

Une démarche pédagogique pour l'apprentissage de l'algorithmique

Au moment où l'enseignement d'ISN connaît sa deuxième année de mise en œuvre sur le terrain, force est de constater que les ressources disponibles en ligne sont pour bon nombre d'entre elles aussi appropriées les unes que les autres à enrichir cet enseignement en termes de contenus de cours. En revanche, peu de textes sont dévolus à mettre du contenu en relation avec une démarche pédagogique.

Cet article a pour objet de fournir un outil pédagogique ; en ce sens, il s'agit non pas de fournir ici le contenu d'un cours mais de livrer le scénario d'une séance développée avec des précisions sur toutes ses composantes pédagogiques.

L'idée de cet article est d'une part, de présenter dans le détail une démarche pédagogique pour l'apprentissage des algorithmes simples ; d'autre part, de répondre à la question suivante :

Comment construire une séquence pédagogique pour traiter de façon efficace la partie du programme concernant les algorithmes simples ?

Nous ne prétendons pas ici avoir trouvé la pierre philosophale de l'enseignement de l'algorithmique mais nous souhaitons avec ce texte - qui se veut être modeste - fournir le terreau d'une réflexion sur le thème de :

Quelle pédagogie développer dans le cadre de l'enseignement de la spécialité ISN ?

Par souci d'efficacité :

- nous présentons cette démarche en la déclinant sur un exemple : l'algorithme de la somme de deux nombres binaires. Nous avons choisi cet exemple parce qu'il répond à l'enjeu du professeur de transmettre du contenu en « co-construction » avec les élèves ;
- nous utilisons des logos placés dans le texte qui mettent en relation chaque contenu à transmettre ou savoir-faire à acquérir avec une pratique pédagogique clairement identifiée.



Rappelons que les capacités à acquérir dans cette partie du programme sont les suivantes :

Comprendre un algorithme et **expliquer** ce qu'il fait.

Modifier un algorithme existant pour obtenir un résultat différent.

Concevoir un algorithme.

Programmer un algorithme.

◆ **S'interroger** sur l'efficacité d'un algorithme.

C'est dans l'objectif précis d'amener l'élève à acquérir ces compétences que nous plaçons la construction du scénario que nous allons vous présenter.

Nous fournissons ci-dessous un tableau qui répertorie les différents logos que le lecteur retrouvera dans le texte, ces logos ont pour fonction d'associer directement à chaque partie développée le ou les modes pédagogiques qui peuvent être mis en œuvre.

Logo	Traduction pédagogique
	<p>Processus de « co-construction » avec la classe qui correspond à un ensemble de questions posées par le professeur qui amènent progressivement les contenus et les savoir-faire.</p>
	<p>Point délicat qui nécessite de réduire le rythme du cours et de privilégier un processus d'explications interactif.</p>
	<p>Point central du raisonnement sur lequel il est intéressant d'effectuer un diagnostic propre à mesurer le degré d'acquisition des élèves de ce point.</p>
	<p>Phase d'appropriation à mettre en œuvre, c'est-à-dire une lecture simple suivie d'une explication de chaque composante présentée.</p>
	<p>Question à poser aux élèves ; celle-ci apparaît au tableau de façon à ce que l'on ne soit pas uniquement dans un registre auditif. On s'attache à ce que les réponses fournies par les élèves permettent non seulement d'avancer dans le cours mais aussi à ce que ces réponses (ou les non-réponses) fournissent un diagnostic sur les éventuelles difficultés rencontrées.</p>
	<p>Accent à mettre sur l'imbrication des éléments, c'est-à-dire expliquer comment ces derniers interviennent l'un par rapport à l'autre.</p>
	<p>Processus de remédiation à conduire pour amener tous les élèves à surmonter les éventuelles difficultés rencontrées.</p>
	<p>Mise en activité des élèves sur poste informatique. La précision de la saisie des lignes de codes fait partie des savoir-faire à acquérir, ceci étant, il ne faut pas consacrer ce temps à trop de dactylographie.</p>
	<p>Phase de décortication spécialement détaillée à laquelle on doit consacrer du temps et éventuellement développer des explications structurées avec un plan.</p>

Nous considérons dans cet article l'algorithme classique de **l'addition de deux nombres binaires**.

