Le plus petit chiffre

On utilise ici des sous-programmes : ils rendent accessibles des situations relativement complexes par le découpage des tâches qu'ils permettent. La même situation devient beaucoup trop lourde à mettre en place avec un logiciel n'autorisant pas les sous-routines (comme c'est le cas pour la version actuelle d'Algobox).

Dans les programmes

Réalisation d'une simulation, probabilité sur un ensemble fini, calculs sur des nombres entiers, boucle, instruction conditionnelle. Découverte d'une croissance exponentielle ("explosion" du temps de calcul).

1 Plus petit chiffre

Écrire la partie traitement de l'algorithme suivant :

Entrée	un entier naturel n
Traitement	
Sortie	le plus petit chiffre de l'écriture décimale de n

Le traduire en machine.

2 Simulation

Soit $j \geqslant 2$ un entier. On tire au hasard un entier s'écrivant avec j chiffres (écriture décimale usuelle). Si le plus petit chiffre est 0, je gagne et si le plus petit chiffre est 2, vous gagnez.

Écrire un programme qui simule n parties et affiche les fréquences de sortie de chacun des chiffres de 0 à 9.

Acceptez vous de jouer à ce jeu ?

3 Dénombrement

Écrire un programme qui affiche les probabilités de sortie de chacun des chiffres de 0 à 9 (qui dénombrera donc le nombre d'apparitions de 0 comme plus petit chiffre d'un entier à j chiffres, le nombre d'apparitions de 1 comme plus petit chiffre d'un entier à j chiffres ...)

4 Mystère

On donne le programme ci-dessous :

Entrée	un entier $j \geqslant 2$

Traitement	initialisation de liste à [0,0,0,0,0,0,0,0,0,0]
	b prend la valeur $10^j - 10^{j-1}$
	Pour k de 0 jusque 9 faire
	    liste[k] prend la valeur $b - (10 - (k+1))^j$
	    b prend la valeur $b - \text{liste}[k]$
	Fin_ Pour

Sortie	Affichage de la liste

Faire le lien avec les questions précédentes. Expliquez le fonctionnement.

5 Somme des plus petits chiffres des entiers à j chiffres

Où l'on contaste qu'un peu de mathématiques économise des décennies de calcul.

1. Écrire la partie traitement puis traduire sur machine :

Entrée	un entier naturel $j \geqslant 2$
Traitement	
Sortie	la somme des plus petits chiffres de tous les entiers à j chiffres

   On se servira d'une boucle parcourant l'ensemble des nombres à j chiffres.

2. Évaluer expérimentalement le temps de calcul pour l'exécution avec l'entrée $j = 4$. En déduire une évaluation du temps de calcul nécessaire à l'exécution avec l'entrée $j = 16$.
3. Proposer une formule permettant le calcul de cette somme beaucoup plus rapidement.

Éléments de réponses – XCAS

1 Plus petit chiffre

Xcas

```
saisir(n):;
si n==0 alors ppc:=0;
sinon
ppc:=9;
tantque n!=0 faire
unite:=irem(n,10);
si unite<ppc alors ppc:=unite;fsi;
n:=iquo(n,10);
ftantque;
fsi:;
afficher(ppc):;
```

Une version avec paramètre pour réutilisation dans les programmes qui suivent :

Xcas

```
pluspetitchiffre(n):={
/* renvoie le plus petit chiffre de l'écriture décimale de l'entier n*/
local ppc,unite;
si n==0 alors return ppc;fsi;
ppc:=9;
tantque n!=0 faire
unite:=irem(n,10);
si unite<ppc alors ppc:=unite;fsi;
n:=iquo(n,10);
ftantque;
return ppc;
}:;
```

Une version récursive :

Xcas

```
pptchif(n):=quand(n<10,n,min(irem(n,10),pptchif(intDiv(n,10))))
```

2 Simulations

Xcas

```
saisir("Entrez la valeur de j :",j):;
saisir("Entrez le nombre n de parties : ", n):;
liste:=[0,0,0,0,0,0,0,0,0,0];
/*liste contiendra les effectifs,
par exemple  liste[2]=cardinal(plus petit chiffre=2) */
pour k de 1 jusque n faire
tirage:=floor(rand(10^(j-1),10^j));/*On tire au hasard un entier à j chiffres*/
ppc:=pluspetitchiffre(tirage);/*on détermine le plus petit chiffre de cet
    entier*/
liste[ppc]:=liste[ppc]+1;/*on incrémente le compteur correspondant*/
fpour:;
liste:=evalf(liste/n,3):;/* 3 est le nombre de digits à l'affichage */
afficher(liste):;
```

3 Dénombrement exact par boucles

Xcas

```
saisir("Entrez la valeur de  j :",j):;
/* compte le nombre d'apparitions de chacun des chiffres
comme plus petit chiffre parmi les entiers à j chiffres*/
liste:=[0,0,0,0,0,0,0,0,0,0]:;
pour k de 10^(j-1) jusque 10^j-1 faire
ppc:=pluspetitchiffre(k);
liste[ppc]:=liste[ppc]+1;
fpour:;
afficher(liste):;
liste:=evalf(liste/(10^j-10^(j-1)),3):;/* 3 est le nombre de digits à l'
    affichage */
afficher(liste):;
```

