Lëtzebuerger Informatiksolympiad 2019

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 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.**

TÂCHE 1 COUPURES

Description

Un employé de guichet de banque vous demande d'écrire un programme qui détermine le nombre de billets de banque qui correspondent à un certain montant d'argent. On considère les coupures pour la devise Euro : 200, 100, 50, 20, 10 et 5. Des grandes coupures sont à favoriser par rapport à des coupures plus petites.

Exemples

Le montant de 1395 Euro correspond aux billets de banque avec les coupures suivantes :

6 × 200 ; 1 × 100 ; 1 × 50 ; 2 × 20 ; 0 × 10 et 1 × 5.

Le montant de 110 Euro correspond aux billets de banque avec les coupures suivantes :

0 × 200 ; 1 × 100 ; 0 × 50 ; 0 × 20 ; 1 × 10 et 0 × 5.

Restrictions

Dans le cadre de cette tâche, on ne considère que des montants d'argent qui sont un multiple de 5 dans l'intervalle entier [5; 30000]. Votre programme n'a pas besoin de tester des débordements.

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient le montant d'argent.

Sortie

Le résultat affiché sur l'écran sont les nombres des billets de banque par coupure correspondants au montant d'argent, une ligne par coupure, par ordre descendant sur la valeur des coupures, sans interlignes. Notez l'espace avant et après le symbole « x ».

Exemple d'exécution

Fichier d'entrée	Résultat
1395	6 × 200 1 × 100 1 × 50 2 × 20 0 × 10 1 × 5

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

TÂCHE 2 — PANGRAMME

Description

Un pangramme (nom composé de deux mots issus du grec ancien : « pan » signifiant « tout » et « gramma » signifiant « lettre ») est une phrase qui comporte chaque lettre de l'alphabet au moins une fois. Une des finalités d'un pangramme était de tester des machines à écrire et des Telex. On l'utilise aussi en typographie pour faire des tests de polices de caractères. L'usage a perduré en informatique, les programmes d'aperçu de polices de caractères faisant également usage de pangrammes.

Il s'agit d'écrire un programme qui détecte si une phrase représente ou non un pangramme.

Dans le cadre cette tâche, on se base sur les 26 lettres de l'alphabet latin, sans lettres accentuées ni ligatures. On ignore la casse des lettres (lettres minuscules ou majuscules), les apostrophes, les symboles de ponctuation ainsi que l'espace.

Voici quelques exemples de pangrammes

Portez ce vieux whisky au juge blond qui fume

The Quick Brown Fox Jumps Over The Lazy Dog

pfgbvudjhakznrwcymiqeltosx

Dans un wagon bleu, tout en mangeant cinq kiwis frais, vous jouez du xylophone.

Restrictions

Dans le cadre de cette tâche, on ne considère que des phrases ayant une longueur entre 1 et 255 lettres. Pour les valeurs possibles de ces dernières, voir description. Votre programme n'a pas besoin de tester des débordements respectivement de vérifier les valeurs des lettres.

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient la phrase à analyser.

Sortie

Le résultat affiché sur l'écran est le texte « OK » si la phrase du fichier d'entrée représente un pangramme ou le texte « NOK » sinon.

Exemples d'exécution

Fichier d'entrée	Résultat
Portez ce vieux whisky au juge blond qui fume	OK

Fichier d'entrée	Résultat
La haine tue toujours, l'amour ne meurt jamais.	NOK

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

TÂCHE 3 — PIRATES DES CARAÏBES

Un chef pirate possède une flotte de N bateaux. Il vous demande d'écrire un programme qui calcule l'état de tous ses bateaux. Les bateaux sont désignés par un numéro i variant de 1 à N. Pour chaque bateau i, il y a deux paramètres : le nombre de fûts de rhum F_i et le nombre de pirates P_i. Si sur un bateau il y a moins de fûts que de pirates, c'est mauvais. S'il y a au moins autant de fûts que de pirates, c'est bon (même si les fûts ne peuvent pas être distribués équitablement parmi les pirates). Si en plus on peut distribuer les fûts équitablement parmi les pirates, c'est même génial. Un nombre égal de fûts et de pirates compte comme génial. Mais si la somme du nombre de fûts et du nombre de pirates est supérieur ou égal à 100, c'est catastrophique puisque le bateau deviendrait trop lourd et pourrait couler. Le programme indique l'état de chaque bateau par un entier : -1 est catastrophique, 0 est mauvais, 1 est bon et 2 est génial.

Restrictions

$N \in \mathbb{N}$ et $1 \leq N < 100$

$F_i \in \mathbb{N}$ et $0 \leq F_i < 100$

$P_i \in \mathbb{N}$ et $0 \leq P_i < 100$

Votre programme n'a pas besoin de tester des débordements.

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient une première ligne avec le nombre N, suivie de N lignes avec chacune les deux nombres F_i et P_i séparés par un espace.

Sortie

Le résultat affiché sur l'écran sont N lignes avec chacune l'état du bateau i.

Exemple d'exécution

Fichier d'entrée	Résultat
5 53 40 21 25 68 34 9 9 48 16	1 0 -1 2 2

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

TÂCHE 4 DEVOIR EN CLASSE

Description

Après avoir corrigé toutes les ***N*** copies d'un devoir en classe, un enseignant aimerait bien porter un peu d'ordre dans toutes ces copies, ceci afin de pouvoir les remettre aux élèves par ordre décroissant sur les notes.

Exemple

Les notes d'un devoir en classe après l'avoir corrigé (***N*** = 8) :

13 49 22 37 1 46 22 58

Les notes du même devoir en classe triées par ordre décroissant :

58 49 46 37 22 22 13 1

Restrictions

Dans le cadre de cette tâche, on ne considère que des notes qui respectent les limites, c.-à-d. qui se situent dans l'intervalle [1; 60]. Votre programme n'a pas besoin de tester des débordements.

$N \in \mathbb{N}$ et $1 \leq N \leq 100$

Entrée et sortie du programme

Entrée

Le fichier d'entrée contient une première ligne avec le nombre de copies ***N***, suivie de ***N*** lignes avec chacune la note d'une copie.

Sortie

Le programme affiche toutes les notes des copies dans le bon ordre, séparées par un espace. Notez l'affichage sur 2 chiffres ainsi que l'alignement à droite des notes.

Exemple d'exécution

Fichier d'entrée	Résultat
8 13 49 22 37 1 46 22 58	58 49 46 37 22 22 13 1

Remarque importante

Le tri des notes doit être effectué explicitement par un algorithme – il n'est donc pas permis d'avoir recours à des fonctions prédéfinies respectivement des automatismes intégrés au langage de programmation.

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