

Cet exercice est proposé aux élèves de seconde en début d'année. Il permet la mise en place progressive de différentes notions au programme : traitement conditionnel et itérations en ce qui concerne l'algorithmique, et logique par exemple.

Il peut être utile de faire découvrir aux élèves la fonction « partie entière » avant de débiter cet exercice.

On peut également utiliser la fonction « mod » présente sur Scratch ou sur le tableur, et prendre le temps d'expliquer aux élèves qu'elle se construit à l'aide de la fonction « partie entière ».

Une première séance avec pour seul matériel le papier et le crayon paraît nécessaire pour que l'élève comprenne les consignes.

On peut faire travailler les élèves en groupes.

Ceux ci prennent vite conscience de l'intérêt de l'ordinateur ou de la calculatrice pour ce type d'exercice.

Dans les pages suivantes, on propose d'aborder le problème à chaque fois sous quatre facettes différentes : en langage naturel, avec une feuille de calculs sur Excel, avec la TI 82 ou enfin avec le logiciel Scratch. On trouvera en annexe des programmes pour Casio, et un exemple de fichier Excel.

L'écriture de l'algorithme en langage naturel peut être demandée à l'élève en travail personnel, puisqu'elle ne nécessite aucun autre matériel que papier et crayon.

1. Dans un premier temps, on s'intéresse uniquement au passage d'un entier au suivant.

Cela peut paraître peu ambitieux, mais demande à l'élève un réel effort d'abstraction. On propose une progression en trois étapes.

► On travaille sur le passage d'un entier au suivant, sans condition.

Il s'agit donc ici simplement de faire un test de parité, et le traitement adapté selon le résultat de ce test. Par exemple, si l'utilisateur entre 18, le programme donne 9, s'il entre 27, il donne 82.

– En langage naturel :

```
Demander un nombre entier strictement positif n
à A attribuer n
Si A est pair
Alors
    à A attribuer  $\frac{A}{2}$ 
Sinon
    à A attribuer  $3A+1$ 
Afficher A
```

– Sur Excel on utilise la formule « =SI(condition; faire si Vrai; faire si Faux) » : l'entier de départ est placé en A1, et en A2 on écrit :

« = SI(EST.PAIR(A1);A1/2;3*A1+1) »

Pour entraîner les élèves à la syntaxe de l'instruction « SI », on peut aussi leur demander de recommencer en utilisant la fonction « EST.IMPAIR » :

« = SI(EST.IMPAIR(A1);3*A1+1;A1/2) »

Si le logiciel ne dispose pas de ces fonctions, on peut utiliser la formule suivante :

« = SI(ENT(A1/2)=A1/2;A1/2;3*A1+1) »

– Sur la TI 82, on crée un programme : le test de parité se fait avec la fonction partie entière, comme ci dessus.

```
program : Syracuse 1
Input « entier de départ », A
If int(A/2)=A/2
Then A/2 → A
Else 3A + 1 → A
End
Disp A
```

- Sur Scratch, on utilise l'instruction « mod » :



- On recommence la partie précédente en ajoutant une contrainte supplémentaire : si l'entier de départ est supérieur à 100, on envoie un message de fin : « ce nombre dépasse 100 ».

- En langage naturel :

Demander un nombre entier strictement positif n
à A attribuer n
Si A est strictement plus grand que 100
Alors
dire « ce nombre dépasse 100 »
Sinon
Si A est pair alors
à A attribuer $\frac{A}{2}$
Sinon
à A attribuer $3A+1$
Afficher A

- Sur tableur, il s'agit également d'une formule avec deux « SI » imbriqués : on remplace ici le message « ce nombre dépasse 100 » par le mot « fin » pour des raisons évidentes de gain de place.

« =SI(ESTNUM(A1);SI(A1>100;"FIN";SI(EST.PAIR(A1);A1/2;3*A+1));"") »

Pourquoi cette nouvelle fonction « ESTNUM » ?

