

1 Algorithmes à reconnaître

Que calculent les algorithmes suivants ? Aidez-vous si nécessaire en faisant une trace sous forme de tableau : chaque colonne correspond à une variable, chaque ligne à une étape de calcul.

Est-ce qu'ils se terminent toujours ? en particulier est-ce qu'on sort toujours de chaque instruction d'itération ?

Pouvez-vous compter combien de fois on passe dans chaque instruction d'itération ?

Algorithme 1

Données : a et b strictement positifs.

```
{entiers a, b ;  
lire a, b ;  
tantque (a ≠ b)  
  si (a>b) a←a-b ; sinon b←b-a ;  
écrire b ;  
}
```

Algorithme 2

Données : vf et nb quelconques.

```
{entiers s, nb, vf, i ;  
lire vf ; lire nb ;  
s←0 ; i←0 ;  
tantque (nb≠vf)  
  { i←i+1 ; s←s+nb ; lire nb ; }  
écrire i,s ;  
}
```

Algorithme 3

Donnée : n supérieur à 4.

```
{entiers n, sprec, nb, s, d ;  
lire n ; sprec←0 ;  
pour (nb←4 ; nb≤n ; nb←nb+1)  
  { s←0 ;  
    pour (d←2 ; d≤nb/2 ; d←d+1)  
      si (nb modulo d=0) s←s+1 ;  
    si (s>sprec)  
      {écrire nb ; sprec←s ;}  
  }  
}
```

2 Fortunes...

A, B et C (qui préfèrent rester anonymes) possèdent respectivement 5, 4 et 3 millions.

Chaque jour :

A, qui a des loisirs, va porter à B la moitié de sa fortune ;

B, qui ne veut pas être en reste, va porter à C la moitié de sa (nouvelle) fortune ;

C, galvanisé par l'exemple, se précipite chez A pour lui donner la moitié de sa (nouvelle) fortune.

Ecrivez un algorithme, puis un programme qui calcule les fortunes respectives chaque soir pendant 100 jours, et imprime les résultats obtenus chaque jour pendant les 10 premiers jours, puis tous les 10 jours ensuite.

Remarque : cet exercice est essentiellement destiné à faire comprendre correctement la notion d'affectation dans un langage impératif.

3 Calcul avec arrondi

Écrivez un algorithme qui calcule $\sum_{n=1}^{10000} (-1)^{(n-1)}/n$ de deux façons :

- une première fois de gauche à droite : $1 - 1/2 + 1/3 \dots$,
- une deuxième fois de droite à gauche : $-1/10000 + 1/9999 - \dots$

Écrivez un programme qui implémente cet algorithme avec des réels de type `double` et qui affiche les résultats avec les 15 chiffres significatifs de ces réels.

Écrivez un autre programme qui implémente cet algorithme avec des réels de type `float` et qui affiche les résultats avec les 7 chiffres significatifs de ces réels.

Que pensez-vous de ces résultats ? (vous pouvez faire des essais avec d'autres valeurs que 10000, plus grandes ou plus petites)

4 Nombres parfaits

Un nombre entier positif est dit *parfait* s'il est égal à la somme de ses diviseurs propres + 1.

Exemples : $6 = 2 \times 3$ et $6 = 2 + 3 + 1$;
 $28 = 2 \times 14 = 4 \times 7$ et $28 = 2 + 14 + 4 + 7 + 1$;

Écrivez un algorithme, puis un programme qui affiche tous les nombres parfaits \leq à une valeur donnée.

5 Nombres amis

Deux nombres entiers positifs sont dits *amis* si chacun est égal à la somme des diviseurs propres de l'autre + 1.

Exemple : 220 et 284 sont amis car :

$$220 = 2 \times 110 = 4 \times 55 = 5 \times 44 = 10 \times 22 = 11 \times 20 \quad \text{et}$$

$$284 = 2 + 110 + 4 + 55 + 5 + 44 + 10 + 22 + 11 + 20 + 1$$

$$284 = 2 \times 142 = 4 \times 71 \quad \text{et} \quad 220 = 2 + 142 + 4 + 71 + 1$$

Écrivez un algorithme, puis un programme qui affiche tous les nombres amis \leq à une valeur donnée (variante : ne pas afficher les nombres parfaits).

6 Comptine

Description

Une *comptine* est une méthode pour désigner aléatoirement une personne dans un groupe.

On dispose d'un texte bizarre à nombre fini p de mots (exemple : am stram gram) et on montre successivement les personnes en récitant chaque mot du texte. Celle sur laquelle le texte se finit est désignée.

On peut recommencer la comptine sur les personnes restantes, et ainsi en désigner une autre, etc.

On peut ainsi "faire sortir" les n personnes dans un ordre qui dépend uniquement du nombre p de mots de la comptine et du nombre n .

Exercice

On veut simuler sur n personnes une comptine à p mots, dans le cas où n et p sont des entiers >0 et <100 .

Pour cela, on prend un tableau d'entiers pour représenter les personnes ; la case d'indice $i-1$ représente la $i^{\text{ème}}$ personne, le contenu de la case est 1 si la personne est présente, 0 si elle est sortie.

Par ailleurs, on remplace le texte de la comptine par un simple comptage jusqu'à p .

Ecrivez un algorithme, puis un programme qui exécute cette simulation et affiche l'ordre dans lequel les personnes sortent.

Ce programme doit permettre de choisir à chaque exécution les valeurs de p et n dans l'intervalle $]0,100[$.

Remarque : prenez le temps de bien réfléchir à l'algorithme ; s'il est bon, le programme complet, avec suffisamment de commentaires et une présentation soignée, ne dépasse pas une cinquantaine de lignes (sans commentaires du tout, on peut même s'en tirer à une trentaine, mais ce n'est pas conseillé...).