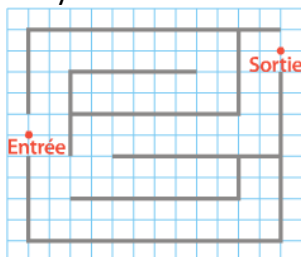


Algorithmique – Fiche 1 – Qu'est ce qu'un algorithme ?

- Un **algorithme** est une suite finie **d'instructions** permettant de réaliser une certaine tâche.
- Exécuter un algorithme c'est effectuer **dans l'ordre** chacune de ses instructions.
- Un algorithme peut être écrit en **langage naturel**, pour être compris par les humains, ou dans un **langage de programmation**, pour être compris par une machine.

Activité 1 : Le labyrinthe



- 1) A partir du point d'entrée du labyrinthe, exécuter l'algorithme suivant :

*Avancer de 1 carreau
Tourner à droite
Avancer de 2 carreaux
Tourner à gauche
Avancer de 9 carreaux*

- 2) Ecrire un algorithme permettant de sortir du labyrinthe en utilisant les instructions suivantes : *Avancer de ... carreau(x) - Tourner à gauche - Tourner à droite*

Activité 2 : Rangement

Les sept tomes d'Harry Potter sont rangés sur une étagère dans l'ordre suivant :

$T2 - T5 - T6 - T4 - T7 - T3 - T1$

- 1) Ecrire un algorithme permettant de ranger ces volumes dans l'ordre en utilisant les instructions suivantes : *Prendre le tome n°... - Insérer le livre avant le tome n° ...*
- 2) Ecrire un autre algorithme en utilisant cette fois-ci uniquement des instructions de la forme : *Echanger les tomes T_i et T_j*

Activité 3 : Constructions géométriques

- 1) Exécuter l'algorithme suivant :

Tracer un segment $[AB]$

Tracer un arc de cercle de centre A et de rayon AB

Tracer un arc de cercle de centre B et de rayon AB

Placer le point C à l'intersection des deux arcs

- 2) Que permet de faire cet algorithme ?
- 3) Ecrire un algorithme qui permet de placer le milieu I d'un segment $[AB]$ en utilisant qu'une règle non graduée et un compas.

Les 3 **phases** d'un algorithme sont :

- **L'entrée** des données.
- Le **traitement** des données.
- La **sortie** du résultat.

Exemple 1 : Dans un moteur de recherche on entre une ensemble de mots-clés et un algorithme permet de générer une page de résultats

Exemple 2 : Dans un GPS on entre le point de départ et le point d'arrivée et un algorithme permet d'afficher une ligne brisée symbolise le chemin à parcourir.

Exemple 3 : Une carte de fidélité utilise un algorithme qui à partir de votre liste de courses permet de sortir des bons de réduction adapté à votre profil de consommateur.

Activité 4 : Formule magique

Voici une formule magique qui permet de trouver votre âge ainsi que votre pointure.

Prenez votre pointure

Multipliez-la par 5

Ajoutez 50

Multipliez le tout par 20

Ajoutez-y 1000

*Ajoutez les 2 derniers chiffres de l'année en cours.
Soustrayez votre année de naissance à ce nombre
Regardez les quatre chiffres obtenus : c'est votre pointure pour les deux premiers et votre âge pour les deux derniers (ou l'âge que vous allez avoir cette année).*

- 1) Exécuter l'algorithme proposé par la formule magique
- 2) Quels sont les données entrées dans l'algorithme ?
- 3) Expliquer comment fonctionne la formule magique.

Activité 5 : IMC

- 1) En utilisant la structure « Entrée – Traitement – Sortie », écrire un algorithme permettant de calculer l'indice de masse corporelle d'un individu sachant que cet indice se calcule à l'aide de la formule : $IMC = \frac{masse}{taille^2}$ (en kg/m^2)
- 2) Exécuter cet algorithme avec votre poids et votre taille. Remarque : Un *IMC* normal est compris entre 18.5 et 25.