En fait, Excel compare aussi bien des nombres que des chaînes de caractères. Ainsi :

- « FIN » >100 donne VRAI
- « FIN » <100 donne FAUX

Excel considère alors 100 comme une chaîne de caractères et compare comme dans un dictionnaire « FIN » et « 100 » (en sachant qu'un chiffre est avant (donc inférieur à) une lettre.

Il faut donc tester d'abord si la cellule précédente contient une valeur numérique ou non. Avec cette formule, un seul mot « FIN » apparaît sur chaque colonne. Sans la fonction « ESTNUM », le mot serait reproduit par recopie vers le bas.

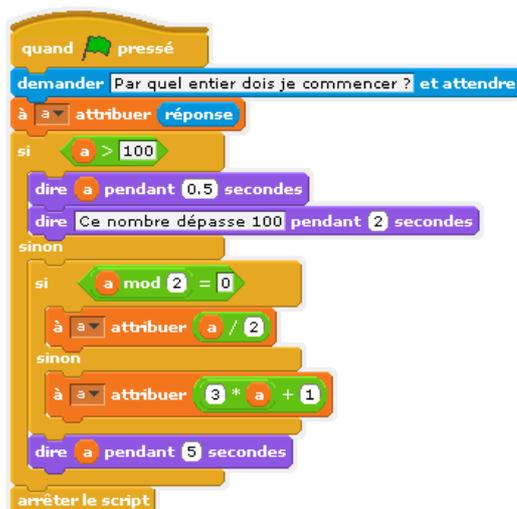
- Sur la T.I. :

```

program : Syracuse 2
Input « Entier de départ ? », A
If A>100
Then Disp "ce nombre dépasse 100"
Else
  If int(A/2)=A/2
  Then A/2 → A
  Else 3A + 1 → A
  End
Disp A
End

```

– Sur Scratch :



► On complique encore un peu : si l'entier de départ est 1 ou est strictement supérieur à 100, on renvoie un message de fin .

Il suffit d'utiliser l'instruction « OU » par exemple. On peut aussi faire travailler les élèves en leur demandant d'utiliser l'instruction « ET ».

Sur tableur, on renvoie le seul mot « fin », mais dans les autres langages il est facile de différencier le message selon qu'on arrive à 1 ou qu'on dépasse 100.

– En langage naturel :

```
Demander un nombre entier strictement positif n
à A attribuer n
Si A est strictement plus grand que 100
Alors
  dire « ce nombre dépasse 100 »
Sinon
  Si A=1
  Alors
    dire « fin »
  Sinon
    Si A est pair
    Alors
      à A attribuer  $\frac{A}{2}$ 
    Sinon
      à A attribuer  $3A+1$ 
  Afficher A
```

– Sur tableur, cela pourrait donner :

« =SI(ESTNUM(A1); SI(OU(A1=1;A1>100);"FIN";SI(EST.PAIR(A1);A1/2;3*A1+1));"") »

ou

« =SI(ESTNUM(A1); SI(ET(NON(A1=1);A1<=100);SI(EST.PAIR(A1);A1/2;3*A1+1);"FIN");"") »

Ces formules peuvent paraître compliquées, mais l'utilisation du tableur se révèle avantageuse pour l'écriture de toutes les chaînes et le calcul de leur longueur éventuelle. Il suffira d'utiliser la poignée de recopie.

– Sur la T.I. :