Comme nous l'annonçons précédemment, nous allons mettre en place une démarche qui présentée sur cet algorithme pourra être reprise dans l'apprentissage d'autres algorithmes. Cette démarche consiste à mettre en place des phases systématiques :

- une question ou un mini-projet pour une mise en activité
- un contenu à transmettre ;
- un savoir-faire à développer ;
- une évaluation diagnostic pour passer à la phase suivante.

I) L'appréhension de l'algorithme de l'addition binaire

1) Une question placée au centre du tableau :



*Arthur a additionné les deux nombres binaires suivants : 1001001 et 1101101.
Il a trouvé : 10010110.
Que pensez-vous de ce résultat ?*



Un temps d'appropriation de cet énoncé est mis en place. Ce temps est consacré à donner des éléments de cours sur le langage binaire et à montrer que cette addition peut être posée à l'identique de ce que l'on fait en base 10.

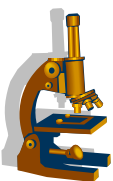
2) Une phase de « Co-construction » pour transmettre des contenus

Les réponses trouvées à la question posée sont recensées. Il s'agit pour le professeur de faire associer à ses élèves l'utilisation de la table aux calculs réalisés dans l'addition de deux nombres à sept chiffres. La compétence à développer est ici de savoir décortiquer le raisonnement conduit lorsque l'on réalise l'addition à la main.



On fait en sorte de mettre en avant la table d'addition de deux chiffres en binaire.

a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	10



On insiste en particulier sur le résultat : **1 + 1 = 10**.



La correction est réalisée collectivement. Elle peut être confiée à un élève qui passe au tableau, le professeur veillant à ce que les différents éléments portés au tableau émanent d'une « co-construction » avec les autres élèves.

Il est utile qu'un processus de remédiation soit conduit pour les élèves en difficulté. Par « bijection réciproque » le professeur peut construire en préambule de la séance une analyse des différentes erreurs produites visant à comprendre le cheminement déductif suivi par les élèves.



3) Des explications détaillées sur la correction pour faciliter l'acquisition d'un savoir faire

L'addition une fois réalisée apparaît au tableau à l'identique de l'image ci-dessous

	1			1			1	
		1	1	0	1	0	0	1
+		1	0	0	1	1	0	1
	1	0	1	1	0	1	1	0

Par « dialogue questionné » avec les élèves, il s'agit ensuite d'insister sur trois points :

la colonne de traitement des unités :

- ❖ on commence par traiter les unités comme dans une addition en base 10, c'est-à-dire les chiffres qui sont écrit le plus à droite du nombre, autrement dit en fin de liste.

la ligne des retenues :

- ❖ cette liste commence après le traitement des unités ;
- ❖ elle va en l'occurrence pour cet exemple jusqu'à un rang après la taille des deux nombres fixée ici à 7 chiffres pour les deux nombres.

la ligne du résultat :

- ❖ le résultat est en l'occurrence un nombre de 8 chiffres alors que les nombres sont de 7 chiffres.

4) Une évaluation diagnostic



Arthur a effacé en partie une addition binaire qu'il avait faite sur son cahier, seul le résultat apparaît : 1110011.

Peux-tu l'aider à retrouver une addition qui donne ce résultat ?



On conduit une phase d'appropriation de cet énoncé : lecture puis explicitation. Le professeur peut amener les élèves à dire qu'il s'agit de traiter une tâche contraire à celle que l'on avait à réaliser auparavant. Il peut être utile de poser une opération à compléter à l'identique de celle figurant ci-dessous :

$$\begin{array}{cccccccc}
 & & \dots & \dots & \dots & \dots & \dots & \dots \\
 + & & \dots & \dots & \dots & \dots & \dots & \dots \\
 \hline
 & & 1 & 1 & 1 & 0 & 0 & 1 & 1
 \end{array}$$

Il s'agit lors de la correction collective mise en place de s'assurer que tous les élèves ont acquis la compétence : **Comprendre** l'algorithme de calcul.

