

Qualifikationsronn

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 sont à respecter absolument.
Sous le fichier d'entrée on entend soit l'entrée directe via le clavier soit la redirection d'un fichier texte en mode console.

TÂCHE 1

ISOGRAMME

Description

Nous considérons qu'un mot est « isogramme en couple » si chaque lettre du mot apparaît exactement deux fois dans ce mot. Par exemple, le mot « papa » est isogramme en couple car chaque lettre apparaît exactement deux fois, tandis que « maman » ne l'est pas, comme la lettre « n » n'apparaît qu'une seule fois.



Votre programme doit pour tout mot donné déterminer si oui ou non il est isogramme en couple.

Entrée et sortie du programme

Entrée

La première ligne du fichier d'entrée contient un nombre entier positif n , indiquant le nombre de mots à traiter. Chacune des n lignes suivantes contient un mot composé de 1 à 52 lettres. On considère uniquement les 26 lettres minuscules de 'a' à 'z'.

Sortie

Pour chaque mot du fichier d'entrée, il faut afficher une ligne résultat. Dans chaque ligne résultat le mot traité est suivi d'un espace, trois tirets et encore un espace et la ligne est terminée par « isogramme en couple » si tel est le cas et par « non isogramme en couple » autrement.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
4	papa --- isogramme en couple
papa	babalala --- non isogramme en couple
babalala	mototama --- isogramme en couple
mototama	maman --- non isogramme en couple
maman	

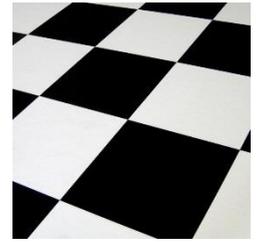
Remettez le programme sous le nom **ISOGRAMME.xxx**, avec **xxx=PAS, C(PP), PY ou JAVA**.

Description

Max doit carrelé un sol rectangulaire avec des carrelages blancs. Afin de casser la monotonie de tout un sol en une seule couleur blanche, il a décidé de rendre l'aspect du sol plus attrayant en le garnissant par des surfaces rectangulaires de couleur noire.

Max vous demande d'écrire un programme qui calcule combien de carrelages noirs et blancs sont nécessaires pour carrelé l'entièreté du sol.

Votre travail consiste à écrire un programme qui réalise cette tâche.

**Entrée et sortie du programme****Entrée**

La première ligne du fichier d'entrée contient trois entiers positifs séparés par des espaces, à savoir dans l'ordre L , W et N , où N , avec $0 < N \leq 100$, représente le nombre de surfaces rectangulaires noires. La largeur et la longueur de la surface à carrelé pourront contenir respectivement L et W carrelages avec $0 < L, W \leq 1000$.

Les N lignes suivantes contiennent chacune quatre entiers positifs séparés par des espaces représentant les coordonnées d'une surface rectangulaire noire. Les deux premiers entiers sont les coordonnées de l'extrémité supérieure gauche du rectangle et les deux derniers les coordonnées de l'extrémité inférieure droite du rectangle.

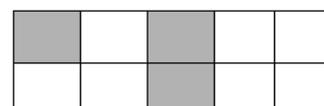
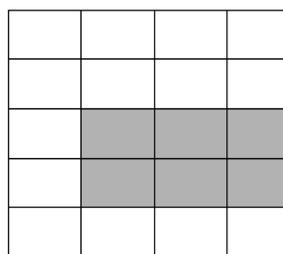
L'exécution du programme s'arrête si L , W et N valent tous les trois 0.

Sortie

Le programme affiche une ligne contenant deux entiers : le nombre de carrelages blancs suivi par le nombre de carrelages noirs. Ces deux nombres sont séparés par un espace.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
5 4 1	14 6
3 2 4 4	7 3
2 5 2	
1 1 1 1	
1 3 2 3	
0 0 0	



Remettez le programme sous le nom CARRELAGES.xxx, avec xxx=PAS, C(PP), PY ou JAVA.