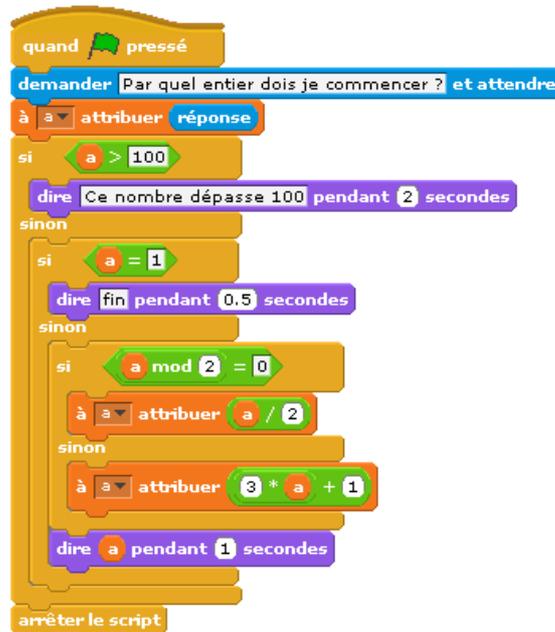
```
program : Syracuse 3
Input « Entier de départ ? », A
If A>100
```

```

Then Disp "ce nombre dépasse 100"
Else
  If A=1
  Then Disp "Fin"
  Else
    If int(A/2)=A/2
    Then A/2 → A
    Else 3A + 1 → A
      Disp A
    End
  End
End
End

```

– Sur Scratch :



2. Nous sommes maintenant prêts pour itérer le processus : création d'une chaîne, et calcul de sa longueur éventuelle.

– En langage naturel :

```

Demander un nombre entier strictement positif  $n$ 
à  $A$  attribuer  $n$ 
à  $C$  attribuer 1
  Tant que  $A \neq 1$  et  $A < 101$  faire :
    Si  $A$  est pair
    Alors
      à  $A$  attribuer  $\frac{A}{2}$ 
    Sinon
      à  $A$  attribuer  $3A + 1$ 
      à  $C$  attribuer  $C + 1$ 
    Fin tant que
  Si  $A = 1$ 
  Alors
    Dire « La longueur est »
    Afficher  $C$ 
  Sinon
    Dire « Chaîne cassée »

```

Les élèves découvrent ici l'instruction « tant que ». Ne pas oublier, même en langage usuel, de signaler la fin du traitement correspondant à cette instruction.

La variable C donne la longueur de la chaîne : elle est initialisée à 1 avant le premier passage dans la boucle, et augmentée de 1 à chaque itération. (Elle est initialisée à 1 et non à 0 pour tenir compte du dernier entier (1) de chaque chaîne. En effet pour une chaîne de longueur n , il y a $n - 1$ passages dans la boucle « tant que ».

L'itérative « tant que » est une structure qui permet l'exécution d'une action ou d'une séquence d'actions tant qu'une condition est vérifiée :

```
TANT QUE condition
      Instructions
FIN DU TANT QUE
```

Si la condition *condition* est vraie, les instructions dans *Instructions* sont exécutées. Puis, on retourne tester la condition. Dans le cas contraire, l'itération est terminée. On dit alors que l'on sort de la boucle.

La condition est toujours évaluée avant de faire le traitement. Donc, si la condition n'est jamais vraie, le traitement n'est jamais effectué. (Inversement, si elle est toujours vraie, il faudra un marteau et un burin pour sortir du programme!)

- Sur tableur, cela ne pose aucun problème, puisqu'il suffit pour écrire la chaîne complète d'utiliser la poignée de recopie sur une plage suffisamment grande pour que toutes les chaînes soient complètes.

L'éventuelle longueur d'une chaîne est donnée par la formule suivante, placée en fin de chaque colonne :

$$= \text{Si}(\text{MAX}(\text{plage}) > 101 ; \text{« cnc »} ; \text{NB}(\text{plage}))$$

Ici, « cnc » est renvoyé pour « chaîne non conforme » lorsqu'un nombre de la chaîne a dépassé 100, et la longueur de la chaîne est le nombre de cellules contenant des nombres dans cette chaîne.

La plage est sélectionnée à l'aide de la souris par exemple.

Tous les maillons de la chaîne apparaissent très facilement, et c'est l'avantage du tableur. C'est un peu plus pénible de faire afficher toutes les valeurs d'une chaîne sur la calculatrice.

On se contente dans le programme suivant de calculer sa longueur éventuelle.