II) Vers une écriture de l'algorithme en langage naturel

1) Présentation d'un mini-projet



On souhaite fournir à Arthur un programme qui effectue la somme de deux nombres binaires de 7 chiffres maximum avec le souhait qu'il pourra le faire tourner chez lui avec un logiciel libre. Pourrais-tu l'aider à nouveau ?

2) L'écriture de l'algorithme en langage naturel

L'idée est ici de produire l'algorithme en langage naturel en s'appuyant sur l'exemple décortiqué dans la première partie, l'opération posée qui a été effectuée apparaît au tableau.

	1			1			1	
		1	1	0	1	0	0	1
+		1	0	0	1	1	0	1
	1	0	1	1	0	1	1	0

3) Un contenu à transmettre : le passage des nombres aux listes et la mise en place des notations

Les nombres en binaire sont notés a et b.



Le professeur explique qu'étant donné que l'on va réaliser des opérations successives faisant intervenir les chiffres de a et de b, nous choisissons de les considérer comme des listes.



Ainsi pour notre exemple : $a = [1, 1, 0, 1, 0, 0, 1]$.

On fait la même chose pour b. Les élèves écrivent la liste.

2) Un savoir-faire : la mise en place des notations

Il s'agit à présent de montrer la lecture informatique d'une liste. En précisant que dans de nombreux langages de programmation, le premier élément de la liste a pour rang zéro, ainsi dans notre liste, nous avons :



$a[0]$ $a[6]$
 $a = [1, 1, 0, 1, 0, 0, 1]$.

3) Deux questions diagnostics posées aux élèves



Que vaut $a[2]$? $b[3]$? et $a[7]$?



Y-a-t-il d'autres listes à définir ?



Il vient alors la définition de la liste s du résultat puis de la liste r des retenues.

III) L'écriture de l'algorithme en langage naturel

L'idée est ici de produire l'algorithme en langage naturel en s'appuyant sur l'exemple décortiqué dans la première partie, l'opération posée qui a été effectuée apparaît au tableau.

	1			1			1	
		1	1	0	1	0	0	1
+		1	0	0	1	1	0	1
	1	0	1	1	0	1	1	0



On rédige la présentation sous la forme d'un tableau à deux colonnes avec :

- à droite l'exemple traité de façon décortiqué ;
- et à gauche, la description en langage naturel de ce qui est réalisé dans la colonne de gauche.

1) Une première tâche à proposer aux élèves



Dans la colonne de droite du tableau ci-dessous, vous reconnaissez l'exemple que nous avons traités précédemment. Vérifier le contenu de cette colonne.

Langage naturel	Traitement de l'exemple
	<p>$a = 1101001$ et $b = 1001101$</p> <p>$a = [1, 1, 0, 1, 0, 0, 1]$ $b = [1, 0, 0, 1, 1, 0, 1]$</p> <p>On ajoute 1 et 1. Cela donne 10 Ici et $s = [0]$ $r = [1]$</p> <p>On a successivement : $r = [1, 0]$ et $s = [0, 1]$ $r = [1, 0, 0]$ et $s = [0, 1, 1]$ $r = [1, 0, 0, 1]$ et $s = [0, 1, 1, 0]$ $r = [1, 0, 0, 1, 0]$ et $s = [0, 1, 1, 0, 1]$ $r = [1, 0, 0, 1, 0, 0]$ et $s = [0, 1, 1, 0, 1, 1]$ $r = [1, 0, 0, 1, 0, 0, 1]$ et $s = [0, 1, 1, 0, 1, 1, 0, 1]$</p> <p>On renverse la liste s pour obtenir le résultat de l'addition : $[1, 0, 1, 1, 0, 1, 1, 0]$</p>

2) Une correction collective pour le réinvestissement d'un contenu



La correction est conduite collectivement en « co-construction » avec la classe, il s'agit ici de réinvestir les connaissances acquises sur l'addition binaire en passant de l'addition posée à son écriture avec l'intervention des listes r et s que l'on remplit progressivement.

Il est utile que l'addition posée apparaisse au tableau afin que les explications concernant la construction progressive des listes r et s aient un support visuel concret.

