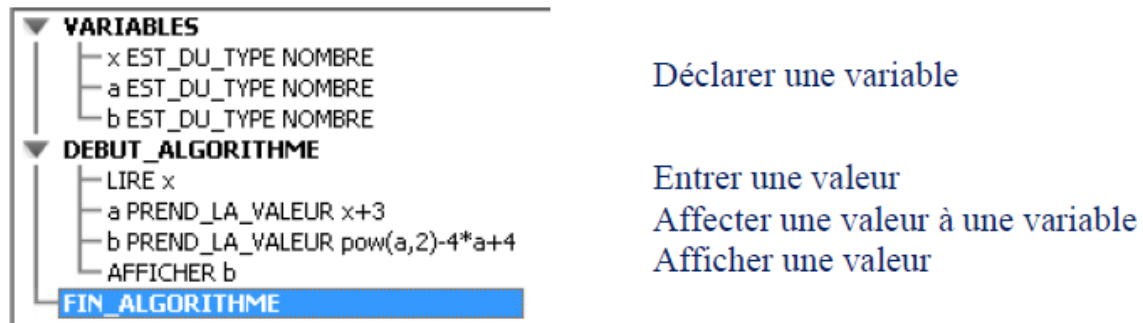


INFORMATIQUE

Dans cette fiche, plusieurs programmes vous sont demandés à faire avec algofox. Vous les réalisez et vous les envoyez par mail à cnamdevoir@gmail.com

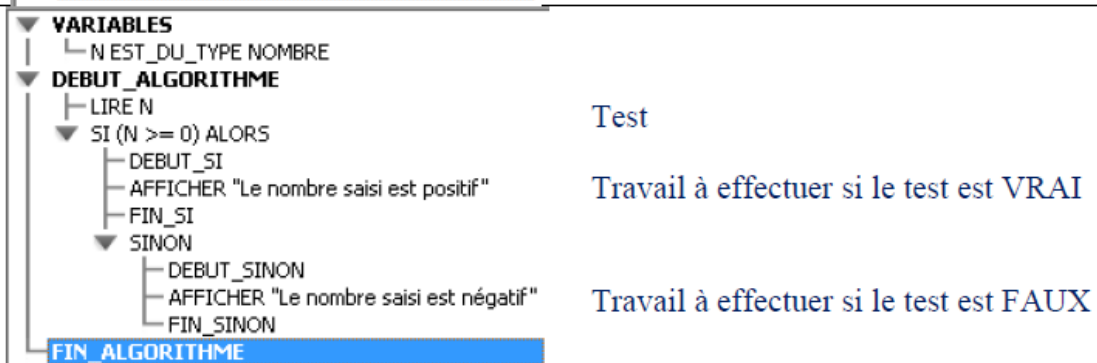
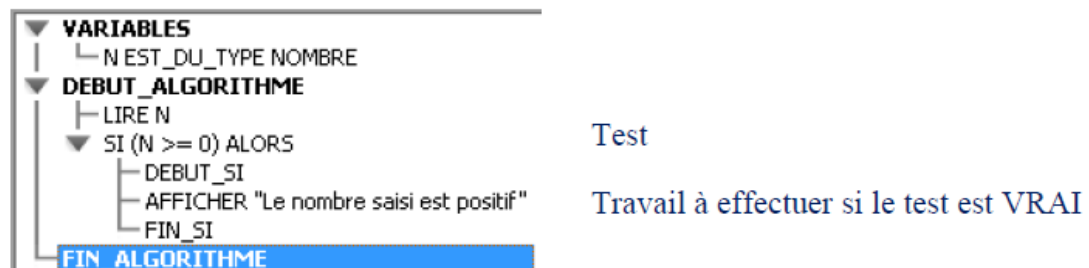
ALGORITHMIQUE AVEC ALGOBOX

Instructions élémentaires



INSTRUCTIONS CONDITIONNELLES :

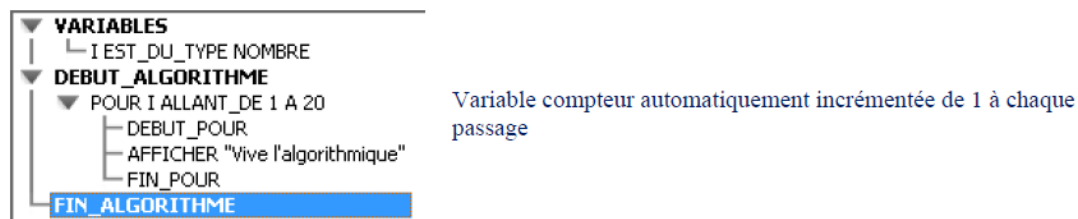
« Si ... Alors ... » et « Si ... Alors ... Sinon ... »



