

Algorithmes et programmation avec Scratch ou Python.

Formation T1.

Des références aux programmes sont rappelées dans la marge et précédées du symbole \diamond .

\diamond Pour les premières activités : Notion de variable informatique - Séquences d'instructions, boucles, instructions conditionnelles (Programme de cycle 4 - Thème E : Algorithmique et Programmation)

Activité 1. *Variable, et changement de type.*

Ecrire un programme :

- demandant et enregistrant dans des variables le prénom et l'année de naissance d'une personne,
- renvoyant une phrase commençant par le nom de la personne et donnant son âge.

En Python, on aura besoin de d'utiliser la fonction `int()` qui transforme une chaîne de caractères en l'entier correspondant. Par exemple, `int("12")` renvoie l'entier 12 à la place de la chaîne de caractères "12".

Activité 2. *Variable et premier branchement conditionnel.*

1. Proposer (en pseudocode, ou à l'aide d'un organigramme) un algorithme :

- demandant et enregistrant trois coefficients réels a , b et c d'un polynôme du second degré $ax^2 + bx + c$,
- affectant à une variable δ le discriminant de ce polynôme,
- donnant les éventuelles racines de ce polynôme en traitant les différents cas.

2. Implémenter cet algorithme.

Activité 3. *Triangle rectangle.*

1. Ecrire le pseudocode d'un algorithme :

- demandant les mesures des trois côtés d'un triangle
- affectant à une variable *grand_cote* la plus grande ces deux mesures
- affectant à des variables *petit_cote1* et *petit_cote2* les deux autres valeurs (sans tenir compte d'ordre pour ces deux autres valeurs, et avec éventuellement des répétitions),
- affirmant si le triangle est rectangle ou non.

2. Implémenter cet algorithme sans le dernier point.

3. Implémenter complètement cet algorithme.

Activité 4. *Boucles while et for.*

1. Ecrire le pseudocode d'un algorithme :

- prenant en entrée un entier n
- et calculant la somme des carrés des n premiers entiers $1^2 + 2^2 + \dots + n^2$

2. Implémenter cet algorithme en utilisant une structure *for* ou *pour*.

3. Implémenter cet algorithme en utilisant une structure *while* ou *tant que*.

4. Cette question est pour les personnes travaillant avec Python : à partir du travail effectué, définir une fonction f qui à un entier n associe la somme des carrés des n premiers entiers.

5. Ecrire un programme testant pour toutes les valeurs de n jusqu'à 100 que cette somme est bien égale à $n(n + 1)(2n + 1)/6$.

◇ Dans le cas d'une limite infinie, étant donné une suite croissante (u_n) et un nombre réel A , déterminer à l'aide d'un algorithme un rang à partir duquel u_n est supérieur à A . (*programme de TS*)

Activité 5. Problèmes de seuil.

Illustrer le point précédent du programme de TS (on pourra par exemple l'appliquer avec la suite de réels $(u_n)_{n \in \mathbb{N}}$ définie par son premier terme $u_0 = 1$ et la relation de récurrence $u_{n+1} = 1 + u_n^2$).

Activité 6. Algorithme d'Euclide.

Ecrire un programme aboutissant au calcul du p.g.c.d de deux entiers a et b (entrés par l'utilisateur) via l'algorithme d'Euclide. On rappelle que l'algorithme d'Euclide est basé sur l'égalité $p.g.c.d(a, b) = p.g.c.d(b, r)$ pour des entiers a, b, q et r vérifiant $a = bq + r$. On déduit de divisions euclidiennes successives la construction d'une suite finie de couples d'entiers $(a_n, b_n)_{n \in \{1, \dots, N\}}$ de même p.g.c.d et telle que $b_N = 0$.

Avec Scratch, on pourra utiliser le bloc X modulo Y issu des *opérateurs*.

Avec Python, on pourra utiliser l'instruction $a \% b$ qui renvoie le reste dans la division euclidienne de a par b .