Quelques résultats :

1. Avec $j=3$:
   liste :[171,217,169,127,91,61,37,19,7,1]
   liste :[0.19,0.241,0.188,0.141,0.101,0.068,0.041,0.021,0.008,0.001]
2. Avec $j=4$:
   liste :[2439,2465,1695,1105,671,369,175,65,15,1]
   liste :[0.271,0.274,0.188,0.123,0.075,0.041,0.019,0.007,0.002,0.0]

4 Programme mystère

Xcas

```
saisir(j):;
liste:=[0,0,0,0,0,0,0,0,0,0]:;
b:=10^j-10^(j-1):;
pour k de 0 jusque 9 faire
liste[k]:=b-(10-(k+1))^j;
b:=b-liste[k];
fpour:;
afficher(liste):;
```

Le programme mystère a le même rôle que le programme précédent de décompte exact.
Pour les entiers à 3 chiffres, on peut dénombrer de la façon suivante (et c'est le principe du programme mystère) :

1. Il existe $10^3 - 10^2 = 9 \times 10 \times 10 = 900$ entiers à trois chiffres.
2. Il en existe $9 \times 9 \times 9$ qui ne contiennent que des chiffres $\geqslant 1$. Le nombre d'entiers à trois chiffres dont le plus petit chiffre est 0 est donc de $900 - 9^3 = 171$.
3. Il existe 8^3 entiers à trois chiffres de plus petit chiffre $\geqslant 2$. Il y a donc $900 - 171 - 8^3 = 217$ entiers de trois chiffres de ppc 1.
4. Il existe 7^3 entiers à trois chiffres de plus petit chiffre $\geqslant 3$. Il y a donc $900 - 171 - 217 - 7^3 = 169$ entiers de trois chiffres de ppc 2.
5. etc

Pour les entiers entre 10^{j-1} et $10^j - 1$, c'est à dire les entiers à j chiffres :

Entiers à j chiffres	Effectif
à j chiffres	$10^j - 10^{j-1} = 10^{j-1} \times 9$
tous les chiffres $\geqslant 1$	9^j
tous les chiffres $\geqslant 2$	8^j
tous les chiffres $\geqslant 3$	7^j
tous les chiffres $\geqslant 4$	6^j
tous les chiffres $\geqslant 5$	5^j
tous les chiffres $\geqslant 6$	4^j
tous les chiffres $\geqslant 7$	3^j
tous les chiffres $\geqslant 8$	2^j
tous les chiffres $\geqslant 9$	1

Plus petit chiffre	Effectif
ppc=0	$9\times10^{j-1}-9^j$
ppc=1	$9\times10^{j-1}-\left(9\times10^{j-1}-9^j\right)-8^j=9^j-8^j$
ppc=2	$9\times10^{j-1}-\left(9\times10^{j-1}-9^j\right)-\left(9^j-8^j\right)-7^j=8^j-7^j$
ppc=3	7^j-6^j
ppc=4	6^j-5^j
ppc=5	5^j-4^j
ppc=6	4^j-3^j
ppc=7	3^j-2^j
ppc=8	2^j-1^j
ppc=9	1

5 Somme des plus petits chiffres des entiers à j chiffres

1. Exemple de programme déterminant la somme :

   Xcas
   ```
   saisir(j):;
   som:=0:;
   pour k de 10^(j-1) jusque 10^j-1 faire
   som:=som+pluspetitchiffre(k);
   fpour:;
   afficher(som):;
   ```

   Ou en utilisant la fonction somme du langage Xcas :

   Xcas
   ```
   s(j):=somme(pluspetitchiffre(k),k,10^(j-1),10^j-1)
   ```

   Pour la somme sur les entiers à 4 chiffres, on a un temps de calcul dépassant déjà 16 s sur ma machine et plus de 2,6 minutes pour les entiers à 5 chiffres...

2. On part sur un temps de une seconde pour l'exécution avec l'entrée $j=4$. Ce temps correspond à peu près à l'exécution de la boucle qui comprend à peu près $10^4-10^3=9\times10^3$ tours.
   Dans l'exécution avec l'entrée $j=16$, la boucle comprend à peu près 9×10^{15} tours, soit 10^{12} fois plus. En considérant que le temps est multiplié à peu près dans les mêmes proportions, on trouve plus de 31 700 ans.
   Avec $j=20$, on multiplie par $\approx10^{16}$ et on obtient un temps de plus de 300 millions d'années.

3. D'après le tableau d'une question précédente, la somme :

$$S=\sum_{n \text{ entier à } j \text{ chiffres}} \text{pluspetitchiffre}(n)$$

est donnée par :

$$\begin{aligned} S &= 0\times\left(9\times 10^{j-1}-9^j\right)+1\times\left(9^j-8^j\right)+2\times\left(8^j-7^j\right)+3\times\left(7^j-6^j\right)+\cdots+8\times\left(2^j-1\right)+9 \\ &= 9^j+8^j+\cdots+2^j+1 \end{aligned}$$

d'où un programme simple de calcul :

Xcas

```
S(j):=somme(k^j,k,1,9)
```