



## Problème I - Suites de CILonacci

Les termes de la suite de Fibonacci sont les nombres 1, 1, 2, 3, 5, 8, 13, ... Le terme suivant dans la suite est calculé par sommation des deux termes précédents. Afin de déterminer la suite de Fibonacci, on a donc besoin des 2 premiers termes.

La suite de CILonacci est une généralisation de la suite de Fibonacci. Les termes de la suite de CILonacci sont formés par sommation des  $M$  termes précédents ( $M \geq 2$ ). Afin de déterminer une suite de CILonacci, on a donc besoin des  $M$  premiers termes.

### Exemples

<u>M</u>	<u>Premiers termes</u>	<u>Suite de CILonacci</u>
3	2, 3, 4	2, 3, 4, 9, 16, 29, 54, 99, ...
4	1, 1, 1, 1	1, 1, 1, 1, 4, 7, 13, 25, 49, ...
2	4, 0	4, 0, 4, 4, 8, 12, 20, 42, ...
2	1, 1	1, 1, 2, 3, 5, 8, 13, 21, 34, ... (ce qui représente la suite de Fibonacci)

Ecrivez un programme qui affiche le  $n$ -ième terme d'une suite de CILonacci.

### Entrées et sorties du programme

- ⇒ Le programme lit de l'entrée standard (clavier) le nombre  $M$ , puis les  $M$  premiers termes et ensuite le nombre  $n$ .
- ⇐ Le programme affiche sur la sortie standard (écran) le  $n$ -ième terme de la suite. Les termes sont séparés par un espace.

### Restrictions

$M \in \mathbb{IN}$ ,  $2 \leq M \leq 100$                        $n \in \mathbb{IN}$ ,  $1 \leq n \leq 10000$



Remettez le programme sous le nom CIL.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable CIL.EXE correspondant au programme.

## Problème II - Dates palindromes

Un palindrome est un mot qui ne change pas si on inverse l'ordre de toutes ses lettres. Ecrivez un programme qui détermine si une date est palindrome.

### Exemples

20.02.2002 est une date palindrome (on négligera les deux points)  
12.12.1212 n'est pas une date palindrome

### Entrées et sorties du programme

- ⇒ Le programme lit de l'entrée standard (clavier) un texte dans le format jj.mm.aaaa correspondant à la date à analyser.
- ⇐ Le programme affiche sur la sortie standard (écran) le nombre  
0 si l'entrée ne correspond pas à une date possible (p.ex. 35.14.2001 ou 30.02.2002)  
1 si l'entrée correspond à une date possible mais qui n'est pas un palindrome et  
2 si l'entrée correspond à une date possible qui est un palindrome.

**Remarque**

Ignorez les années bissextiles, c.-à-d. on considère que le février comporte toujours 28 jours.



Remettez le programme sous le nom PALINDRO.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable PALINDRO.EXE correspondant au programme.

**Problème III - Nombres distincts**

On donne N entiers dans un fichier d'entrée. Ecrivez un programme qui détermine le nombre d'entiers distincts contenus dans ce fichier.

**Entrées et sorties du programme**

- ⇒ Le fichier texte INPUT.TXT dont la première ligne contient le nombre d'entiers N, les lignes suivantes les N entiers.
- ⇐ Le programme affiche sur la sortie standard (écran) le nombre d'entiers distincts dans le fichier d'entrée.

**Exemple entrée et sortie**

INPUT.TXT	Sortie standard
10	5
12	
27	
-9	
27	
41	
85	
41	
-9	
12	
27	

**Restrictions**

$N \in \mathbb{IN}, 1 \leq N \leq 10000$   
 Les N entiers se situent dans l'intervalle [-10000, 10000]



Remettez le programme sous le nom NOMBRES.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable NOMBRES.EXE correspondant au programme et le fichier d'entrée INPUT.TXT correspondant à l'exemple ci-dessus.

**Problème IV - CHALLENGE**

Selon les règles du jeu d'échecs, une dame peut battre toute autre figure de l'adversaire qui se trouve dans sa rangée, sa colonne ou ses deux diagonales sous condition qu'il ne se trouve pas d'autre figure entre les deux. Sur l'échiquier suivant, considérons que les figures **a** et **b** appartiennent à l'adversaire, la dame (☺) peut battre toutes les figures **a**, mais pas les figures **b**.

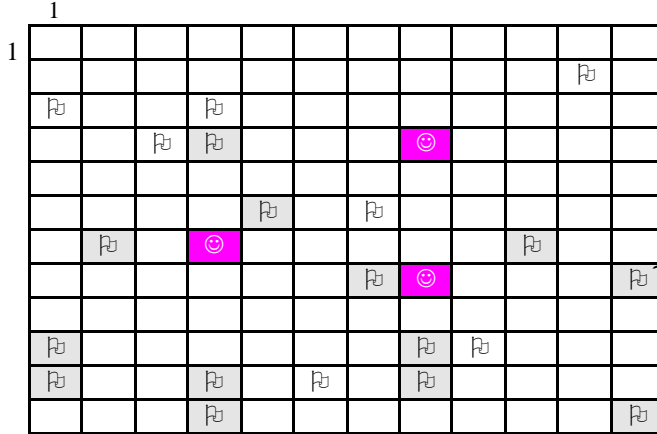
	a						
	b						
	a		☺			a	b
				a			
		b					
a							b
			a		b		

On donne un échiquier carré avec  $N \times N$  cases où sont réparties vos  $P$  dames et  $Q$  figures de l'adversaire. Ecrivez un programme qui détermine le nombre de figures de l'adversaire qui peuvent être battues par au moins une de vos dames. Une figure qui peut être battue par plusieurs dames n'est comptée qu'une seule fois.

**Exemple**

$N = 12, P = 3; Q = 20$

Le symbole ♚ désigne une dame et ♜ une figure de l'adversaire.



cette figure peut être battue par 2 dames

Pour cet exemple, le résultat est 13.

**Entrées et sorties du programme**

- ⇒ Le fichier texte ECHECS.TXT contient dans sa première ligne la taille de l'échiquier  $N$ . La deuxième ligne contient le nombre de dames  $P$  et la troisième ligne contient le nombre de figures de l'adversaire  $Q$ . Les  $P$  lignes suivantes contiennent les positions (sous la forme «ligne» «espace» «colonne») des  $P$  dames et les  $Q$  dernières lignes contiennent les positions des figures de l'adversaire.
- ⇨ Le programme affiche sur la sortie standard le nombre de figures que les  $P$  dames peuvent battre.

**Exemple entrée et sortie**

**ECHECS.TXT**

```

12
3
20
8 4
4 7
8 8
11 2
1 3
4 3
3 4
4 4
5 6
7 6
2 7
10 7
7 8
12 8
1 10
8 10
9 10
1 11
4 11
6 11
8 11
4 12
12 12
    
```

**Sortie standard**

13

**Restrictions**

$N \in \mathbb{IN}, 4 \leq N \leq 100$

$P \in \mathbb{IN}, 1 \leq P \leq N$

$Q \in \mathbb{IN}, 1 \leq Q \leq N \cdot (N-1)$

Vous n'avez pas besoin de vérifier si le fichier ECHECS.TXT est correctement construit: ce fichier ne contient que des coordonnées valables et deux figures ne peuvent pas se trouver sur la même case.



Remettez le programme sous le nom ECHECS.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable ECHECS.EXE correspondant au programme et le fichier d'entrée ECHECS.TXT correspondant à l'exemple ci-dessus.