	1			1			1	
		1	1	0	1	0	0	1
+		1	0	0	1	1	0	1
	1	0	1	1	0	1	1	0

3) Une nouvelle mini-tâche pour une appropriation de savoir-faire

L'idée est à présent de passer de l'exemple traité précédemment au cas général en sachant gérer l'écriture indicielle des listes.




Dans la colonne de gauche, nous avons fait apparaître en partie le développement de l'algorithme en langage naturel. Compléter les espaces vides (...).

Langage naturel	Traitement de l'exemple
<p>Initialisation On se donne deux entiers en binaire a et b. On les écrit sous forme de liste $a = [a(0), a(1), \dots, \dots, \dots, \dots]$ $b = [\dots, \dots, \dots, \dots, \dots, \dots, \dots]$ On définit la listes vides s et la variable r qui au départ vaut 0.</p>	<p>$a = 1101001$ et $b = 1001101$</p> <p>$a = [1, 1, 0, 1, 0, 0, 1]$ $b = [1, 0, 0, 1, 1, 0, 1]$</p>
<p>Déroulement On ajoute les chiffres des unités, c'est-à-dire $a[\dots]$ et $b[\dots]$. On met le chiffre des unités dans la liste s. Si la retenue vaut 1, on affecte la valeur 1 à la variable r.</p>	<p>On ajoute 1 et 1. Cela donne 10 Ici et $s = [0]$ $r = [1]$</p>
<p>Jusqu'au traitement de $a[0]$ et $b[0]$</p> <p>On additionne les chiffres sans oublier la conservation du chiffre des et celui de(on augmente <u>à droite</u> les listes r et s).</p>	<p>On a successivement :</p> <p>$r = [1,0]$ et $s = [0,1]$ $r = [1,0,0]$ et $s = [0,1,1]$ $r = [1,0,0,1]$ et $s = [0,1,1,0]$ $r = [1,0,0,1,0]$ et $s = [0,1,1,0,1]$ $r = [1,0,0,1,0,0]$ et $s = [0,1,1,0,1,1]$ $r = [1,0,0,1,0,0,1]$ et $s = [0,1,1,0,1,1,0,1]$ $[1,0,1,1,0,1,1,0]$</p>
<p>On reverse la liste s On fait afficher la liste s</p>	



La correction de chacune de ces mini-tâches est élaborée collectivement en « co-construction » avec la classe.

La colonne de gauche est remplie progressivement

Langage naturel	Traitement de l'exemple
<p>Initialisation On se donne deux entiers en binaire a et b. On les écrit sous forme de liste $a = [a(0), a(1), a(2), a(3), a(4), a(5), a(6)]$ $b = [b(0), b(1), b(2), b(3), b(4), b(5), b(6)]$ On définit deux listes vides r et s</p> <p>Déroulement On ajoute les chiffres des unités, c'est-à-dire a[6] et b[6]. On met le chiffre des unités dans la liste s. On met la retenue éventuelle dans r.</p> <p>Jusqu'au traitement de a[0] et b[0] On additionne les chiffres sans oublier la conservation du chiffre des unités et celui de la retenue (on augmente <u>à droite</u> les listes r et s).</p>  <p>On renverse la liste s On fait afficher la liste s</p>	<p>a = 1101001 et b = 1001101</p> <p>a = [1 , 1 , 0 , 1 , 0 , 0 , 1] b = [1 , 0 , 0 , 1 , 1 , 0 , 1]</p> <p>On ajoute 1 et 1. Cela donne 10 Ici et s = [0] r = [1]</p> <p>On a successivement :</p> <p>r = [1,0] et s = [0,1] r = [1,0,0] et s = [0,1,1] r = [1,0,0,1] et s = [0,1,1,0] r = [1,0,0,1,0] et s = [0,1,1,0,1] r = [1,0,0,1,0,0] et s = [0,1,1,0,1,1] r = [1,0,0,1,0,0,1] et s = [0,1,1,0,1,1,0,1]</p> <p>[1,0,1,1,0,1,1,0]</p>