Algorithmique – Fiche 2 – La notion de « variable »

- Pour pouvoir fonctionner un algorithme doit pouvoir stocker les données qu'il manipule.
- En informatique, la **mémoire** est un dispositif électronique permettant de stocker des données.
- Une **variable** est un espace de stockage qui contient une information élémentaire.
- Elle est définie par :
 - Un **nom** qui permet de la désigner ;
 - La **valeur** qu'elle contient ;
 - Le **type** de données qu'elle peut stocker : *entier*, *flottant* (nombre à virgule), *chaîne de caractères*.

Ex 1 : La variable nommée x de type entier qui contient la valeur « 5 ».

Ex 2 : La variable nommée *prénom* de type chaîne de caractère qui contient la valeur « Alice ».

- On peut représenter une variable à l'aide d'une **boîte** portant une **étiquette**.



- Il y a 3 instructions de base que l'on peut utiliser avec une variable :
 - La **saisie** : On demande à l'utilisateur de saisir la valeur de la variable.
Ex : *Saisir x*
 - **L'affectation** : On assigne à la variable une certaine valeur. On utilisera alors l'expression *prend la valeur* ou bien le symbole d'affectation « \leftarrow » avec la syntaxe *nom* \leftarrow *valeur* :
Ex : x prend la valeur 5 ou bien $x \leftarrow 5$
 - **L'affichage** : On affiche la valeur de la variable.
Ex : *Afficher x*

Activité 1 : Présentation

```
Saisir prenom
annee_en_cours  $\leftarrow$  2018
Saisir annee_naissance
age  $\leftarrow$  annee_en_cours - annee_naissance
Afficher « Je m'appelle » + prenom
Afficher « J'ai » + age + « ans. »
```

Le signe « + » dans cet algorithme sert à mettre bout à bout deux chaînes de caractères. On parle de **concaténation**.

- 1) Exécuter l'algorithme précédent avec vos données personnelles.
- 2) Identifier les différentes variables présentes dans cet algorithme, leur type et leur valeur.
- 3) Compléter cet algorithme de présentation en rajoutant le nom de famille.

Activité 2 : Variables numériques

Variables : $a ; b ; c$

Saisir a

$b \leftarrow 1$

$c \leftarrow a + b$

$b \leftarrow a + c$

$a \leftarrow b + c$

$c \leftarrow a + b + c$

Afficher c

1) Exécuter l'algorithme avec $a = 1$, en précisant à chaque étape la valeur des variables a, b et c .

2) Quelle valeur doit-on saisir dans l'algorithme pour voir s'afficher 604 ?

Activité 3 : Prix TTC

Variables : $HT ; TTC ; TVA ; \text{taux}$

Entrée :

Saisir HT

$\text{taux} \leftarrow 0,196$

Traitement :

$TVA \leftarrow \text{taux} * HT$

$TTC \leftarrow HT + TVA$

Sortie :

Afficher TTC

- 1) Que permet de faire cet algorithme ?
- 2) Exécuter-le pour calculer le prix TTC d'un ordinateur d'une valeur hors taxe de 550 €
- 3) Depuis le 1^{er} Janvier 2014 le taux de TVA est passé à 20%. Modifier cet algorithme afin qu'il soit en adéquation avec la nouvelle législation.

Activité 4 : Solde -30%

En utilisant la structure « Variables – Entrée – Traitement – Sortie » écrire un algorithme qui applique une solde de 30% sur le prix d'un produit.

Activité 5 : Périmètre d'un rectangle

En utilisant la structure « Variables – Entrée – Traitement – Sortie » écrire un algorithme qui calcule le périmètre d'un rectangle.