- Sur la T.I. :

```
program : Syracuse 4
Input « Entier de départ ? », A
1 → C
  While A ≠ 1 et A < 101
    If int(A/2)=A/2
      Then A/2 → A
    Else 3A + 1 → A
    End
  C + 1 → C
  End
If A = 1
Then Disp C
Else Disp « Chaîne cassée »
End
```

- Pour faire apparaître toutes les valeurs de la chaîne, on crée une liste :

```
program : Syracuse 5
ClrList L1
Input « Entier de départ ? », A
A → L1(1)
1 → C
  While A ≠ 1 et A < 101
    If int(A/2)=A/2
      Then A/2 → A
    Else 3A + 1 → A
    End
  C + 1 → C
  A → L1(C)
```

```

End
If A = 1
Then Disp C
Else Disp « Chaîne cassée »
End

```

Les différents maillons de la chaîne sont ainsi stockés dans la liste L1, en position C à chaque nouveau passage dans la boucle. On efface la liste précédente à chaque exécution du programme.

– Sur Scratch, il n'y a pas l'instruction « tant que », mais « répéter jusqu'à ».

C'est l'occasion de travailler la logique avec nos élèves : le contraire de « A et B » est « non A ou non B ».

Un exemple :

```

quand pressé
supprimer tout de chaîne
demander Quel est le nombre de départ et attendre
à depart attribuer réponse
à a attribuer depart
répéter jusqu'à a = 1 ou a > 100
ajouter a à chaîne
si a mod 2 = 0
à a attribuer a / 2
sinon
à a attribuer a * 3 + 1
ajouter a à chaîne
si a > 100
dire chaîne non conforme
arrêter le script

```

Voilà un exemple du résultat produit :

```

depart 17
a 1

```



chaîne	
1	17
2	52
3	26
4	13
5	40
6	20
7	10
8	5
9	16
10	8
11	4
12	2
13	1

+longueur: 13

Attention : la longueur de la chaîne indiquée en bas de liste n'est pas celle attendue dans cet exercice : c'est le nombre de cellules dans le tableau créé !

Deux exemples :

depart 12
a 1



chaîne
1 12
2 6
3 3
4 10
5 5
6 16
7 8
8 4
9 2
10 1
11
12
+longueur: 12

depart 57
a 172



chaîne non conforme

chaîne
1 57
2 172
+longueur: 2

Pour calculer la longueur éventuelle d'une chaîne :

```

quand pressé
supprimer tout de chaîne
demander Quel est le nombre de départ et attendre
à depart attribuer réponse
à a attribuer depart
à longueur attribuer 1
répéter jusqu'à a = 1 ou a > 100
ajouter a à chaîne
à longueur attribuer longueur + 1
si a mod 2 = 0
à a attribuer a / 2
sinon
à a attribuer a * 3 + 1
ajouter a à chaîne
si a > 100
dire chaîne non conforme
sinon
dire La longueur de la chaîne est pendant 1 secondes
dire longueur
arrêter le script
  
```

3. Il ne reste plus qu' à réaliser un algorithme permettant de créer toutes les chaînes pour un entier de départ de 1 à 100, et de repérer la chaîne la plus longue :

- Sur tableur, c'est très rapide : par recopie vers la droite, puis vers le bas, toutes les longueurs éventuelles apparaissent en ligne : on demande l'affichage du maximum de cette ligne : = MAX(plage).
- On peut faire une mise en forme conditionnelle sur ce maximum pour le repérer plus facilement sur la ligne, ou utiliser la fonction « recherche » pour retrouver l'entier de départ donnant la chaîne la plus longue (voir document joint).
- En langage usuel :

```

à L attribuer 0
à D attribuer 0
Pour B variant de 1 à 100 par pas de 1 faire :
  à A attribuer B
  à C attribuer 1
  Tant que A ≠ 1 et A < 101 faire :
    Si A est pair
      Alors
        à A attribuer A / 2
    Sinon
      à A attribuer 3A + 1
  
```

```

        à C attribuer C + 1
    Fin tant que
Si A = 1
Alors
    Si C > L
    Alors
        à L attribuer C
        à D attribuer B
    Fin de boucle
Dire « La chaîne la plus longue débute à »
Afficher D
Dire « sa longueur est »
Afficher L

```

Et voilà maintenant la boucle « Pour *variable* variant de ... à ... ».

