

Problème J1 : Les carrés

Gigi aime manipuler les carrés. Elle possède une collection de tuiles carrées identiques. Gigi veut disposer ses tuiles sur une table de façon à créer un carré solide, en utilisant certaines tuiles ou toutes les tuiles. Quelle est la dimension du côté du plus grand carré pouvant être construit par Gigi?

Par exemple, quand Gigi a 9 tuiles, elle peut toutes les utiliser pour construire un carré dont la longueur du côté est 3. Mais si elle ne dispose que de 8 tuiles, le plus grand carré qu'elle peut construire a un côté dont la longueur est 2.

Écrivez un programme qui demande à l'utilisateur de décider du nombre de tuiles, puis qui imprime la longueur du côté du plus grand carré. Vous tenez pour acquis que l'utilisateur se limitera aux nombres entiers inférieurs à dix mille. Quand votre programme aura lu l'entrée de l'utilisateur et imprimé le plus grand carré, l'exécution cessera.

Votre programme peut faire appel à plusieurs méthodes pour trouver la réponse. Vous pouvez utiliser la méthode de votre choix. Par exemple, en voici une. D'abord, vérifiez s'il existe assez de tuiles pour construire un carré dont la longueur du côté est 1. Si oui, passez à 2, 3, 4, etc. jusqu'à ce que votre programme découvre une longueur qui est trop grande.

Exemple de session 1

Sortie du programme : Nombre de tuiles?
Entrée de l'utilisateur : 9
Sortie du programme : La longueur du côté du plus grand carré est 3.

Exemple de session 2

Sortie du programme : Nombre de tuiles?
Entrée de l'utilisateur : 8
Sortie du programme : La longueur du côté du plus grand carré est 2.

Exemple de session 3

Sortie du programme : Nombre de tuiles?
Entrée de l'utilisateur : 7535
Sortie du programme : La longueur du côté du plus grand carré est 86.

Problème J2 : Mandat électoral

Dans la ville CS, ville mathématique par excellence, le mandat du maire est de 4 ans, le trésorier est nommé pour une période de 2 ans, le programmeur supérieur est élu pour un mandat de 3 ans et l'agent responsable du contrôle des chiens est remplacé chaque 5 ans.

Cette année, année X , le nouveau maire a annoncé la nomination du nouveau trésorier et du nouvel agent de contrôle des chiens et a félicité le programmeur supérieur de sa récente victoire électorale. C'est-à-dire, une nouvelle personne siège à chacun de ces postes. Cet événement est très rare. Vous allez en calculer la fréquence.

Écrivez un programme qui entre l'année X et l'année Y à venir, et qui présente une liste de toutes les années de X à Y où tous les postes changent.

Exemple de session

Sortie du programme :	Entrez l'année courante :
Entrée de l'utilisateur :	2004
Sortie du programme :	Entrez l'année future :
Entrée de l'utilisateur :	2100
Sortie du programme :	Tous les postes changent en 2004
Sortie du programme :	Tous les postes changent en 2064

Problème J3 : Les comparaisons amusantes

Une comparaison est une combinaison d'un nom et d'un adjectif qui produit des expressions comme *Simple comme bonjour* et *Blanc comme neige*.

Écrivez un programme permettant d'entrer n adjectifs ($1 \leq n \leq 5$) et m noms ($1 \leq m \leq 5$) fournis par le surveillant du concours, puis d'imprimer toutes les comparaisons possibles. Les deux premières lignes d'entrée fourniront les valeurs de n et de m dans cet ordre, suivi, ligne par ligne, des n adjectifs et des m noms. Votre programme peut sortir des comparaisons dans n'importe quel ordre.

Exemple d'entrée :

3
2
Simple
Blanc
Doux
bonjour
neige

Exemple de sortie :

Simple comme bonjour
Simple comme neige
Blanc comme bonjour
Blanc comme neige
Doux comme bonjour
Doux comme neige

Problème J4 : Algorithme de cryptage

Une façon simple de crypter un message est de décaler ou déplacer les lettres. Par exemple, si vous déplacez les lettres dans le message original de 5 places, A dans le message original devient F dans le message crypté ($B \rightarrow G, C \rightarrow H, \dots, T \rightarrow Y, U \rightarrow Z, V \rightarrow A, \dots, Z \rightarrow E$). Pour décoder le message, il suffit de reculer par le même nombre de lettres.

Une façon plus compliquée de crypter fait appel à un mot clé pour déterminer le décalage. Par exemple, si vous utilisez le mot clé ACT pour crypter le message, il faut prendre le message original, y retirer tout sauf les lettres de l'alphabet, et transcrire les lettres dans un bloc dont la largeur est la même que celle du mot clé. Voici un exemple d'un message à crypter :

BANANE & MURE

La version du message en forme de bloc figure ci-dessous, avec le mot clé comme titre.

A	C	T
B	A	N
A	N	E
M	U	R
E		

Maintenant, le message est crypté à l'aide du décalage des lettres. Cependant, cette fois-ci, le décalage n'est pas uniforme. Il dépendra du mot clé qui figure en première ligne du bloc. Si la lettre au haut de la première colonne est un A, les lettres dans cette colonne ne sont pas décalées. Si la lettre est un B, les lettres dans cette colonne sont décalées de 1, et ainsi de suite. Dans l'exemple, les lettres de la troisième colonne sont décalées de 19, puisque T est la 20^e lettre de l'alphabet.