4) Un diagnostic de l'appropriation de l'algorithme en langage naturel



Nous avons effacé la colonne de droite, il s'agit pour toi à présent de la remplir à partir l'exemple de ton choix en prenant deux nombres d'au plus 7 chiffres

Langage naturel	Traitement de l'exemple
<p>Initialisation On se donne deux entiers en binaire a et b. On les écrit sous forme de liste $a = [a(0), a(1), a(2), a(3), a(4), a(5), a(6)]$ $b = [b(0), b(1), b(2), b(3), b(4), b(5), b(6)]$ On définit deux listes vides r et s</p> <p>Déroulement On ajoute les chiffres des unités, c'est-à-dire a[6] et b[6]. On met le chiffre des unités dans la liste s. On met la retenue éventuelle dans r.</p> <p>Jusqu'au traitement de a[0] et b[0] On additionne les chiffres sans oublier la conservation du chiffre des unités et celui de la retenue (on augmente <u>à</u></p>	<p>a = et b =</p> <p>a = [...] b = [...]</p> <p>On ajoute ... et Cela donne Ici et s = [...] r = [...]</p> <p>On a successivement :</p> <p>r = [] et s = [] r = [] et s = [] r = [] et s = []</p>

<u>droite</u> les listes r et s).	r = [] et s = [] r = [] et s = [] r = [] et s = []
On reverse la liste s On fait afficher la liste s	[]

Il s'agit à partir de cette question d'effectuer un rétro-pédalage et de vérifier sur un exemple la conformité de l'écriture de l'algorithme en langage naturel. Un processus de remédiation pourra être mis en œuvre en particulier pour les élèves qui rencontrent des difficultés à « alimenter » les listes r et s.



Cette étape d'écriture de l'algorithme en langage naturel a conduit les élèves à travailler la compétence de **Comprendre** un algorithme.

IV) Conception et programmation d'un algorithme simple : l'addition de deux chiffres binaires.



Comme en base 2, c'est comme en base 10 : la somme de deux chiffres peut donner un nombre d'un seul chiffre ou de deux chiffres, par exemple la somme de 1 et de 1 donne 10. Quand on fait tourner l'algorithme, à chaque addition réalisée de 2 chiffres, il va falloir conserver d'un côté le chiffre des unités pour le mettre dans la variable U et le chiffre des retenues que l'on va placer dans la variable R.

1) L'écriture de l'algorithme en langage naturel



Construire dans la colonne de gauche du tableau ci-dessous un algorithme en langage naturel qui produise la somme de deux chiffres binaires à l'identique de la table de sommation donnée dans la colonne de droite.

Algorithme en langage naturel	La table d'addition															
	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>a+b</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td>0</td> </tr> <tr> <td>0</td> <td>1</td> <td>1</td> </tr> <tr> <td>1</td> <td>0</td> <td>1</td> </tr> <tr> <td>1</td> <td>1</td> <td>10</td> </tr> </tbody> </table>	a	b	a+b	0	0	0	0	1	1	1	0	1	1	1	10
a	b	a+b														
0	0	0														
0	1	1														
1	0	1														
1	1	10														


La mise en activité des élèves a pour objectif de les conduire à acquérir la compétence **Concevoir** un algorithme. Il convient donc –après avoir conduit une large phase d'appropriation de la tâche à effectuer – de les laisser chercher.



Des explications doivent être fournies visant à conduire les élèves à envisager 4 cas possibles puis si possible 3 cas.



La production de l'algorithme est produite en « co-construction » avec la classe.

Algorithme en langage naturel	La table d'addition		
 <p>Initialisation On se donne deux chiffres a et b.</p> <p>Déroulement Si a = 0 et b = 0 alors U = 0 et R = 0 Sinon si a = 1 et b = 1 alors U=0 et R=1 Sinon U = 1 et R = 0.</p>	a	b	a+b
	0	0	0
	0	1	1
	1	0	1
	1	1	10

Une question diagnostic



Dans le tableau ci-dessous, on a placé l'algorithme que nous venons de construire dans la colonne de gauche et la table d'addition vide dans la colonne de droite. À partir de la mise en œuvre de l'algorithme, remplir la table d'addition fournie dans la colonne de gauche.