ITÉRATIONS ET BOUCLES

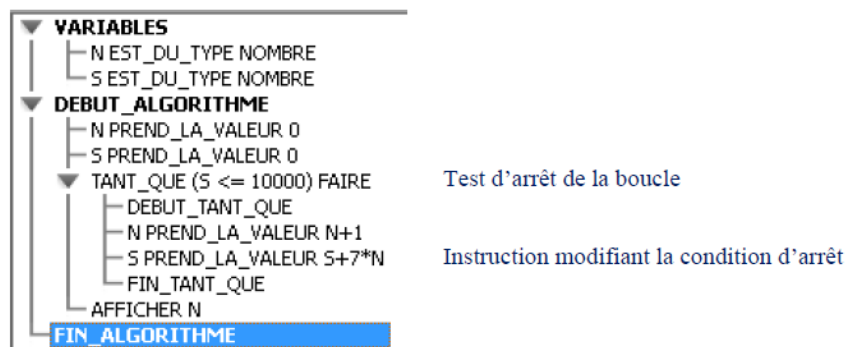
Boucle « Pour ... »

Elle sera utilisée lorsque l'on connaît par avance le nombre d'itérations à réaliser.



Boucle « Tant que ... »

Elle sera utilisée lorsqu'il n'est pas possible de connaître par avance le nombre d'itérations à réaliser.



Exercice 1

Soient deux variables de type nombre A et B. Vous écrivez un programme qui réalise les opérations suivantes :

- Demandez à l'utilisateur une valeur et l'affecter à A ;
- Demandez à l'utilisateur une valeur et l'affecter à B ;
- Inversez les deux valeurs (A reçoit la valeur contenue dans B et B reçoit la valeur contenue dans A).

Exercice 2

Par sécurité, un véhicule doit respecter une distance minimale avec le véhicule qui le précède, afin d'avoir le temps de freiner avant une collision. Ce temps correspond à celui de la perception puis de la réaction du conducteur, ainsi que des possibilités de freinage du véhicule.

Ce temps est fonction de la vitesse du véhicule. Des études statistiques ont montré que cette distance peut être calculée par la formule :

$D = 8 + 0,2 v + 0,003 v^2$, où v représente la vitesse en kilomètres/h et D la distance en mètres. Créer un algorithme avec algobox qui nous donne cette distance en fonction de la vitesse et testez le pour les valeurs de vitesse suivante : 50, 90, 130.

Exercice 3

Soient deux variables de type nombre A et B. Vous écrivez un programme qui réalise les opérations suivantes :

- Demandez à l'utilisateur une valeur et l'affecter à A ;
- Demandez à l'utilisateur une valeur et l'affecter à B ;
- Affichez la plus petite des deux valeurs.

Exercice 4

Pour organiser des rencontres sportives, un moniteur doit connaître l'âge des enfants, puis constituer des équipes homogènes. Parmi les moins de 16 ans et les plus de 6 ans, les catégories sont : "

- "Poussin" de 6 à 7 ans,
- "Pupille" de 8 à 9,
- "Minime" de 10 à 11
- "Cadet" après 12 ans.

Ecrivez un programme qui classe l'enfant dans la catégorie de son âge.

Suggestion : commencer par :

SI âge > 12

ALORS Afficher "....."

SINON

FINSI

Exercice 5

Ecrivez un programme qui demande 5 notes, calcule la moyenne et attribuer la mention correspondante :

Si moyenne ≥ 16 , mention "Très bien"

Si moyenne ≥ 14 , mention "Bien"

Si moyenne ≥ 12 , mention "Assez bien"

Si moyenne ≥ 10 , mention "Passable"

Si moyenne ≥ 8 , "Admis oral du deuxième groupe"

Sinon "Recalé"

Exercice 6

Un magicien demande à un spectateur de penser à un nombre nb et de l'écrire sur une ardoise. Il l'invite à cacher cette ardoise le temps du numéro. Il lui demande d'ajouter 3 puis de multiplier cette somme par le nombre auquel il a pensé au départ (on fait $(nb + 3) * nb$). Enfin il demande de soustraire le carré du nombre de départ du précédent calcul. Au spectateur un peu hagar après tous ces calculs, le magicien

demande de dire à haute voix le résultat final. Instantanément le magicien annonce le nombre pensé déclenchant une salve d'applaudissements alors que le spectateur brandit son ardoise en preuve.

