

Nombres premiers et décomposition en facteurs premiers avec Scratch



A12
A13
INFO
Fiche
professeur

Exercice 1 Nombre premier ou pas ?

Plusieurs corrections possibles.

PROPOSITION 1 de CORRECTION

On teste la divisibilité du nombre par un diviseur potentiel qui augmente de 1 à chaque tour, et qui est compris entre 2 et (nombre - 1), et on stoppe tout dès qu'on en trouve un.

On pourrait aussi dans le même ordre d'idée créer un programme qui teste pour des diviseurs potentiels compris entre 2 et (racine du nombre) pour minimiser le nombre de calculs.

variables à créer : *nombre*, *k*

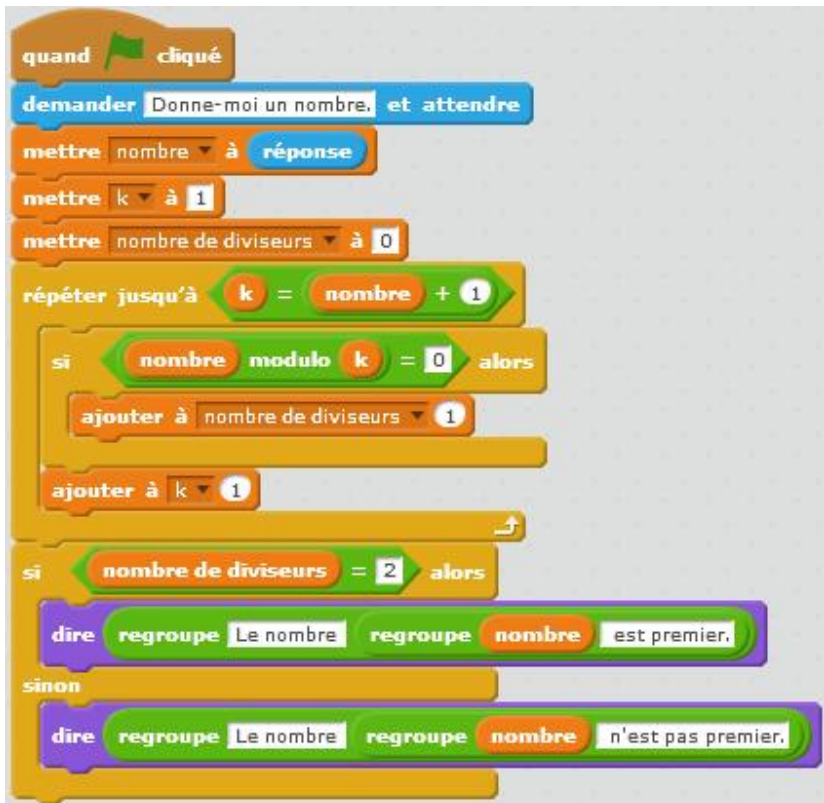
```
quand cliqué
demander "Donne-moi un nombre." et attendre
mettre nombre à réponse
mettre k à 2
si nombre = 2 alors
  dire "Le nombre " nombre " est premier."
sinon
  répéter jusqu'à k = nombre - 1
  si nombre modulo k = 0 alors
    dire "Le nombre " nombre " n'est pas premier."
    stop ce script
  sinon
    ajouter à k 1
  dire "Le nombre " nombre " est premier."
```

PROPOSITION 2 de CORRECTION

On compte le nombre de diviseurs du nombre, si il y a exactement 2 diviseurs, ce nombre est premier, sinon il ne l'est pas.

Ma préférence va à ce programme, puisqu'on colle vraiment à la définition. De plus on peut s'en servir pour la suite.

variables à créer : *nombre*, *k*, *nombre de diviseurs*



Remarque : On doit aller jusqu'à $k = \text{nombre} + 1$, car cela correspond à « pour $k < \text{nombre} + 1$ », et donc « pour $k \leq \text{nombre}$ » car on travaille avec des entiers.

