

Ecrit 2. Problème 1, question 5

La question 5 du problème 1 de la deuxième composition du CAPES de mathématiques a pour objectif l'écriture de quelques algorithmes classiques de l'arithmétique. Ce document propose quelques solutions réalisées ici avec le logiciel nSpire CAS.

1. L'énoncé¹ de la question 5.

Pour les algorithmes demandés, on utilisera uniquement les opérations \times , $+$, \wedge et la fonction de deux variables « reste » où $\text{reste}(a, b)$ donne le reste de la division euclidienne de a par b pour a entier naturel et b entier strictement positif. On pourra également utiliser des boucles de type « **for** », « **while** » et la construction « **if...then...else...** »

On précisera le logiciel de calcul formel ou le modèle de calculatrice utilisé.

5.1. Écrire une procédure $\text{Test}(,)$ ayant comme arguments deux entiers naturels k et n avec $n > 1$ affichant "1" si \bar{k} appartient à I_n et "0" sinon.

5.2. Écrire une procédure $\text{Card}()$ ayant comme argument un entier n avec $n > 1$ affichant le cardinal de I_n .

5.3. Écrire une procédure $\text{Ord}(,)$ ayant comme arguments deux entiers naturels k et n avec $n > 1$ affichant la valeur de $\varpi(\bar{k})$, l'ordre de \bar{k} dans (I_n, \times) , si \bar{k} appartient à I_n et "Erreur" sinon.

2. Question 5.1. Exemples de fonctions « Test »

La capture d'écran ci-dessous présente plusieurs variantes :

Variante 1. L'écriture très simple de la fonction nommée **test1**, pourvue de deux arguments n et k , tient compte du fait que le texte de l'énoncé accepte l'utilisation de la fonction « **pgcd** ». Une construction « **If...Then...Else...EndIf** » permet d'afficher 1 ou 0 suivant que le PGCD de k et de n est égal à 1 ou non.

Variante 2. La fonction **test2** n'utilise que le calcul du reste (**remain**) de la division euclidienne d'un entier par un autre. Elle reproduit l'algorithme d'Euclide à l'aide d'une liste locale u de trois nombres. Elle est mise à jour à chaque itération (boucle **While ... EndWhile**). À la fin de l'algorithme, le troisième nombre de cette liste est 0, et le deuxième est le PGCD de k et de n . Une construction « **If...Then...Else...EndIf** » permet d'afficher 1 ou 0 suivant la valeur du PGCD.

<code>test1(15,8)</code>	1	"test1" enregistrement effectué
<code>test1(30,24)</code>	0	Define test1 (k,n)=
<code>test2(15,8)</code>	1	Func
<code>test2(30,24)</code>	0	If gcd(k,n)=1 Then
		Return 1
		Else
		Return 0
		Return 0
		"test2" enregistrement effectué
		Define test2 (k,n)=
		Func
		Local u
		{k,n,remain(k,n)} → u
		While u[3]>0
		{u[2],u[3],remain(u[2],u[3])} → u
		EndWhile
		If u[2]=1 Then
		Return 1
		Else
		Return 0
		EndIf
		EndFunc

Variante 3. Pour information, la fonction test3 , définie dans la page de **Calculs** elle-même, utilise l'instruction conditionnelle **when**. Malgré sa simplicité d'écriture, nous ne l'utiliserons pas par la suite car l'instruction **when** ne semble pas être acceptée par le texte du problème.

¹ L'énoncé complet du problème se trouve sur le site du jury du CAPES.

2. Question 5.2. Exemples de fonctions « Card »

Lorsque l'entier n est fixé, quelle que soit la valeur de l'entier k , chacune des fonctions « Test » de l'écran précédent est indicatrice du fait que \bar{k} est ou non un élément inversible. Pour obtenir le cardinal de I_n , il suffit de compter parmi tous les entiers de l'ensemble $\{1; 2; \dots; n-1\}$ combien ont un « Test » égal à 1.

Variante 1 : construction d'une fonction card avec l'éditeur de fonctions.

Sur la droite de la copie d'écran ci-dessous, la fonction **card1** utilise un compteur local u et une boucle « For ... EndFor ». Le compteur u est initialisé à 0. Pour chaque entier k de $\{1; 2; \dots; n-1\}$, l'une des fonctions « test » incrémente le compteur u de la valeur $\text{test}(k, n)$. Ainsi, u reste inchangé lorsque \bar{k} n'est pas inversible, et augmente d'une unité lorsque \bar{k} est inversible.

Variante 2. Le même résultat peut être obtenu dans l'écran de calcul lui-même (fonction **card2**) à l'aide d'une sommation.

Si n est un nombre premier, à l'exemple de 37, le cardinal de I_n est l'entier $n-1$.

The screenshot shows a software interface with two panes. The left pane displays a table of function values:

$\text{card1}(10)$	4
$\text{card2}(10)$	4
$\text{card1}(12)$	4
$\text{card2}(12)$	4
$\text{card1}(37)$	36
$\text{card2}(37)$	36

The right pane shows the code for the function **card1**:

```

Define card1(n)=
Func
Local c,k
0 → c
For k,1,n-1
test1(k,n)+c → c
EndFor
Return c
EndFunc

```

At the top of the left pane, the formula for $\text{card2}(n)$ is given as $\sum_{k=1}^{n-1} \text{test1}(k,n)$.

3. Question 5.3. Un exemple de fonction « Ord »

La notation « ord » étant réservée pour le logiciel, nous nommerons cette fonction « **ord1** », pourvue de deux arguments k et n . Une boucle « If ... EndIf » discrimine les entiers k non premiers avec n . Lorsque k est premier avec n , une boucle « While ... EndWhile » détermine le premier entier v tel que $k^v \equiv 1 (n)$.

Application.

Il est désormais possible d'écrire un petit programme nommé **grin** (groupe I_n) qui dresse, pour une valeur de n donnée, la liste **in** des éléments de I_n ainsi que la liste **ok** de leurs ordres. Si on lance le programme pour les entiers 10 et 12, on obtient la réponse aux questions 3 et 4 de l'énoncé. On conclut que I_n est cyclique pour $n=10$ (les entiers 3 et 7 en sont des générateurs) mais pas pour $n=12$.

The screenshot shows a software interface with two panes. The top pane shows the code for the function **ord1**:

```

Define ord1(k,n)=
Func
Local v,i
If test1(k,n)=0 Then
Return "Erreur"
Else
1 → v
While remain(k^v,n)>1
v+1 → v
EndWhile
Return v
EndIf
EndFunc

```

The bottom pane shows the code for the program **grin**:

```

Define grin(n)=
Prgm
Local i,k
newList(card1(n)) → in
newList(card1(n)) → ok
1 → i
For k,1,n-1
If test1(k,n)=1 Then
k → in[i]
ord1(k,n) → ok[i]
i+1 → i
EndIf
EndFor
Disp in
Disp ok
EndPrgm

```

The left pane displays a table of function values:

$\text{ord1}(3,10)$	4
$\text{ord1}(3,12)$	"Erreur"
$\text{ord1}(5,10)$	"Erreur"
$\text{ord1}(5,12)$	2
$\text{ord1}(5,37)$	36
$\text{ord1}(8,37)$	12

The bottom part of the table shows the output of the **grin** program:

$\text{grin}(10)$	{1,3,7,9}
	{1,4,4,2}
	Terminé
$\text{grin}(12)$	{1,5,7,11}
	{1,2,2,2}
	Terminé