# Thème : Les nombres premiers

## Activité 2. Algorithmes sur les nombres premiers (3 exercices)

**Exercice 1 : *Tester la primalité (identique à l’exercice 3 de l’activité 1 « Autour des nombres premiers »)***

Prérequis : Diviseurs, nombres premiers.

Objectifs : Test de primalité et sa programmation. Amélioration d’un algorithme.

1. A l'aide de divisions successives, trouver les nombres premiers parmi : 107, 227, 375, 377, 379, 571.
2. Recommencer la question précédente en réduisant le nombre de tests. Expliquer votre démarche. Avez-vous l'impression de pouvoir être encore plus "économe" ? Comment peut-on minimiser le nombre de tests ?
3. On considère l'algorithme suivant :

Déclaration des variables : *N* et *I* sont des entiers

Début de l'algorithme

Saisir *N*

Si

Alors

Si

Alors

Afficher "*N* PREMIER"

Sinon

Afficher "*N* NON PREMIER"

Fin Si

Sinon

*I* prend la valeur 2

Tant que (*I* ne divise pas *N*) et (*I* E)

*I* prend la valeur *I* + 1

Fin Tant que

Si *I* divise *N*

Alors

Afficher "*N* NON PREMIER"

Sinon

Afficher "*N* PREMIER"

Fin Si

Fin Si

Fin de l'algorithme

***Note :***

La fonction  **est la fonction partie entière :**

A tout réel elle associe l’unique entier relatif qui lui est immédiatement inférieur ou égal.

Exemples :

; ; .

* Sur les calculatrices TI, il s’agit de la commande **partEnt** - ou **int** pour les calculatrices en anglais - (touche math, menu MATH)
* Sur les Casio, il s’agit de la commande **Intg** (touche OPTN, menu NUM).

Faire fonctionner sur le papier cet algorithme pour et . Qu'affiche-t-il ?

1. a) Programmer cet algorithme sur la calculatrice (Nommer ce programme TESTA).
   1. Vérifier que le programme fonctionne pour et puis l'utiliser pour dire parmi les nombres suivants lesquels sont premiers : 2 011 ; 2 013 ; 2 015 ; 2 017 et 2 019.
   2. Chronométrer le temps que met le programme TESTA pour tester la primalité de
2. a) Modifier l'algorithme précédent pour tester la division par 2 et pour ensuite ne pas tester la division par tous les autres nombres pairs.
   1. Programmer cet algorithme sur la calculatrice (Nommer ce programme TESTB).
   2. Vérifier que le programme fonctionne correctement pour , et pour pair.
   3. Chronométrer le temps que met le programme TESTB pour tester la primalité de

**Exercice 2 : *Liste des nombres premiers inférieurs à 1000***

Pré requis : Avoir le programme TESTB de test de la primalité de l’exercice 1.

Objectif : Créer un algorithme qui crée la liste des nombres premiers inférieurs ou égaux à 1009. Créer le programme LISTPREM sur la calculatrice.

# Principe

* On utilise une nouvelle liste qui sera la Liste6 (cela laisse libres les Listes 1, 2, 3… souvent utilisées pour des calculs courants). On utilisera une version aménagée en sous-programme du programme TESTB pour tester si un entier naturel est premier.
* On commence avec . Si est premier

Alors on écrit dans la liste Liste6 (à la ligne )

Comme est premier on écrit donc à la première ligne de la liste ce qui donne Liste6[1] = 2

* On incrémente[[1]](#footnote-1) de . On a alors

Si est premier

Alors on écrit dans la liste Liste6 (à la ligne )

Comme est premier on écrit donc à la deuxième ligne de la liste ce qui donne Liste6[2] = 3

* On incrémente de . On a alors

Si est premier

Comme n’est pas premier on n’écrit donc rien dans la liste Liste6.

* On incrémente de . On a alors

Si est premier

Alors on écrit dans la liste Liste6 (à la ligne )

Comme est premier on écrit donc à la troisième ligne de la liste Liste6 qui donne Liste6[3] = 5

* On réitère cette démarche tant que

Après l’exécution du programme, on aura dans la liste Liste6 les premiers nombres premiers :

|  |
| --- |
| Liste6 |
| 2 |
| 3 |
| 5 |
| 7 |
| 11 |

# Modification du programme TESTB (renommé PREM)

On modifie le programme de test de primalité TESTB car on va en avoir besoin comme sous-programme dans le nouvel programme LISTPREM

|  |  |
| --- | --- |
| **Procédure sur calculatrice TI** | **Procédure sur calculatrice Casio** |
| * Créer un nouveau programme PREM * Y copier le programme TESTB. Pour cela :   Le curseur étant sur la première ligne (juste après **:** )  2nd sto→  Prgm EXEC TESTB  Entrée Entrée   * Enlever avec la touche suppr la première ligne Prompt N * Dans le programme PREM,   + Remplacer les trois lignes **Disp « N NON PREMIER »** par **0 P** (touche annul pour effacer la ligne sans la supprimer)   + Remplacer les deux lignes **Disp « N PREMIER »** par **1 P**   + Ecrire sur la dernière ligne : **Return** (se trouve dans prgm CTL) | Renommer le programme TESTB en PREM   * Enlever avec la touche DEL la première ligne ‘‘N=’’?→N * Dans le programme PREM,   + Remplacer les trois lignes **‘‘N NON PREMIER’’**◢ par **0 P** (touche annul pour effacer la ligne sans la supprimer)   + Remplacer les deux lignes **Disp « N PREMIER »** par **1 P**   Ecrire sur la dernière ligne : **Return** (se trouve dans prgm CTL) |