CORRECTION

2) a)



2) b)



2) c)

```
quand cliqué
demander "Donne-moi un nombre." et attendre
mettre nombre à réponse
mettre k à 1
mettre nombre de diviseurs à 0
répéter jusqu'à k = nombre + 1
  si nombre modulo k = 0 alors
    ajouter à nombre de diviseurs 1
    ajouter à k 1
```

2) d) Bien sûr on n'attend pas nécessairement des élèves qu'ils utilisent « regroupe ».

```
quand cliqué
demander "Donne-moi un nombre." et attendre
mettre nombre à réponse
mettre k à 1
mettre nombre de diviseurs à 0
répéter jusqu'à k = nombre + 1
  si nombre modulo k = 0 alors
    ajouter à nombre de diviseurs 1
    ajouter à k 1
si nombre de diviseurs = 2 alors
  dire regroupe "Le nombre " regroupe nombre " est premier."
sinon
  dire regroupe "Le nombre " regroupe nombre " n'est pas premier."
```

Exercice 2 Liste des premiers nombres premiers

1) Liste des nombres premiers inférieurs à 100.

variables à créer : *k*, *nombre*, *nombre de diviseurs*

liste à créer : *nombres premiers*



2) Liste des nombres premiers inférieurs ou égaux à un nombre donné.

