

Concours Informatique Luxembourgeois 2010

Questionnaire de l'Epreuve Demi-Finale (sur 100 points)

9 mars 2010

Problème I – PPPP ET PGPP

30 points

Ecrivez un programme qui trouve et affiche à l'écran le PPPP (abréviation de Plus Petit Produit Possible) ainsi que le PGPP (abréviation de Plus Grand Produit Possible) d'une série de N nombres entiers.

Le PPPP de N nombres entiers est le plus petit nombre que l'on peut créer par le produit de 2; 3; ...; N-1; N facteurs.

Le PGPP de N nombres entiers est le plus grand nombre que l'on peut créer par le produit de 2; 3; ...; N-1; N facteurs.

Chez le PPPP ainsi que le PGPP, le nombre minimal de facteurs est 2, le nombre maximal de facteurs est N.

Exemple

Soit la série des quatre nombres entiers suivants: -5; -2; 7 et 10. Le PPPP de cette série est -350, le PGPD est 700.

Restrictions

- $2 \leq N \leq 10$
- Les nombres entiers sont compris dans l'intervalle [-10; 10]

Entrée du programme

Le fichier texte SERIEIN.TXT correspondant à la série des N nombres entiers sous la forme suivante: la première ligne contient la valeur de N, les N lignes suivantes contiennent les nombres entiers de la série.

Sortie du programme

Ecriture sur la sortie standard (affichage à écran) de deux lignes avec chacune un seul nombre entier: le PPPP et le PGPP recherchés.



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

Problème II - TIMBRES

30 points

L'Office des Timbres désire construire un distributeur automatique de timbres. Le distributeur a un stock inépuisable de timbres de N catégories, dont chaque catégorie a une valeur différente.



Un client peut entrer le prix P de l'envoi dans le distributeur et la machine lui fournit exactement les timbres dont la somme des valeurs est égale au prix P .

Tâche

Écrivez un programme qui lit les N valeurs des catégories de timbres du fichier texte IN.TXT et qui, pour un prix P , détermine et affiche les timbres correspondants.

Restrictions

$1 \leq N \leq 100$	Le nombre de catégories
$1 \leq V_k \leq 200$	La valeur de la catégorie k
$1 \leq P \leq 10000$	Le prix de l'envoi

Entrée du programme

- La première ligne du fichier texte IN.TXT contient un seul nombre entier positif: le nombre de catégories N .
- La deuxième ligne du fichier texte contient un seul nombre entier positif: le prix de l'envoi P .
- Les N prochaines lignes décrivent les différentes catégories, une catégorie par ligne. La k -ième ligne contient un seul nombre entier positif V_k qui représente la valeur de la catégorie k . Les catégories dans le fichier texte sont triées par ordre croissant.

Sortie du programme

Le programme doit écrire sur la sortie standard (affichage à écran) N lignes, une ligne par catégorie. La k -ième ligne contient un seul nombre entier positif: le nombre de fois que le timbre de la valeur V_k est distribué. Le programme doit afficher le message «impossible» lorsqu'il est impossible de distribuer des timbres dont la somme des valeurs est égale au prix P de l'envoi.

Exemple d'exécution 1

Contenu du fichier IN.TXT:	Affichage:
3	0
150	1
2	1
50	
100	

Avec un prix de l'envoi de 150, la machine distribue un timbre d'une valeur de 100 et un timbre d'une valeur de 50 car $1 \times 100 + 1 \times 50 = 150$.

Exemple d'exécution 2

Contenu du fichier IN.TXT:	Affichage:
2	0
50	2
1	
25	

Avec un prix de l'envoi de 50, la machine distribue deux timbres d'une valeur de 25 et zéro timbres d'une valeur de 1 car $2 \times 25 + 0 \times 1 = 50$.

Exemple d'exécution 3

Contenu du fichier IN.TXT:	Affichage:
3	impossible
13	
2	
15	
50	

Avec un prix de l'envoi de 13, la machine ne peut pas distribuer de timbres tel que la somme des valeurs des timbres est égale à 13.



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

Problème III - VENDREDI 13

40 points

Est-ce qu'un vendredi 13 est un événement rare, comme beaucoup de gens le prétendent? Autrement demandé: est-ce que le 13 d'un mois tombe moins souvent sur un vendredi que sur les autres jours de la semaine?



Afin de répondre à cette question, écrivez un programme qui calcule la fréquence que le 13 de chaque mois tombe sur un lundi, mardi, mercredi, jeudi, vendredi, samedi et dimanche sur une période donnée de N années. La période à tester va du premier janvier 1900 au 31 décembre 1900 + (N-1) pour un nombre N d'années données.

Restrictions

$N \in \mathbb{IN}$
 $N \geq 1$
 $N \leq 400$

Voici encore quelques faits, intéressant à savoir, avant de résoudre le problème.

Le premier janvier 1900 fut un lundi.

Les mois suivants ont 30 jours: avril, juin, septembre et novembre. Tous les autres ont 31 jours excepté le mois de février, qui n'en a que 28 (sauf dans les années bissextiles où le février comporte 29 jours).

Une année bissextile est une année qui est divisible par 4. Cette règle n'est pas applicable aux années de siècle. Une année de siècle est bissextile si elle est divisible

par 400, sinon pas. Ainsi l'année 2008 fut bissextile (divisible par 4), 2000 fut bissextile (divisible par 400) et 2100 ne sera pas bissextile (pas divisible par 400).

Attention

Il est interdit d'utiliser des fonctions de date prédéfinies dans le programme source.

Entrée du programme

Lecture de l'entrée standard (clavier) du nombre N.

Sortie du programme

Le fichier texte FRIDAY13.TXT contient une seule ligne, composée de 7 nombres entiers, séparés par un espace. Ces entiers représentent la fréquence que le 13 de chaque mois tombe sur un lundi, mardi, ... et dimanche.



Remettez le programme sous le nom FRIDAY13.xxx, avec xxx=PAS ou C(PP). Remettez également le fichier binaire exécutable FRIDAY13.EXE correspondant au programme.