Le message crypté est :

A	C	T
B	C	G
A	P	X
M	W	K
E		

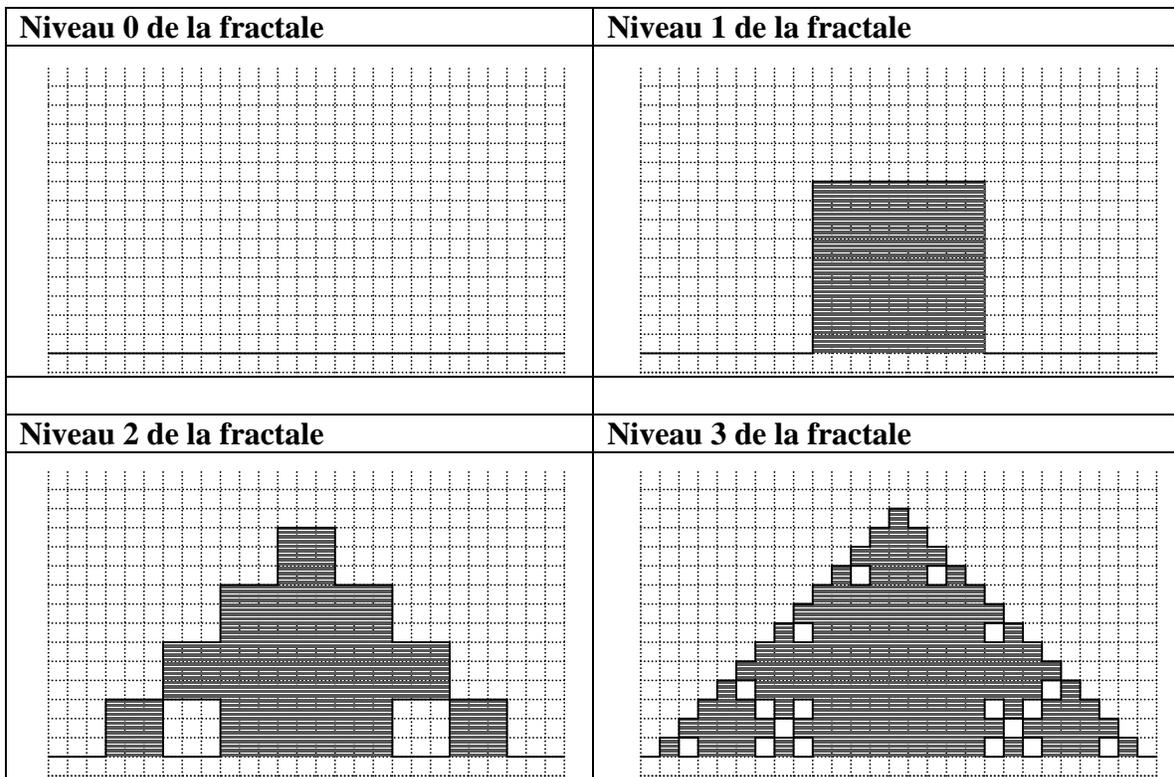
Écrivez un programme qui acceptera un mot clé et une suite de caractères à être cryptée. Le mot clé ne doit jamais contenir plus de 6 caractères. Le message sera toujours entré en lettres majuscules. La longueur totale du message ne doit pas dépasser 60 caractères.



Exemple d'entrée 1	Exemple de sortie 1
ACT	BCGAPXMWKE
BANANE & MURE	
Exemple d'entrée 2	Exemple de sortie 2
TRICKY	BCWXONKFOTKKFZVI
I LOVE PROGRAMMING!	

Problème J5 : Les fractales

Une fractale est une forme géométrique dont la disposition des éléments de la forme complète se répète dans chaque sous-section de la forme. Ci-dessous, une fractale simple. À chaque étape de croissance de la fractale, chaque ligne droite de la fractale est divisée en trois parties égales. La première et la dernière section demeurent droites. La section du milieu comporte une bosse carrée qui a la même hauteur que la largeur de la section du milieu. (Vous voudrez considérer les quatre orientations du segment de la ligne à l'intérieur de la fractale. Selon le segment de ligne, la bosse peut s'orienter vers le haut, le bas, la gauche ou la droite.)



La fractale est dessinée sur un plan cartésien, où $(0, 0)$ se situe dans le coin inférieur gauche. Dans l'exemple ci-dessus, tenez pour acquis que le point inférieur gauche de la fractale est à $(0, 1)$ et le point inférieur droit de la fractale est à $(27, 1)$. Par exemple, le haut de la fractale du niveau 3 est une ligne allant de $(13, 14)$ à $(14, 14)$.

Écrivez un programme qui tiendra compte des points de treillis (points dont les coordonnées sont des entiers) dans une fractale semblable, dont le coin inférieur gauche est à $(0, 1)$. Le programme doit accepter trois entiers comme entrée : le niveau de la fractale, la largeur de la fractale, et une abscisse. Vous pouvez tenir pour acquis que la largeur de la fractale sera une puissance de trois, et qu'elle sera de taille suffisante pour

permettre à chaque coin de la fractale de toucher une intersection représentée par un point de treillis dans le plan cartésien. La largeur ne dépassera jamais 81. L'abscisse, x , se

situera dans l'intervalle $0 \leq x \leq \text{largeur}$. Votre programme doit sortir les valeurs de l'ordonnée, y , des points de treillis (x, y) situés sur des lignes de la fractale.

Vous pouvez dessiner une représentation graphique de la fractale pour fins de débogage (et d'intérêt). Cependant, les essais peuvent vous demander de définir des fractales de dimensions trop grandes pour apparaître sur un seul écran.

Exemple d'entrée 1	Exemple de sortie 1
3 27 5	4 5 6
Exemple d'entrée 2	Exemple de sortie 2
3 27 18	1 2 3 4 7 8 9 10
Exemple d'entrée 3	Exemple de sortie 3
2 27 19	1 4 7
Exemple d'entrée 4	Exemple de sortie 4
4 81 38	37 38 39