variables à créer : *k*, *nombre*, *nombre de diviseurs*

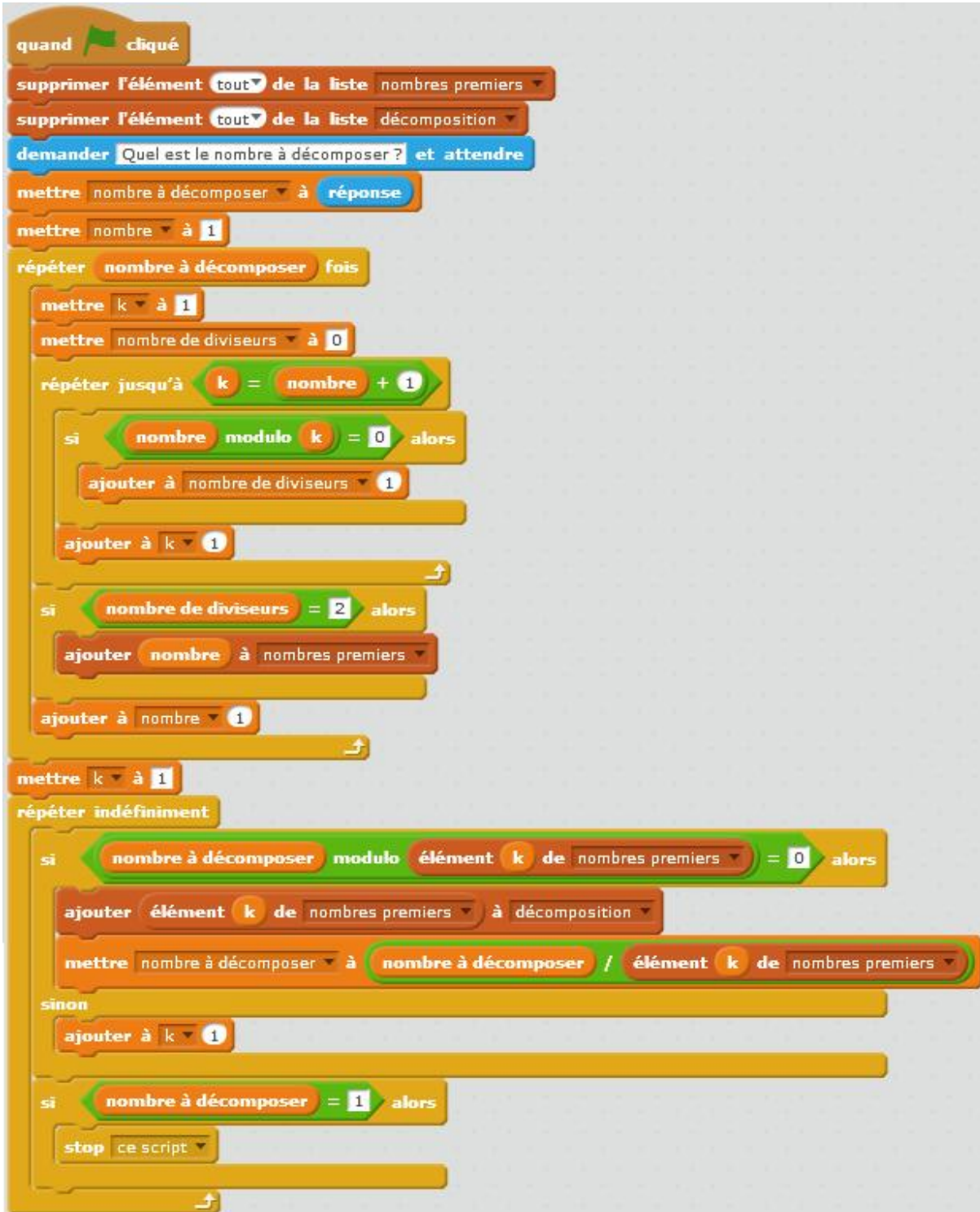
liste à créer : *nombres premiers*



Exercice 3 Décomposition en produit de facteurs premiers

variables à créer : *k*, *nombre*, *nombre de diviseurs*, *nombre à décomposer*

liste à créer : *nombre premiers*, *décomposition*



```

quand cliqué
  supprimer l'élément tout de la liste nombres premiers
  supprimer l'élément tout de la liste décomposition
  demander Quel est le nombre à décomposer ? et attendre
  mettre nombre à décomposer à réponse
  mettre nombre à 1
  répéter nombre à décomposer fois
    mettre k à 1
    mettre nombre de diviseurs à 0
    répéter jusqu'à k = nombre + 1
      si nombre modulo k = 0 alors
        ajouter à nombre de diviseurs 1
        ajouter à k 1
    si nombre de diviseurs = 2 alors
      ajouter nombre à nombres premiers
      ajouter à nombre 1
  mettre k à 1
  répéter indéfiniment
    si nombre à décomposer modulo élément k de nombres premiers = 0 alors
      ajouter élément k de nombres premiers à décomposition
      mettre nombre à décomposer à nombre à décomposer / élément k de nombres premiers
    sinon
      ajouter à k 1
    si nombre à décomposer = 1 alors
      stop ce script

```

The image shows a Scratch script for prime factorization. It starts with a 'when clicked' event, clearing two lists: 'nombres premiers' and 'décomposition'. It then asks the user for a number to decompose and stores it in a variable 'nombre à décomposer'. A loop 'répéter nombre à décomposer fois' is used to find prime factors. Inside this loop, 'k' is set to 1 and 'nombre de diviseurs' is set to 0. A 'répéter jusqu'à k = nombre + 1' loop checks for divisibility. If 'nombre modulo k = 0', the number of divisors is incremented and 'k' is incremented. If the number of divisors is 2, the number is added to the 'nombres premiers' list and the number is decremented by 1. After the loop, 'k' is reset to 1. An 'indéfiniment' loop then checks if the remaining number is divisible by any element in the 'nombres premiers' list. If yes, it adds the element to the 'décomposition' list and divides the number by that element. If no, 'k' is incremented. When the number to decompose reaches 1, the script stops.

Exemple avec 54 :

nombres premiers	
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19

+ longueur: 16

décomposition	
1	2
2	3
3	3
4	3

+ longueur: 4



Remarque concernant ce programme : il est long en calculs, on pourrait se contenter de lister les nombres premiers jusqu'à la racine du nombre à décomposer.

répéter nombre à décomposer fois

à remplacer par :

répéter plafond de racine de nombre à décomposer fois

Exemple avec 54 :

nombres premiers	
1	2
2	3
3	5
4	7

+ longueur: 4

décomposition	
1	2
2	3
3	3
4	3

+ longueur: 4



Exemple avec le 10 890 du cours :

nombres premiers	
1	2
2	3
3	5
4	7
5	11
6	13
7	17
8	19

+ longueur: 27

décomposition	
1	2
2	3
3	3
4	5
5	11
6	11

+ longueur: 6



(Alors qu'avec le programme précédent, c'était beaucoup plus long de chercher tous les nombres premiers inférieurs à 10 890, là la longueur de la liste *nombres premiers* n'est que 27.)