Activité 6 : Formule magique

Un magicien demande à 3 personnes de choisir un chiffre (de 0 à 9). Il demande à la 1^{ère} personne : « Multiplie ton chiffre par 2, puis ajoute 3 et multiplie par 5. Donne le résultat obtenu à ton voisin ». Il demande à la 2^{ème} : « Ajoute-y ton chiffre puis multiplie par 10 et donne le résultat à ton voisin ». Il demande à la 3^{ème} personne « Ajoute-y ton chiffre et donne-moi le résultat ». Le nombre obtenu moins 150 est composé par les 3 chiffres, dans l'ordre, choisis par les 3 personnes.

- 1) Tester la formule magique.
- 2) Ecrire un algorithme permettant d'exécuter cette formule magique.
- 3) Expliquer son fonctionnement.



Algorithmique – Fiche 3 – Instruction conditionnelle

- Pour effectuer certaines tâches complexes, une machine doit parfois « prendre une décision ».
- Dans un algorithme, l’instruction conditionnelle permet de tester une **condition** qui peut être soit *vraie* soit *fausse*, et dont le résultat déterminera quelles sont les instructions qui seront exécutées.
- L’instruction conditionnelle se présente sous la forme suivante :

Si Condition Alors

Bloc d’instructions 1

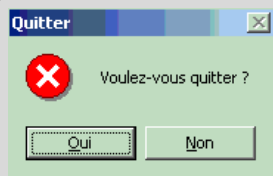
Sinon

Bloc d’instructions 2

Fin Si

- Si la condition est vraie alors le *premier* bloc d’instructions sera exécuté. Sinon, la condition est fausse et c’est alors le *deuxième* bloc d’instructions qui sera exécuté.

Ex 1 : La boîte de dialogue qui s’affiche lors de la fermeture d’une application utilise une instruction conditionnelle.



Variables : Reponse

Afficher « Voulez-vous quitter ? »

Saisir Reponse

Si Reponse = Oui Alors

Fermer l’application

Sinon

Retour à l’application

Fin Si

- Le « Sinon » n’est pas obligatoire. S’il n’est pas présent, lorsque la condition est fausse, on passe tout simplement à la suite de l’algorithme.

- Il est parfois nécessaire de tester une condition en plusieurs étapes. Dans ce cas, on utilise l’instruction *Sinon si* :

Ex 2 : Cet algorithme lit un nombre en entrée et renvoie son signe :

Variables : x

Saisir x

Si x > 0 Alors

Afficher « Positif »

Sinon Si x < 0

Afficher « Négatif »

Sinon

Afficher « Nul »

FinSi

Activité 1 : Etat de l’eau

Ecrire un algorithme qui donne l’état de l’eau (solide, liquide ou gazeux) selon sa température.

Activité 2 : Maximum

Variables : a, b

Entrée :

Saisir a

Saisir b

Traitement :

Si $a \leq b$ Alors

$M \leftarrow b$

Sinon

$M \leftarrow a$

Fin Si

Sortie :

Afficher M

- 1) Exécuter cet algorithme avec les nombres :
a. $a = 4$ et $b = 9$ b. $a = 14$ et $b = 5$
- 2) Que permet d’afficher cet algorithme.
- 3) Ecrire un algorithme qui prend en entrée trois nombres a, b et c et affiche le maximum.

Activité 3 : Nature d’un triangle.

- 1) Ecrire un algorithme qui prend en entrée les longueurs des côtés d’un triangle et renvoie si ce triangle est rectangle ou non.
- 2) Compléter cet algorithme pour que celui-ci affiche la nature du triangle (rectangle, isocèle ou équilatéral).

Activité 4 : Frais kilométriques.

Un employeur propose à ses salariés le barème suivant pour le calcul de leur frais de déplacement :

Distance Domicile/Travail	Valeur fixe	Valeur kilométrique
De 1 à 16 km	0,7584 €	0,1895 €/km
De 17 à 32 km	0,2440 €	0,2110 €/km
De 32 à 64 km	2,0181 €	0,1557 €/km
De 64 à 109 km	2,8159 €	0,1451 €/km

Ecrire un algorithme qui prend en entrée la distance entre le domicile et le travail d’un salarié et permet de calculer le montant de ses frais de déplacements journaliers (aller-retour)



Activité 5 : Impôts sur le revenu

Pour calculer le montant de l'impôt sur le revenu, on applique sur le revenu annuel, un barème progressif qui comporte 5 tranches d'imposition :

- Fraction de revenu jusqu'à 9 807€ : 0%.
- Fraction de revenu de de 9 807€ à 27 086€ : 14%.
- Fraction de revenu de 27 086€ à 72 617€ : 30%.
- Fraction de revenu de 72 617€ à 153 783€ : 41%.
- Fraction de revenu au-delà de 153 783€ : 45%.