Description

La chambre de Max fourmille d'appareils électriques dont l'utilisation pose parfois des problèmes. En effet, selon les appareils branchés, il se peut que le fusible saute provoquant le crash des ordinateurs.



Afin de remédier à cette situation il vous demande d'écrire un programme lui permettant d'y voir plus clair, c'est-à-dire de pouvoir prédire si les appareils branchés et allumés provoquent oui ou non une coupure de courant.

Entrée et sortie du programme**Entrée**

La première ligne du fichier d'entrée contient trois entiers positifs séparés par des espaces, à savoir dans l'ordre N , K et A avec $0 < N, K, A \leq 100$. L'exécution du programme s'arrête si ces trois valeurs sont nulles.

N représente le nombre d'appareils électriques, K représente le nombre d'opérations effectuées sur ces appareils et A la limite maximale de courant supportée par le fusible.

Les N lignes suivantes contiennent chacune un entier positif représentant la consommation c_i du i -ième appareil. Ces lignes sont suivies par K lignes contenant chacune un entier x avec $1 \leq x \leq N$ représentant le numéro d'un des appareils électriques. Au départ tous les appareils sont éteints. Lorsque le numéro d'un appareil apparaît dans cette liste des K valeurs, il change d'état. Cela signifie que s'il était éteint, il s'allume et s'il était allumé, il s'éteint.

L'exécution du programme s'arrête si N , K et A valent tous les trois 0.

Sortie

Si la simulation indique une surcharge, le programme affiche « Le fusible saute » et dans le cas contraire, il affiche le message « Consommation maximale : xxxA » indiquant la consommation maximale atteinte lors de cette simulation.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
2 4 16 10 7 1 1 1 2 3 4 10 4 5 3 1 2 2 3 0 0 0	Le fusible saute Consommation maximale : 9A

Remettez le programme sous le nom FUSIBLES.xxx, avec xxx=PAS, C(PP), PY ou JAVA.

Description

Tous les ans le comité des élèves du campus scolaire se donne beaucoup de peine pour organiser la fête de fin d'année destinée à tous les élèves de toutes les écoles du campus. Malheureusement, il y a toujours des élèves qui, au lieu de faire la fête avec leurs collègues, préfèrent quitter le campus. Par curiosité on voudrait savoir quel est le nombre le plus élevé d'élèves absents de la même école. Votre programme doit ainsi déterminer et afficher ce nombre.

**Entrée et sortie du programme****Entrée**

La toute première ligne du fichier d'entrée contient le nombre entier positif K , donnant le nombre de jeux de données.

Pour chaque jeu de données (Data Set), la première ligne contient deux nombres entiers positifs c et n . Le nombre c ($1 \leq c \leq 1000$) indique le nombre d'écoles à considérer. Le nombre n ($1 \leq n \leq 10000$) indique le nombre total d'élèves effectivement présents sur le campus le jour de la fête du campus scolaire.

La deuxième ligne de chaque jeu de données contient c nombres $m_1, m_2, \dots, m_i, \dots, m_c$ avec $m_i \geq 0$, m_i représentant le nombre d'élèves de l'école i .

La troisième ligne de chaque jeu de données contient n nombres $b_1, b_2, \dots, b_j, \dots, b_n$, avec $1 \leq b_j \leq c$, b_j indiquant le numéro de l'école du j -ième élève effectivement présent sur le campus le jour de la fête du campus scolaire.

Sortie

Pour chaque jeu de données il faut d'abord afficher le texte « Data Set k : », k étant le numéro du jeu de données, et dans la ligne suivante le nombre maximal d'élèves absents d'une même école.

Après chaque jeu de données on laisse une ligne vide.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
2	Data Set 1:
3 8	2
5 5 2	
1 2 1 2 1 2 1 2	Data Set 2:
5 6	9
10 10 10 10 10	
1 2 4 5 3 1	

Remettez le programme sous le nom CAMPUS.xxx, avec xxx=PAS, C(PP), PY ou JAVA.