Concevoir un algorithme avec AlgoBox qui calcule le nombre d'arrivée en connaissant le nombre de départ. Pourquoi le magicien devine-t-il le nombre ? Faites afficher la réponse sous forme d'une phrase en fin de votre programme.

Exercice 7

Écrire un programme qui demande un nombre de départ et qui affiche ensuite les dix nombres suivants.

Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

Exercice 8

Écrire un programme qui demande un nombre et qui calcule la somme des entiers consécutifs de 1 jusqu'à ce nombre. Par exemple, si on entre 5, l'algorithme doit calculer $1+2+3+4+5$

Exercice 9

Écrire un programme qui demande un nombre de départ et qui ensuite donne la table de multiplication de ce nombre (de 1 à 9).

Exercice 10

Écrire un programme qui demande un nombre de départ et qui calcule ensuite combien de fois il est possible de lui retirer 10 sans que ce nombre devienne négatif. Le nombre entré doit être positif et vous devez le contrôler. Si le nombre fourni est négatif vous devez redemander à l'utilisateur un nouveau nombre. Vous affichez à la fin le nombre de fois où 10 a pu être retranché du nombre.

Exercice 11

Les élections législatives, en Guignolerie Septentrionale, obéissent à la règle suivante :

- Lorsque l'un des candidats obtient plus de 50 % des suffrages, il est élu dès le premier tour.
- En cas de deuxième tour, peuvent participer uniquement les candidats ayant obtenu au moins 12,5 % des voix au premier tour.

Vous devez écrire un programme qui permette la saisie des scores de quatre candidats au premier tour.

Cet algorithme traitera ensuite le candidat numéro 1 (et uniquement lui) : il dira s'il est élu, battu, s'il se trouve en ballottage favorable (il participe au second tour en étant arrivé en tête du premier tour) ou défavorable (il participe au second tour sans avoir été en tête au premier tour).

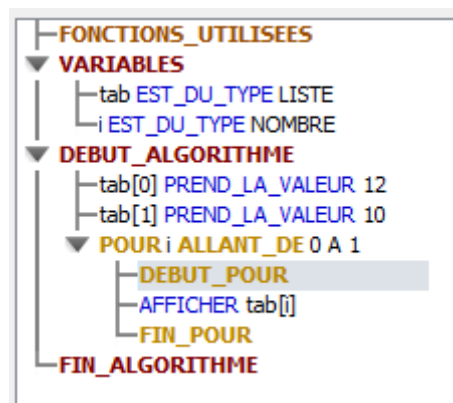
Scores des candidats en %

Scores des candidats en %	A 60 C 13	B 15 D 12	A 38 C 32	B 25 D 4	5A 45 C 12	B 25 D 18	A 8 C 32	B 25 D 35
Résultat du candidat A								

Exercice 12

Les listes AlgoBox sont des listes numérotées de nombres.

Pour utiliser un terme d'une liste dans un calcul, il faut utiliser la syntaxe suivante : nomliste[rang]. Le programme ci-dessous déclare une variable tab de type liste. Le premier élément tab[0] est initialisé avec la valeur 12 et le second élément tab[1] est initialisé avec la valeur 10. Le boucle Pour parcourt chaque élément de tab et affiche sa valeur. Il s'affiche donc 15 10.



A l'aide de liste, réalisez le programme qui répond à l'énoncé suivant.

Un client achète un nombre n d'articles. Vous demandez combien d'articles ont été acheté puis les prix de ces articles que vous stockez dans une variable de type liste Prix ; les prix dans un premier temps sont fournis par ordre décroissant.

Ce sont les soldes. Le premier article bénéficie d'une réduction de 50 %, le second et le troisième article bénéficie chacun d'une réduction de 30 % et le quatrième article d'une réduction de 10 % ; les autres articles ne sont pas soldés. (Si le client achète seulement 1 article, il a droit à 50% ; 2 articles 50 % et 30%, etc..).

En fin de programme, vous affichez la somme des remises et la somme que doit payer le client.

Jeu de test :

	Article 1	Article 2	Article 3	Article 4	Article 5	Article 6
prix	100	75	60	45	12	
prix	323	50	30	12		
prix	154	55	30			

BONUS : les prix des articles sont fournis dans un ordre quelconque. Votre programme les trie par ordre décroissant (vous vous limitez à 3 ou 4 articles)