L'algorithme suivant permet de calculer le montant de l'impôt pour une personne seule déclarant un revenu annuel compris dans les trois premières tranches d'imposition ($< 72\,617\text{€}$)

Variables : *Revenu, Impot*

Entrée :

Saisir Revenu

Traitement :

Si Revenu \leq 9807 Alors

Impot \leftarrow 0

Sinon Si 9807 $<$ Revenu $<$ 27086 Alors

*Impot \leftarrow 0,14 * (Revenu – 9807)*

Sinon

*Impot \leftarrow 2419 + 0,30 * (Revenu – 27086)*

// 2419 correspond au 14% de la 2^{ème} tranche.

Fin Si

Sortie :

Afficher Impot

- 1) Exécuter l'algorithme précédent pour calculer le montant de l'impôt pour une personne percevant un revenu annuel de 25000€? 32000€?
- 2) Compléter l'algorithme pour qu'il prenne en compte les autres tranches d'imposition.
- 3) Si le foyer comporte plusieurs personnes, alors le foyer fiscal comporte plusieurs *parts* :

Nombre d'enfants	Nombre de parts
0	2
1	2.5
2	3
3	4
4	5

Calcul du nombre de part pour les couples mariés en fonction du nombre d'enfants.

Pour calculer l'impôt, on commence par diviser le revenu du foyer par le nombre de part, on applique au résultat le barème progressif de l'impôt, puis on re-multiplie le résultat obtenu par le nombre de part.

- a. Compléter l'algorithme précédent pour qu'il prenne en compte le nombre de part du foyer.
- b. Appliquer cet algorithme pour calculer le montant de l'impôt d'un couple marié avec 2 enfants ayant déclaré 55 950€.



Algorithmique – Fiche 4 – Boucle itérative

- Pour effectuer certaines tâches, il est parfois nécessaire d'exécuter plusieurs fois une même suite d'instructions.
- Dans un algorithme, la **boucle itérative** permet de **répéter** un certain nombre de fois un même bloc d'instruction.
- La boucle itérative se présente sous la forme suivante :

Pour i allant de 1 à n

Bloc d'instructions

Fin Pour

- n représente le **nombre de répétition**. Le bloc d'instructions sera alors donc exécuté n fois

Ex 1 : Algorithme « Trois fois Bonjour »

<u>Algorithme</u>	<u>Exécution</u>
Pour i allant de 1 à 3	Bonjour
Afficher « Bonjour »	Bonjour
Fin Pour	Bonjour

- La variable i est un **compteur**. Elle augmente automatiquement de 1 à chaque tour. On peut l'utiliser dans le bloc d'instruction à répéter

Ex 2 : Algorithme « 1,2,3 Soleil »

<u>Algorithme</u>	<u>Exécution</u>
Pour i allant de 1 à 3	1
Afficher i	2
Fin Pour	3
Afficher « Soleil »	Soleil

Activité 1 : Listes de nombres

Variables : n, x

Saisir n

Pour i allant de 1 à n

$x \leftarrow 2 * i$

Afficher x

Fin Pour.

- 1) Exécuter cet algorithme avec $n = 5$
- 2) Que permet d'afficher cet algorithme
- 3) Modifier l'algorithme pour que celui-ci affiche :
 - a. Les n premiers nombres impairs.
 - b. Les n premiers multiples de 3.
 - c. Les n premiers carrés.
 - d. Les n premières puissances de 10.

