

Lëtzebuenger Informatiksolympiad 2020

Qualifikationsronn

Inscription ,déroulement, règlement et dates importantes de la compétition sur

www.infosolympiad.lu



Tous les programmes doivent être réalisés sous forme d'applications console (voir remarques sur le site www.infosolympiad.lu sous la rubrique "Les questionnaires").



Les formats des données ainsi que des résultats représentés dans les exemples d'exécution sont à respecter absolument.



Sous le fichier d'entrée on entend soit l'entrée directe des données via le clavier soit la redirection d'un fichier texte en mode console.



Pour l'évaluation, tous les programmes doivent être soumis au Contest Management System (CMS).

TÂCHE 1

PINCODE

15 POINTS

Description

Malheureusement, Jean a oublié le code d'accès à 4 chiffres de son téléphone portable. La seule chose qu'il se rappelle est que la combinaison des chiffres qui constituent le code représentent une suite arithmétique.

Par définition, une suite arithmétique est une série de nombres dont la différence entre 2 nombres consécutifs est une constante.

Exemples de suites arithmétiques appliquées aux nombres entiers :

1 ; 4 ; 7 ; 10 ; 13 ; 16 ; ... La constante est 3.

2 ; 14 ; 26 ; 38 ; 50 ; 62 ; ... La constante est 12.

Exemples de suites arithmétiques appliquées à 4 chiffres, donc au problème de Jean :

2 ; 4 ; 6 ; 8 La constante est 2.

3 ; 4 ; 5 ; 6 La constante est 1.

9 ; 6 ; 3 ; 0 La constante est -3.

6 ; 6 ; 6 ; 6 La constante est 0.

Il s'agit d'aider Jean à retrouver graduellement son code d'accès.



Tâche

Écrivez un programme qui, en fonction du premier chiffre d'un code d'accès à 4 chiffres, affiche le nombre de combinaisons de chiffres possibles de façon à ce que ces chiffres représentent une suite arithmétique.

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient le premier chiffre du code d'accès.

Sortie

Le programme affiche le nombre de combinaisons possibles.

Exemples d'exécution

Premier exemple

Fichier d'entrée	Résultat
2	3

Explication : les combinaisons possibles sont 2 2 2 2, 2 3 4 5 et 2 4 6 8.

Deuxième exemple

Fichier d'entrée	Résultat
6	4

Explication : les combinaisons possibles sont 6 6 6 6, 6 7 8 9, 6 5 4 3 et 6 4 2 0.

TÂCHE 2

ARMSTRONG

15 POINTS

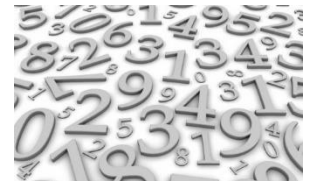
Description

Un nombre narcissique (aussi appelé nombre d'Armstrong) est un entier naturel non nul qui est égal à la somme des puissances n -ièmes de ses chiffres où n désigne le nombre de chiffres de l'entier. Tous les chiffres, à l'exception du zéro, sont des nombres narcissiques.

Exemple :

Le nombre à 3 chiffres 153 est un nombre narcissique puisque

$$1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$$



Tâche

Écrivez un programme qui détermine tous les nombres narcissiques d'un intervalle entier $[a ; b]$ donné.

Contraintes

$a ; b \in \mathbb{N}^*$ avec $a \leq b$ et $1 \leq a ; b \leq 10\,000\,000$

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient 2 entiers naturels séparés par un espace représentant les bornes inférieur et supérieur de l'intervalle à considérer.

Sortie

Le programme affiche tous les nombres narcissiques possibles, un nombre par ligne, en ordre ascendant. S'il n'y a pas de nombre narcissique, le programme affiche le texte « Rien ».

Exemples d'exécution

Premier exemple

Fichier d'entrée	Résultat
100 999	153 370 371 407

Deuxième exemple

Fichier d'entrée	Résultat
1000 100000	1634 8208 9474 54748 92727 93084

Troisième exemple

Fichier d'entrée	Résultat
10 150	Rien

Répartition des points

Sous-tâche	Points	Contraintes supplémentaires
1	10	$1 \leq a ; b \leq 10\,000$
2	5	aucune

Remarques pour la programmation

Vous n'avez pas besoin de tester d'éventuels débordements.

TÂCHE 3

BANDITTI

30 POINTS

Description

Sur un terrain plat muni d'un repère orthonormé se trouvent n bandits, dont les coordonnées (x_i, y_i) sont des nombres entiers. Il s'ensuit une fusillade par étapes successives, d'après les règles suivantes :

À chaque étape, chaque bandit vivant vérifie si parmi tous ses adversaires encore vivants, il existe exactement un qui soit plus proche (à une distance euclidienne plus petite) que tous les autres ; si c'est le cas, il l'abat. Lorsque plusieurs adversaires se trouvent à cette distance minimale, il ne fait rien. Tous les bandits appliquent ce principe simultanément ; ainsi un bandit peut tirer sur quelqu'un et être abattu en même temps, et deux bandits peuvent tirer sur la même cible.

Les étapes se succèdent jusqu'à ce qu'aucun tir ne puisse plus avoir lieu d'après les règles mentionnées.