Il convient de conduire un processus de remédiation concernant les éventuelles difficultés d'appréhension de l'algorithme produit.



2) Le développement de l'algorithme sous Python





Complétez la colonne de gauche du tableau ci-dessous qui donne le développement en langage Python de l'algorithme donné en langage naturel dans la colonne de droite.

Programme en langage Python	Algorithme en langage naturel
 <p>Nous définissons la fonction somde2deuxchif qui nous aurons à fournir en entrée.</p> <pre>>>> def somde2chiffres(a,b) : if a == 0 and b == 0 : Else : If and : U,R = 0,1 Else : Return [U,R]</pre>	<p>Initialisation On se donne deux chiffres a et b.</p> <p>Déroulement Si a = 0 et b = 0 alors U = 0 et R = 0 Sinon si a= 1 et b = 1 alors U=0 et R=1 Sinon U = 1 et R = 0. Renvoyer la liste [U, R]</p>



Ce développement en Python (remplissage de la colonne de gauche) se fait là-aussi en « co-construction » avec les élèves.

L'idée est ici de développer cette fois-ci la compétence **Programmer** un algorithme.

Programme en langage Python	Algorithme en langage naturel
 <p>Nous définissons la fonction somde2chiffres</p> <pre>>>> def som2chiffres(a,b): if a==0 and b==0: U,R=0,0 else : if a==1 and b==1: U,R=0,1 else: U,R=1,0 return (U,R)</pre>	 <p>Initialisation On se donne deux chiffres a et b.</p> <p>Déroulement Si a = 0 et b = 0 alors U = 0 et R = 0 Sinon si a= 1 et b = 1 alors U=0 et R=1 Sinon U = 1 et R = 0</p>

Une question diagnostic



À l'aide de l'algorithme de la colonne de gauche, remplir le tableau de la colonne de droite.

Programme en langage Python	Tableau donnant l'addition décortiquée																									
<pre>>>> def som2chiffres(a,b): if a==0 and b==0: U,R=0,0 else : if a==1 and b==1: U,R=0,1 else: U,R=1,0 return (U,R)</pre>	<table border="1"> <thead> <tr> <th>a</th> <th>b</th> <th>U</th> <th>R</th> <th>a+b</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0</td> <td></td> <td></td> <td></td> </tr> <tr> <td>0</td> <td>1</td> <td></td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>0</td> <td></td> <td></td> <td></td> </tr> <tr> <td>1</td> <td>1</td> <td></td> <td></td> <td></td> </tr> </tbody> </table>	a	b	U	R	a+b	0	0				0	1				1	0				1	1			
a	b	U	R	a+b																						
0	0																									
0	1																									
1	0																									
1	1																									

V) Le développement de l'addition binaire en langage Python

On revient à nouveau au projet de construire un programme qui additionne deux nombres binaires de 7 chiffres. Il s'agit à présent d'utiliser l'algorithme de somme de deux chiffres binaires pour produire la somme de deux nombres binaires de 7 chiffres.

1) Étape préalable : la prise en compte des retenues pour passer d'une étape du calcul à la suivante.



Le tableau ci-dessous comporte tous les cas possibles concernant les deux chiffres à additionner et la retenue à prendre en compte quand on réalise une étape dans la somme binaire. Les colonnes U et R fournissent les résultats produits par l'addition des chiffres a et b. Compléter le tableau.

Somme à effectuer					Résultat obtenu	
a	b	a+b		retenue	U posée	Nouvelle retenue
0	0	U	R			
0	0	0	0	0		
0	1	1	0	0		
1	0	1	0	0		
1	1	0	1	0		
0	0	0	0	1		
0	1	1	0	1		
1	0	1	0	1		
1	1	0	1	1		



La correction est fournie collectivement en « co-construction » avec la classe. Le tableau une fois construit est décortiqué, on observe les points centraux mentionnés dans les encadrés bleus et les encadrés rouges.