Activité 2 : Rangement

Combien y'a-t-il de façons de ranger 5 livres dans une bibliothèque ? Pour répondre à cette problématique, on propose l'algorithme suivant :

Variables : n, P

Saisir n

$P \leftarrow 1$

Pour i allant de 1 à n

$P \leftarrow P * i$

Fin Pour.

Afficher P

- 1) Exécuter cet algorithme avec $n = 5$.
- 2) Quelle opération effectue la variable P ?
- 3) Expliquer en quoi cet algorithme répond à la problématique de départ.

Activité 3 : Somme des n premiers entiers.

- 1) Ecrire un algorithme permettant de calculer la somme des n premiers entiers.
- 2) Expliquer comment utiliser cet algorithme pour déterminer :
 - a. Le nombre de poignée de main obtenu si tous les élèves de la classe se serre la main.
 - b. Le nombre de diagonale dans un décagone.

Activité 4 : Polygones réguliers.

La somme des angles d'un triangle vaut l'angle plat (180°). La somme des angles d'un quadrilatère convexe vaut deux angles plats (360°). Plus généralement la somme des angles d'un polygone convexe à n côtés vaut $n - 2$ fois l'angle plat. On rappelle qu'un polygone régulier est un polygone dont tous les côtés et les angles sont égaux.

Ecrire un algorithme permettant de dessiner un polygone régulier à n côtés en utilisant les instructions graphiques ci-dessous :

Baisser crayon ; Lever crayon ;

Avancer d'une unité ; Tourner d'un angle de x°

Activité 5 : Double boucle

Que permet d'afficher l'algorithme suivant ?

Pour i allant de 1 à 10

Pour j allant de 1 à 10

Afficher $i * j$

Fin Pour

Fin Pour.



Algorithmique – Fiche 5 – Boucle conditionnelle

- Dans un algorithme, la **boucle conditionnelle** permet de **répéter** une suite d'instructions tant qu'une condition est réalisée.

- La boucle conditionnelle se présente sous la forme suivante :

Tant que Condition

Bloc d'instructions

Fin Tant que

- Le bloc d'instructions sera alors répété tant que la condition reste vraie. Lorsque que la condition devient fausse, on passe alors à la suite de l'algorithme.

Ex : Algorithme « Code d'accès »

L'algorithme suivant demande à l'utilisateur son mot de passe tant que celui n'est pas correct.



Variables : *mot_de_passe*

Afficher « Saisissez votre mot de passe »

Saisir *mot_de_passe*

Tant que *mot_de_passe* ≠ *v7VnF46u*

Afficher « Saisissez votre mot de passe »

Lire *mot_de_passe*

Fin Tant que

Afficher « Identification réussi »

- Le nombre de répétition n'est donc pas fixé à l'avance. Si la condition n'est jamais réalisée, il s'agit alors d'une **boucle infinie**. La plupart des langages de programmation ont une sécurité pour éviter que les machines tournent en boucle sur le même algorithme.

Activité 1 : Division par 2.

Variables : *x*, *c*

x ← 1

c ← 0

Tant que *x* > 0,1

x ← $\frac{x}{2}$

c ← *c* + 1

Fin Tant que

Afficher *c*

- 1) Exécuter cet algorithme.
- 2) Quel est le rôle de la variable *c* ?

Activité 2 : Diminution d'une population

Une ville compte aujourd'hui 11 523 habitants. La mairie estime que dans les prochaines années la population va perdre 3% par an et considère qu'en dessous de 10000 habitants la dynamique de population de la ville sera inversé. On considère l'algorithme suivant :