Elle est utilisée de préférence à l'instruction « tant que » lorsqu'on connaît d'avance le nombre de passages dans la boucle.

Ici, on répète 100 fois le programme précédent, pour des entiers n de départ variant automatiquement de 1 à 100.

La variable C compte la longueur de chaque chaîne, elle est donc initialisée à 1 à chaque nouvelle incrémentation de B. La variable L recueille progressivement la longueur de la chaîne la plus longue. Elle est écrasée chaque fois qu'une chaîne plus longue est découverte.

De même, la variable D recueille progressivement le nombre de départ donnant la chaîne la plus longue, et est écrasée aussi souvent que nécessaire.

L'affichage de ces deux variables en fin de programme donne la solution du problème.

– Sur la T.I. :

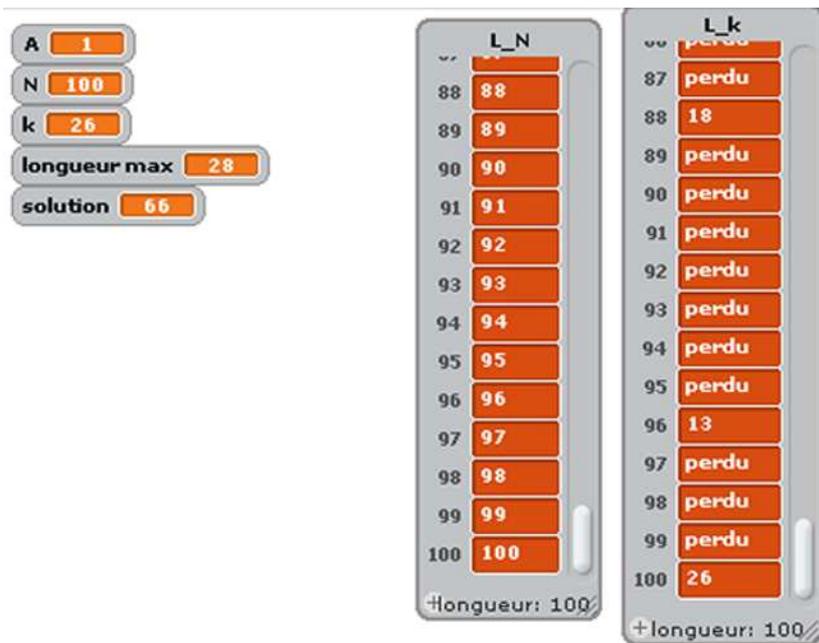
```

program : Syracuse 6
0 → L
0 → D
    For(B,1,100,1)
        B → A
        1 → C
        While A ≠ 1 et A < 101
            If int(A/2)=A/2
            Then A/2 → A
            Else 3A + 1 → A
            End
            C + 1 → C
        End
        If A = 1
        Then
            If C > L
            Then
                C → L
                B → D
            End
        End
    End
    Disp « La chaîne la plus longue débute à », D
    Disp « Sa longueur est », L

```

– Sur Scratch.

On obtient :



avec l'algorithme suivant :

```

quand pressé
supprimer tout de L_N
supprimer tout de L_k
à N attribuer 0
à longueur max attribuer 0
répéter 100 fois
  à N attribuer N + 1
  à A attribuer N
  à k attribuer 1
  répéter jusqu'à A = 1 ou A > 100
    si A mod 2 = 0
      à A attribuer A / 2
    sinon
      à A attribuer 3 * A + 1
      à k attribuer k + 1
  ajouter N à L_N
  si A > 100
    ajouter perdu à L_k
  sinon
    ajouter k à L_k
    si k > longueur max
      à longueur max attribuer k
      à solution attribuer N
arrêter le script
  
```