I25H

e3a concours

CONCOURS ENSAM - ESTP - ECRIN - ARCHIMEDE

Epreuve d'Informatique MP

durée 3 heures

L'usage de la calculatrice est interdit

Indiquez en tête de copie ou de chaque exercice le langage utilisé.

1. Calendrier

Écrire la fonction

```
dateValide          données j, m, a : entiers
                    résultat : booléen
```

qui retourne `Vrai` si la date représentée par le triplet j, m, a (jour, mois, année) est une date valide, et `Faux` sinon

Règles de validité d'une date :

- janvier, mars, mai, juillet, août, octobre, décembre ont 31 jours ;
- avril, juin, septembre, novembre ont 30 jours ;
- février a 29 jours si l'année est bissextile, 28 sinon ;
- une année est bissextile si :
  - pour les années séculaires (1900,2000, etc.), elle est divisible par 400,
  - pour les autres années, si elle est divisible par 4 ;
- on se limitera aux dates postérieures au 15 octobre 1582, date de mise en application en France de ces règles (calendrier Grégorien).

Tournez la page S.V.P.

2. Résistance équivalente

Écrire la fonction

```
resistanceEquivalente    données r : liste de réels
                         résultat : réel
```

qui retourne la valeur de la résistance équivalente aux résistances en parallèle dont les valeurs sont contenues dans la liste r. On utilisera si besoin une fonction donnant le nombre d'éléments d'une liste.

3. Sous-suite de trois nombres de somme maximale.

Écrire la fonction

```
indiceTripletMaximal     données lst : liste d'entiers
                         résultat : entier
```

qui retourne l'indice du premier élément du triplet de somme maximale dans la liste de nombres `lst`, supposée de cardinal au moins égal à 3.

Par exemple, pour la liste suivante

liste	**1**	**5**	7	**8**	**6**	**2**	7	**9**	**4**
indice	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*

la fonction retournerait 3, indice du premier élément du triplet (7,8,6) de somme 21.

4. Suite ordonnée des multiples de 2,3 et 5.

Le but de ce problème est de construire la suite ordonnée des entiers de la forme $2^p3^q5^r$, où p, q et r sont des entiers naturels.

Cette suite commence par

liste	**1**	**2**	**3**	**4**	**5**	**6**	**8**	**9**	**10**	**12**	**15**	**16**				
indice	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
				↑ i_5		↑ i_3		↑ i_2								

Principe :

La liste étant en cours de construction, l'élément suivant de la liste sera à choisir parmi les nombres suivants : $2*l[i_2]$, $3*l[i_3]$, $5*l[i_5]$ où i_2 (respectivement i_3, i_5) est l'indice du plus petit élément de la liste n'ayant pas son double (respectivement triple, quintuple) présent dans la liste.

Dans l'exemple, i_2=8, i_3=6 et i_5=4, et l'élément suivant est donc à choisir parmi 2*9=18, 3*6=18 et 5*4=20.

Le plus petit étant 18, c'est l'élément à ajouter à la liste. Les indices concernés par le choix (ici i_2 et i_3) sont à augmenter de 1, ce qui donne le tableau suivant :

liste	**1**	**2**	**3**	**4**	**5**	**6**	**8**	**9**	**10**	**12**	**15**	**16**	**18**			
indice	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*
				↑ i_5			↑ i_3		↑ i_2							

Écrire, selon ce principe, le programme qui construit la liste triée des n premiers nombres de la forme $2^p3^q5^r$.

5. *Suites de Fibonacci*

Une suite de Fibonacci généralisée est définie par

$u_0=a$

$u_1=b$

$u_n=u_{n-2}+u_{n-1}$

Plusieurs méthodes peuvent être envisagées pour calculer le $n^{ème}$ élément de la suite de Fibonacci initialisée par a et b.

1. Méthode récursive simple

On n'utilise pas de variable locale, on se contente de réécrire dans le langage utilisé la définition mathématique.

Écrire la fonction :

```
fibo1      données n, a, b : entiers
           résultat f : entier
```

qui retourne le $n^{ème}$ terme de la suite de Fibonacci initialisée par `a` et `b` en utilisant la méthode récursive.

2. Méthode itérative

On utilise une boucle avec trois variables locales :

- une variable contient l'avant dernier élément de la suite;
- une variable contient le dernier élément;
- une variable `contient` le nouvel élément.

Écrire la fonction :

```
fibo2      données n, a, b : entiers
           résultat f : entier
```

qui retourne le $n^{ème}$ terme de la suite de Fibonacci initialisée par `a` et `b` en utilisant la méthode itérative.

3. La méthode la plus efficace : récursive par matrices

1. Montrer (mathématiquement) que $\begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} a \\ b \end{pmatrix}$

2. Écrire une fonction de multiplication de deux matrices carrées 2×2.

3. Écrire une fonction d'élévation à la puissance n d'une matrice carrée de 2×2 en utilisant le principe d'exponentiation rapide :

$A^0=Id$

$A^{2n}=A^n.A^n$

$A^{2n+1}=A^{2n}.A$

4. Écrire la fonction :

```
fibo3      données n, a, b : entiers
           résultat f : entier
```

qui retourne le $n^{ème}$ terme de la suite de Fibonacci initialisée par `a` et `b` en utilisant la formule $\begin{pmatrix} u_n \\ u_{n+1} \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ 1 & 1 \end{pmatrix}^n \begin{pmatrix} a \\ b \end{pmatrix}$.

5. On estime que si une addition prend une unité de temps, une multiplication en coûte 4 et une division 7. Au niveau du coût arithmétique, comparer les méthodes `fibo2` et `fibo3` pour calculer u_{2003} .

Tournez la page S.V.P.

Exercices de recherche

Pour ces deux problèmes, vous n'êtes pas guidés. Essayez de proposer un algorithme permettant de répondre aux questions posées.

6. *Suites oscillantes*

Soit f une fonction entière à valeur entière définie de [0;10 000] dans [0;10 000] et u la suite définie par :

$u_0=a$ (avec $0 \leq a \leq 10\,000$)

$u_{n+1}=f(u_n)$

1. Montrer (mathématiquement) que la suite $(u_n)_{n\in\mathbb{N}}$ est périodique à partir d'un certain rang
2. Écrire la fonction

```
periode   données f : fonction; a : entier
          résultat p : entier; e : entier
```

qui étant donné une fonction `f` et un entier `a` retourne `p` (valeur de la période de la suite u définie par $u_0=a$ et $u_{n+1}=f(u_n)$) et `e` un élément de cette période.

7. *Plateau maximal*

Écrire la fonction

```
indicePlateauMaximal   données lst : liste d'entiers
                       résultat : entier
```

qui retourne l'indice du premier élément de la plus grande sous-liste constante de `lst`, liste croissante au sens large de nombres entiers.

Dans l'exemple suivant,

liste	1	1	2	2	2	3	3	4	4	4	4	5	5	6	6	7
indice	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*10*	*11*	*12*	*13*	*14*	*15*	*16*

la fonction retournerait 8 (indice de début de la plus longue sous-liste constante)