Ainsi, le programme PREM reçoit un entier et renvoie la variable qui vaut lorsque est premier et lorsque est composé.

PREM sur TI

If N ≤ 3

Then

If N=2 ou N=3

Then

1P

Else

0P

End

Else

If partEnt(N/2)=N/2

Then

0P

Else

3I

While (partEnt(N/I)≠N/I) et (I≤ partEnt(√(N)))

I+2I

End

If partEnt(N/I)=N/I

Then

0P

Else

1P

End

End

Return

PREM sur Casio

If N≤ 3

Then

If N=2 Or N=3

Then

1P

Else

0P

IfEnd

Else

If Intg(N÷2)=N÷2

Then

0P

Else

3→I

While (Intg(N÷I)≠N÷I) And (I≤ Intg(√(N)))

I+2I

WhileEnd

If Intg(N÷I)=N÷I

Then

0P

Else

1P

IfEnd

IfEnd

Return

# Algorithme LISTPREM

Effacer la liste Liste6

reçoit

reçoit

**Tant que** **faire**

Appel du sous-programme PREM

**Si**

**Alors** Liste6[*I*] reçoit

reçoit

**FinSi**

reçoit

**FinTant que**

# Programmation de la calculatrice

On traduit l’algorithme précédent en langage TI ou Casio.

## Ecriture du programme LISTPREM

Saisir le programme « LISTPREM » sur la calculatrice.

|  |  |
| --- | --- |
| Calculatrice TI | Calculatrice Casio |
| * On choisit L6 dans 2nd listes NOMS * prgmPREM s’écrit en tapant prgm EXEC PREM entrer |  |

LISTPREM sur TI

EffListe L6

2N

1J

While N≤1009

prgmPREM

If P=1

Then

NL6(J)

J+1J

End

N+1N

End

LISTPREM sur Casio

ClrList 6

2N

1J

While N≤1009

Prog ”PREM”

If P=1

Then

NList 6[J]

J+1J

IfEnd

N+1N

WhileEnd

ClrText

## Utilisation du programme LISTPREM

1. Exécutez-le en le chronométrant.
2. Vérifiez que vous avez obtenu dans la liste Liste6 les nombres premiers inférieurs ou égaux à en allant dans le menu Statistiques
3. On note le nombre de nombres premiers inférieurs ou égaux à . Utilisez la liste Liste6 des nombres premiers inférieurs ou égaux à pour compléter le tableau ci-dessous :

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000 |
|  |  |  |  |  |  |  |  |  |  |  |

Peut-on parler d'équi-répartition des nombres premiers ? Justifier votre réponse.

**Exercice 3 : *Recherche des diviseurs premiers inférieurs ou égaux à 1000***

Pré requis : Avoir la liste des nombres premiers inférieurs à 1000 de l’exercice 2.

Objectif : Obtenir un algorithme qui donne la liste des diviseurs premiers inférieurs à 1000 d’un entier naturel non nul quelconque.

# Principe

* On donne un entier naturel non nul .
* On cherche dans la liste des nombres premiers inférieurs ou égaux à ceux qui divisent .
* On commence par créer la liste dans la calculatrice des nombres premiers de à à l’aide du programme LISTPREM (voir l’exercice 2).

# Algorithme DIVISPRE

Saisir

reçoit

reçoit

Tant que et faire

reçoit l’élément *I* de la liste Liste6

Si *N* est divisible par *P*

Alors

Afficher « Divisible par », P

Fin Si

reçoit

FinTant que

# Programmation de la calculatrice

*Travail préalable : Exécutez le programme LISTPREM et assurez-vous que la liste Liste6 contient tous les nombres premiers de à .*

1. Ecrivez le programme « DIVISPRE » dans votre calculatrice.
2. Exécutez-le en cherchant les diviseurs premiers de  
    ;  ;  ; .
3. Quels sont les nombres dont le programme donnera tous les diviseurs premiers ? Testez sur un exemple.
4. Quels sont les nombres dont le programme donnera certains diviseurs premiers mais pas tous ? Testez sur un exemple.
5. Quels sont les nombres dont le programme ne donnera aucun diviseur premier ? Testez sur un exemple.

1. **Incrémenter** une variable, c’est lui ajouter une quantité constante [↑](#footnote-ref-1)