Exemples

- 1) Si 4 bandits se trouvent aux coordonnées (1, 2), (3, 4), (5, 6), (7, 8), à la première étape le bandit 1 abat le bandit 2 et le bandit 4 abat le bandit 3. Les bandits 2 et 3 ne tirent pas, parce qu'il y a 2 bandits à même distance minimale, à savoir les bandits 1 et 3 pour le bandit 2, et les bandits 2 et 4 pour le bandit 3. À la deuxième étape les survivants 1 et 4 s'entre-tuent. Après ces deux étapes, tous sont morts.
- 2) Si 5 bandits se trouvent aux coordonnées (0, 0), (1, 1), (-1, 1), (-1, -1), (1, -1), le premier, au centre, est abattu (même 4 fois) par tous ses adversaires, tandis qu'il ne tire pas lui-même, car tous ses adversaires se trouvent à la même distance minimale. Après cette première étape, la situation est bloquée, car pour chacun des 4 bandits restants, il y a 2 adversaires à distance minimale. La fusillade se termine donc après une seule étape, avec 4 survivants.

Tâche

Écrire un programme qui, à partir d'une configuration de n bandits donnée, détermine le nombre d'étapes de la fusillade et les survivants.

Contraintes

$n \in \mathbb{N}^*$, $x_i \in \mathbb{Z}$, $y_i \in \mathbb{Z}$ avec $1 \leq n \leq 100$, $-1000 \leq x_i \leq 1000$ et $-1000 \leq y_i \leq 1000$

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient une première ligne avec le nombre n , et ensuite n lignes supplémentaires dont chacune contient les coordonnées x_i et y_i (deux nombres entiers séparés par une espace) du bandit d'indice i ($1 \leq i \leq n$).

Sortie

Le programme affiche à la première ligne le nombre d'étapes de la fusillade et à la deuxième la liste des indices (en ordre croissant) des bandits encore vivants. Si après la fusillade tous sont morts, la deuxième ligne contient simplement une croix (symbole d'addition).

Exemples d'exécution

Premier exemple

Fichier d'entrée	Résultat
4 1 2 3 4 5 6 7 8	2 +

Deuxième exemple

Fichier d'entrée	Résultat
5 0 0 1 1 -1 1 -1 -1 1 -1	1 2 3 4 5

Répartition des points

Sous-tâche	Points	Contraintes supplémentaires
1	10	$1 \leq n \leq 3$
2	10	$1 \leq n \leq 10$ et $y_i = 0$ (tous les bandits sont alignés)
3	10	aucune

Remarques pour la programmation

On peut supposer, sans devoir réaliser des tests d'égalités, que les bandits se trouvent à des emplacements deux à deux distincts.

TÂCHE 4

WINDFARM

40 POINTS

Description

Un fournisseur d'énergie veut construire un nouveau parc éolien dans un champ rectangulaire d'une longueur de n unités et d'une largeur de m unités. Le champ peut donc être modelé comme une région de $n \times m$ cellules de même dimensions.

L'administration communale a donné la permission de construire un tel parc rectangulaire d'une longueur de l cellules et d'une largeur de h cellules. Malheureusement, le fournisseur ne sait pas où est la position optimale pour construire le parc. Il a donc fait des mesures des forces éoliennes pour toutes les cellules, et a constaté qu'il n'y a que k cellules où la force est assez important pour être récupérée. Toutes les autres cellules ont des forces éoliennes négligeables.

L'énergie éolienne qu'un parc éolien peut produire est la somme des forces éolienne des cellules sur lesquelles le parc est placé.



Tâche

Il s'agit de trouver l'énergie éolienne maximale qu'un parc éolien de $l \times h$ cellules peut produire, s'il est construit dans une région de $n \times m$ cellules.

Entrée et sortie du programme

Entrée

La première ligne contient quatre nombres séparés par des espaces : $n \ m \ l \ h$.

La seconde ligne contient un nombre k .

Les k lignes suivants contiennent trois nombres séparés par des espaces.

La $(i+2)^{ième}$ ligne (où $1 \leq i \leq k$) contient les nombres entiers $x_i \ y_i \ w_i$ séparés par des espaces. Ceci veut dire que la force éolienne dans la cellule (x_i, y_i) est d'une magnitude w_i .

Sortie

Un nombre entier indiquant l'énergie éolienne maximale.

Contraintes

$$1 \leq n, m \leq 10^9$$

$$1 \leq l \leq \min(10, n) \text{ et } 1 \leq h \leq \min(10, m)$$

$$0 \leq k \leq 100$$

$$1 \leq x_i \leq n \text{ et } 1 \leq y_i \leq m$$

$$0 \leq w_i \leq 10^9$$

Exemples d'exécution

Premier exemple

Fichier d'entrée	Résultat
4 4 2 2 4 1 1 1 1 2 2 3 3 10 4 2 2	12

Deuxième exemple

Fichier d'entrée	Résultat
3 3 1 1 9 1 1 5 1 2 7 1 3 10 2 1 1 2 2 4 2 3 9 3 1 6 3 2 12 3 3 1	12

Répartition des points

Sous-tâche	Points	Contraintes supplémentaires
1	10	$1 \leq n, m \leq 100$
2	10	$l = h = 1$
3	20	aucune

GOOD LUCK. SEE YOU AT...



**32ND INTERNATIONAL
OLYMPIAD IN INFORMATICS
SINGAPORE**