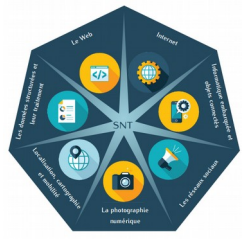
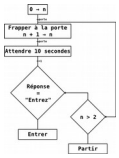


Seconde SNT

Fiche outil Algorithmes



Définitions

Un **algorithme** est une description en langage naturel de la suite des actions effectuées par un programme. L'algorithme utilise un ensemble de mots clés (***début***, ***fin***, ***faire***, ***tant que***, ***répéter***, ***jusqu'à***, …). L'avantage de ce langage est sa transcription facile en langage de programmation dit évolué (Basic, Pascal, C, Java, …).

Exemple : un four à micro-ondes chauffe pendant un temps de fonctionnement tf, jusqu'à ce que tf atteigne le temps tp, programmée par l'utilisateur.

Traduction	***algorithme***
D'abord mesurer la température de fonctionnement tf tant qu'elle est inférieure à tp, la température programmée par l'utilisateur, activer le chauffage du four et mesurer la température de fonctionnement tf. quand la température est atteinte désactiver le chauffage. fin	*début* *mesurer tf* *tant que tf < tp* activer chauffage *mesurer tf* *fin tant que* *désactiver chauffage* *fin*

1. Les variables

Dans un programme informatique, il est souvent nécessaire de stocker provisoirement des valeurs (nombre, texte, etc.). Pour stocker une information au cours d'un programme, on utilise une **variable** qui sera stockée dans la mémoire du système micro-programmé.

Une variable est constituée d'un **nom**, et éventuellement d'une **valeur initiale** donnée au moment de sa **déclaration**.

Une variable peut être de **type** :

- **numérique** : elle peut être déclarée comme un octet, un entier ou un réel :

Type de variable numérique	Plage de valeurs
octet (byte)	0 à 255
entier simple (integer)	-32 768 à 32 767
entier long (long integer)	-2 147 483 648 à 2 147 483 647
réel simple (float)	$-3{,}40\times10^{38}$ à $-1{,}40\times10^{45}$ pour les valeurs négatives $1{,}40\times10^{-45}$ à $3{,}40\times10^{38}$ pour les valeurs positives
réel double (double)	$-1{,}79\times10^{308}$ à $-4{,}94\times10^{-324}$ pour les valeurs négatives $4{,}94\times10^{-324}$ à $1{,}79\times10^{308}$ pour les valeurs positives

- **caractère** (char) : correspond à un caractère alphanumérique (lettre, chiffre, etc.) ;
- **chaîne de caractères** (string) : plusieurs caractères alphanumériques ;
- **booléen** (boolean) : on y stocke uniquement les valeurs logiques VRAI et FAUX ou 0 et 1.

Dans un algorithme, la **déclaration de variables** se fait avant le mot clé début :

variable g de type entier

variables prixHT, tauxTVA, prixTTC de type réel

début

...

fin

Les différentes structures algorithmiques

Il existe trois structures algorithmiques différentes :

- la structure **linéaire** ou séquentielle ;
- la structure **alternatives** ou **conditionnelle** ;
- les structures **répétitives** ou itératives.

1.1. La structure linéaire

Les actions s'exécutent successivement dans l'ordre d'écriture.

Exemple d'une structure linéaire - un feu tricolore placé à un carrefour suit un même cycle à l'infini :

- feu vert allumé, feu orange éteint, feu rouge éteint ;
- temporisation de 20s
- feu vert éteint, feu orange allumé, feu rouge éteint ;
- temporisation de 5s
- feu vert éteint, feu orange éteint, feu rouge allumé ;
- temporisation de 20s

Écrire l'algorithme.

algorithme
Début *Allumer feu vert, éteindre feu orange, éteindre feu rouge* *Attendre 20secondes* *Allumer feu orange, éteindre feu vert, éteindre feu rouge* *Attendre 20secondes* *Allumer feu rouge, éteindre feu vert, éteindre feu orange* *Attendre 20secondes* *fin*

1.2. La structure alternative ou conditionnelle

Elle offre deux possibilités suivant une **condition**. L'exécution d'un des deux traitements dépend du résultat d'un test effectué sur une condition :

traduction	***algorithme***
- si la condition est vraie, seul le premier traitement est exécuté; - si la condition n'est pas vérifiée, seul est effectué le second traitement.	***si*** *condition* ***alors*** *action1* ***sinon*** *action2* ***fin si***

La structure conditionnelle peut aussi se trouver sous la forme **réduite** :

traduction	***algorithme***
si la condition n'est pas vérifiée aucune action n'est exécutée. si la condition est vérifiée alors l'action 1 est exécutée.	***si*** *condition* ***alors*** *action1* ***fin si***

1.3. LES STRUCTURES RÉPÉTITIVES

Il existe différents types de structures répétitives : soit le nombre de répétitions est connu, soit il est inconnu.

1.3.1. Le nombre de répétitions n'est pas connu

répéter … tant que ...	**tant que … faire … fin tant que**
Le traitement est exécuté une première fois puis sa **répétition se poursuit tant que la condition est vérifiée**.	**On commence par tester la condition**, si elle est vérifiée alors le traitement est exécuté **tant que cette condition est vérifiée**.
algorithme	***algorithme***
répéter action **tant que** condition vraie	**tant que** condition vraie **faire** action **fin tant que**
L'action est toujours exécutée au moins une fois	**L'action peut ne jamais être exécutée**

1.3.2. Le nombre de répétitions est connu

La sortie de la boucle d'itération s'effectue lorsque le **nombre souhaité de répétitions est atteint**.

On utilise une variable (ou indice) de comptage d'itération, caractérisé par sa valeur initiale et sa valeur finale.

Si la valeur finale est inférieure à la valeur initiale, la structure est dite décroissante.

Si la valeur finale est supérieure à la valeur initiale, la structure est dite croissante.

structure POUR croissante	**structure POUR décroissante**
algorithme	***algorithme***
pour indice de Vi à Vf action **fin pour**	**pour** indice de Vi à Vf action **fin pour**