

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 $\neq$ 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

$x \leftarrow 1$

$c \leftarrow 0$

Tant que $x > 0,1$

$x \leftarrow \frac{x}{2}$

$c \leftarrow c + 1$

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

$Nb_Hab \leftarrow 11523$

$N \leftarrow 0$

Tant que $Nb_Hab > 10000$

$Nb_Hab \leftarrow 0.97 * Nb_Hab$

$N \leftarrow N + 1$

Fin Tant que

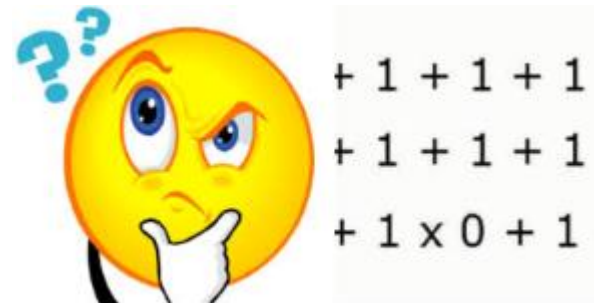
Afficher N

- 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 ?