Variables : *Nb_Hab*, *N*

Nb_Hab ← 11523

N ← 0

Tant que *Nb_Hab* > 10000

Nb_Hab ← 0.97 * *Nb_Hab*

N ← *N* + 1

Fin Tant que

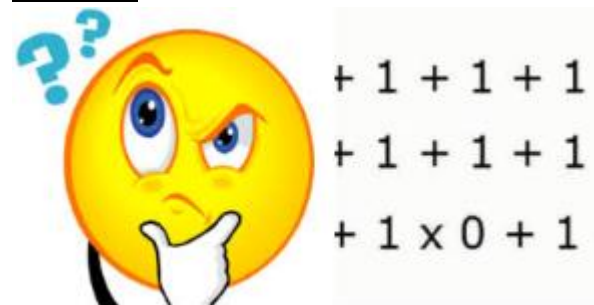
Afficher *N*

- 1) Que permet de faire cet algorithme
- 2) En quelle année la population de cette ville sera-t-elle à nouveau en augmentation ?

Activité 3 : Intérêts bancaires

Pierre place la somme de 100 000€ sur un compte rémunéré à 5% par an. Chaque année, les intérêts s'ajoutent à son capital. Ecrire un algorithme qui permet de déterminer au bout de combien d'année il aura doublé son capital.

Activité 4 : Devinette



Ecrire un algorithme qui propose à l'utilisateur de deviner un nombre entre 1 et 100 choisi aléatoirement par la machine*. L'algorithme demande à l'utilisateur de rentrer un nombre et tant qu'il n'a pas trouvé, il lui donne une indication « Plus grand » ou « Plus petit » pour l'aider trouver. Lorsque le joueur a deviné le nombre, on affichera un message de félicitations ainsi que le nombre de coup qui a été nécessaire.

*Pour choisir un nombre au hasard, on pourra utiliser l'instruction *Entier_aleatoire*(1,100).



Activité 5 : Algorithme d'Euclide

le PGCD de deux nombres entiers a et b est le plus grand nombre qui divise à la fois a et b . Par exemple, on a $PGCD(42; 30) = 6$ car $30 = 2 \times 3 \times 5$ et $42 = 2 \times 3 \times 7$. L'algorithme d'Euclide permet de trouver le PGCD de deux entiers en effectuant des divisions euclidiennes successives :

Variables : a, b, r

Saisir a

Saisir b

Tant que $r \neq 0$

$r \leftarrow \text{reste}(a, b)$

$a \leftarrow b$

$b \leftarrow r$

Fin Tant que

Afficher a

*L'instruction $\text{Reste}(a, b)$ qui renvoie le reste de la division euclidienne de a par b

Exécuter cet algorithme pour trouver le PGCD des nombres 391 et 221.

Activité 6 : Algorithme de Syracuse.

L'algorithme de Syracuse consiste à appliquer le programme de calcul suivant :

- Choisir un nombre entier
 - S'il est pair alors on le divise par 2
 - S'il est impair on le multiplie par 3 et on ajoute 1.
- 1) a. Appliquer ce programme de calcul avec les nombres 5, 10 et 21.
b. Que remarque t-on ?
 - 2) La conjecture de Syracuse (non démontrée à ce jour) énonce que l'on obtient toujours 1 au bout d'un certain nombre d'opérations et la séquence 4,2,1 se répète alors indéfiniment. On appelle temps de vol le nombre d'étape nécessaire avant l'obtention du chiffre 1.
 - a. Quel est le temps de vol de 5, 10 et 21 ?
 - b. Ecrire un algorithme qui demande à l'utilisateur de saisir un nombre n , et affiche les différents termes de la suite de Syracuse jusqu'à ce que l'on obtienne 1.
 - c. Compléter cet algorithme afin que celui affiche aussi le temps de vol du nombre saisi en entrée.

*Pour tester si un nombre est pair ou impair on pourra utiliser l'instruction $\text{Reste}(n, 2)$ qui renvoie le reste de la division euclidienne de n par 2

Activité 7 : De plus en plus petit...

Variables : a, b, r

$N \leftarrow 1$

$x \leftarrow 1$

Tant que $x > 0$

$N \leftarrow N + 1$

$x \leftarrow \frac{1}{N}$

Fin Tant que

Que se passe-t-il lorsque l'on exécute cet algorithme ?