◇ Encadrer une racine d'une équation grâce à un algorithme de dichotomie (*programme de seconde*).

Activité 7. Algorithme de dichotomie.

1. Ecrire le pseudocode d'un algorithme de dichotomie dans le contexte suivant : on considère une fonction f continue, strictement croissante, sur un intervalle $[a, b]$, et un réel $k \in [f(a), f(b)]$. On cherche à donner une valeur approchée d'un réel $c \in [a, b]$ tel que $f(c) = k$. L'algorithme :

- demandera à l'utilisateur d'entrer une précision que l'on enregistrera dans une variable *precision*,
- utilisera des variables x et y telles qu'à chaque itération d'une boucle on ait $x \leq c \leq y$,
- renvoie une valeur approchée de c à *precision* près.

Si on travaille avec Scratch, on se contentera de la fonction $f : [0, 3] \rightarrow [0, 9]$ qui à x associe x^2 .

2. Implémenter cet algorithme.

3. Vérifier que le programme fonctionne.

◇ Pour une fonction monotone positive, mettre en oeuvre un algorithme pour déterminer un encadrement d'une intégrale (*programme de TermSTI2D*).

Activité 8. Encadrement d'intégrales.

Illustrer le point suivant du programme de TSTI2D mettant par exemple en oeuvre la méthode des rectangles à gauche dans le cas d'une fonction continue positive décroissante.

◇ Réalisation d'une figure à l'aide d'un logiciel de programmation... (*Programme de cycle 4*) et Même si les logiciels traceurs de courbes permettent d'obtenir rapidement la représentation graphique d'une fonction définie par une formule algébrique, il est intéressant, notamment pour les fonctions définies par morceaux, de faire écrire aux élèves un algorithme de tracé de courbe. (*programme de Seconde*)

Activité 9 pour ceux travaillant avec Scratch gestion du stylo - tracé d'une droite.

1. Réduire la taille du lutin, puis tracer les axes d'un repère en utilisant :

- aller à $x : ? y : ?$ dans la liste *Stylo*,
- *Stylo en position d'écriture* et *Relever le stylo* dans la liste *Stylo*.

2. Ecrire un bloc aboutissant :

- à la demande d'un coefficient directeur et d'une ordonnée à l'origine,
- au tracé de la droite correspondant dans un repère. On utilisera une autre couleur que celle des

axes du repère. Une possibilité est d'utiliser une structure *répéter jusqu'à* et d'introduire des variables x et y .

Activité 10. *Simulation d'expériences aléatoires.*

Pascal et Nicolas jouent au jeu suivant. Pascal lance 3 dés : son score est la somme des deux plus grands chiffres obtenus sur les 3 dés, à laquelle on soustrait le plus petit chiffre apparaissant sur les trois dés (par exemple, si les trois dés indiquent 5, 2, et 4, le score de Pascal est $5 + 4 - 2 = 7$).

A son tour, Nicolas lance 3 dés. Il range les résultats par ordre croissant, et son score est le résultat du dé du milieu. Par exemple, si les trois dés lancés par Nicolas indiquent 5, 2, et 4, le score de Nicolas est 4.

Le gagnant est celui dont le score est le plus élevé.

1. Ecrire un programme simulant cette expérience : le programme donnera pour chacun des joueurs les chiffres indiqués par les trois dés, le résultat de chaque joueur et le gagnant.
2. En répétant cette expérience un grand nombre de fois, donner une estimation de l'espérance du résultat de chacun des joueurs, ainsi qu'une estimation de la probabilité du gagner de chacun.

Activité 11. *Sondage. A traiter avec les listes.* Ecrire un programme qui demande m fois à l'utilisateur de choisir un entier entre 1 et n où m et n sont des entiers choisis au préalable par l'utilisateur. Le programme présentera les réponses obtenues sous forme de pourcentages.