Somme à effectuer					Résultat obtenu	
a	b	a+b		retenue	U posée	Nouvelle retenue
0	0	U	R			
0	0	0	0	0	0	0
0	1	1	0	0	1	0
1	0	1	0	0	1	0
1	1	0	1	0	0	1
0	0	0	0	1	1	0
0	1	1	0	1	0	1
1	0	1	0	1	0	1
1	1	0	1	1	1	1

Les cadres bleus montrent que lorsque la retenue vaut « 0 », la somme des chiffres a et b fournit directement les valeurs de U posée et de la nouvelle retenue.



Les cadres rouges montrent que l'on peut partir de a et de b et considérer trois cas de figure qui vont conduire à U posée et à la nouvelle retenue.

2) Écriture de l'algorithme en langage naturel et développement sous Python.

Nous ré-utilisons les mêmes cadres bleus et rouges pour écrire l'algorithme.

Question posée aux élèves



Dans le tableau ci-dessous, nous avons fait figurer dans la colonne de gauche l'algorithme de la somme binaire développé sous Python, fournir dans la colonne de droite l'algorithme en langage naturel.

Le tableau ci-dessous contient au départ uniquement la colonne de droite. Il est rempli progressivement en « co-construction » avec la classe.

Programme Python	Langage naturel
------------------	-----------------

```

>>> def somme(a,b):
    S=[]
    L=somme2chiffres(a[6],b[6])
    S.append(L[0])
    retenue=L[1]
    i=5
    while i>=0:
        if retenue==0:
            L=somme2chiffres(a[i],b[i])
            S.append(L[0])
            retenue=L[1]
        else:
            if a[i]!=b[i]:
                S.append(0)
                retenue=1
            else:
                if a[i]==0:
                    S.append(1)
                    retenue=0
                else:
                    S.append(1)
                    retenue=1
            i=i-1
    if retenue==1:
        S.append(1)
    S.reverse()
    print(S)

```

On part d'une liste S vide
 On ajoute les chiffres a[6] et b[6].
 On met U dans la liste S.
 La retenue R est placée dans la variable retenue.
 i vaut 5 car on vient de traiter i=6
 Jusqu'au traitement de a[0] et b[0], on effectue la somme de a[i] et b[i] avec la retenue

Si la retenue vaut « 0 », on effectue la somme de a(i) et de b(i) qui donne l'unité à conserver et la retenue

Sinon (la retenue vaut « 1 »)
 Si les chiffres sont différents
 alors l'unité à poser vaut 0
 et la retenue vaut « 1 »
 Sinon (les deux chiffres sont égaux)

 S'ils valent « 0 »
 alors l'unité vaut « 1 »
 et la retenue vaut « 0 »

 Sinon
 alors l'unité vaut « 1 »
 et la retenue vaut « 1 »

On va considérer les deux chiffres qui viennent juste après.
 A la fin de toutes les sommes, s'il reste une retenue égale à 1, elle sera le 1^{er} chiffre de notre somme, il convient de l'ajouter.
 On a augmenté à droite la liste s, il convient de la renverser.
 On fait afficher la liste s

3) Implémentation du programme et exécution

Le programme peut être tapé à la fin de la séance par les élèves qui doivent apprendre à maîtriser l'indentation sous Python et gagner en capacité de déchiffrement des différents messages d'erreurs fournis éventuellement par Python.

Il reste en fin de séance à tester le programme sur quelques exemples. On utilise à cet effet le verbe « tester » en indiquant à l'élève que :

- si le programme ne tourne pas ou ne donne pas le bon résultat alors il faut reprendre le programme dans sa syntaxe. Si celui-ci continue à renvoyer le mauvais résultat, alors il faut travailler sa partie mathématique.

- si le programme donne le résultat convenable sur plusieurs exemples alors on peut avoir confiance en ce que l'on a fait.

Conclusion

Avec ce texte, il ne s'agit aucunement de fournir une démarche à suivre de façon dogmatique, l'idée est surtout de développer un scénario en décortiquant toutes les phases pédagogiques à mettre en œuvre pour transmettre un contenu.

Tel ou tel professeur reprendra tout ou partie de ce texte, l'important est qu'il serve au développement de « bonnes » pratiques pour l'enseignement de l'algorithmique.

Jean-Alain Roddier
IA-IPR de mathématiques
Académie de Clermont-Ferrand