Lëtzebuerger Informatiksolympiad 2017

Demi-Finale

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.

TÂCHE 1 — 40 POINTS — POLICE

Description

Max roule sur des routes très longues parsemées de contrôles de police à des endroits fixes. Ces contrôles sont de deux types : contrôle d'alcoolémie et contrôle de vitesse. Il note tous les contrôles rencontrés en route pour les signaler à ses amis. Par curiosité il veut savoir quelle est, sur une route donnée, la distance minimale entre un contrôle d'alcoolémie et un contrôle de vitesse.

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

Entrée et sortie du programme

Entrée

Le fichier d'entrée représente un certain nombre de routes où chacune des routes est décrite par deux lignes consécutives : la première désigne la longueur de la route avec $1 \leq K \leq 1000$ et la deuxième contient K caractères indiquant les positions des différents contrôles de la route. Un caractère peut être un

- 'A' pour signaler un endroit avec contrôle d'alcoolémie ;
- 'V' pour signaler un endroit avec contrôle de vitesse ;
- 'T' pour signaler un endroit où les deux contrôles sont effectués ;
- '–' pour indiquer qu'il n'y a pas de contrôle.

Sur chaque route, il y a au moins un contrôle de vitesse et un contrôle d'alcoolémie.

La fin du fichier d'entrée est indiquée par la valeur $K = 0$.

Sortie

Le programme affiche pour chacune des routes du fichier d'entrée la distance minimale entre un contrôle de vitesse et un contrôle d'alcoolémie.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
4 --AV 3 -T- 8 A---V-A- 0	1 0 2

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

TÂCHE 2 — 20 POINTS — CONTROLES

Description

Les équipes de police aiment bien faire des contrôles à des endroits spécifiques. Leurs préférences peuvent dépendre de nombreux critères, comme par exemple la densité du trafic, la proximité du commissariat, les possibilités de se mettre en retrait pour être bien en cachette, la disponibilité de place pour s'occuper des voitures arrêtées, etc.

Vous devez écrire un programme qui permet de déterminer les endroits les plus adaptés. Chaque équipe définit en un premier temps m critères qui lui sont importants et octroie à chaque critère i un poids x_i (entre -20 et 20) en fonction de son importance. Dans un deuxième temps elle évalue chaque endroit en lui donnant pour chacun des m critères une valeur y_i (entre -20 et 20).

Les meilleurs endroits sont ceux pour lesquels la somme des valeurs pondérées est la plus élevée. Il s'agit donc de maximiser pour chaque endroit l'expression suivante : $\sum_{i=1}^{m}(x_i\ y_i)$.

Entrée et sortie du programme

Entrée

La toute première ligne du fichier contient un entier K désignant le nombre d'équipes, suivi de K jeux de données (sachant que $1 \leq K \leq 1000$) ayant chacun la forme suivante :

- La première ligne de chaque jeu de données contient deux entiers n et m représentant respectivement le nombre d'endroits à évaluer et le nombre de critères à utiliser à cet effet. On sait que $1 \leq n \leq 100$ et $1 \leq m \leq 10$.
- La deuxième ligne de chaque jeu de données contient m nombres entiers $x_1, x_2, \ldots, x_m$ compris entre -20 et +20. Il s'agit des poids que l'équipe concernée accorde à chacun des m critères.
- Les n lignes suivantes contiennent m nombres entiers $y_1, y_2, \ldots, y_m$ compris entre -20 et +20. Chaque nombre y_i donne l'évaluation correspondant au $i^{ème}$ critère pour l'endroit concerné.

Sortie

Pour chacune des *K* équipes il faut écrire d'abord le numéro de l'équipe sous la forme « Equipe i : », *i* étant le numéro d'apparition dans le fichier d'entrée de l'équipe concernée.

Ensuite, il faut afficher dans l'ordre d'apparition les numéros des endroits pour lesquels la somme des valeurs pondérées est égale au maximum.

Les résultats de chaque équipe sont terminés par une ligne vide.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
2 3 2 1 -1 2 1 -1 -5 4 2 3 4 0 3 -1 1 2 1 -1 2 -20 -1 1 10 5 2 0 0	Equipe 1: 2 Equipe 2: 1 2 3

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

TÂCHE 3 — 40 POINTS — MESSAGES

Description

Afin d'éviter les contrôles de la police, un groupe de jeunes gens préfère crypter les messages qu'ils s'envoient mutuellement. Ils utilisent uniquement les caractères majuscules non accentués et se basent sur le carré de Vigenère, qui se présente comme suit :

	A	**B**	**C**	**D**	**E**	**F**	**G**	**H**	**I**	**J**	**K**	**L**	**M**	**N**	**O**	**P**	**Q**	**R**	**S**	**T**	**U**	**V**	**W**	**X**	**Y**	**Z**
A	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
B	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
C	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
D	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
E	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
F	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
G	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
H	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
I	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
J	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
K	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
L	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
M	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
N	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
O	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
P	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
Q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
R	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
S	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
T	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
U	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
V	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
W	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
X	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
Y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
Z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Avant d'échanger des messages, ils se mettent périodiquement d'accord sur une **clé** composée de 1 à 256 caractères majuscules non accentués. Chaque caractère de la clé indique la ligne à utiliser pour l'encryptage d'un caractère : le premier caractère de la clé indique la ligne pour encrypter le premier caractère du message, le deuxième caractère de la clé indique la ligne pour encrypter le deuxième caractère du message, etc. Une fois tous les caractères de la clé utilisés pour l'encryptage, on recommence avec le premier caractère de la clé.

Concrètement, l'encryptage des caractères fonctionne comme ci-dessous.

Supposons que la clé est LIO et que le message à transmettre est BONNECHANCE.

Clé :	L	I	O								
Message :	B	O	N	N	E	C	H	A	N	C	E
Emploi clé :	L	I	O	L	I	O	L	I	O	L	I
Message encrypté :	M	W	B	Y	M	Q	S	I	B	N	M

Pour encrypter la première lettre du message, ici la lettre B, on se sert de la première lettre de la clé, la lettre L. La lettre encryptée est celle qui se trouve à l'intersection de la colonne de B et de la ligne de L, c'est-à-dire la lettre M.

Pour encrypter la deuxième lettre du message, ici la lettre O, on se sert de la deuxième lettre de la clé, la lettre I. La lettre encryptée est celle qui se trouve à l'intersection de la colonne de O et de la ligne de I, c'est-à-dire la lettre W.

Etc.

Pour décrypter la première lettre du message transmis, ici la lettre M, on se sert de la première lettre de la clé, la lettre L. On cherche dans la ligne de L la colonne qui contient la lettre M : c'est la colonne B.

Pour décrypter la deuxième lettre du message transmis, ici la lettre W, on se sert de la deuxième lettre de la clé, la lettre I, et on cherche dans la ligne de I la colonne qui contient la lettre W : c'est la colonne O.

Etc.

Il s'agit d'écrire un programme qui permet d'encrypter et de décrypter un message à l'aide de ce procédé.

Restrictions

La clé comprend au minimum 1 et au maximum 256 caractères, considérant que seules les lettres majuscules non accentuées entre A et Z sont utilisées.

Le corps du message proprement dit comprend au minimum 1 et au maximum 256 caractères, considérant que seules les lettres majuscules non accentuées entre A et Z sont utilisées.

Entrée et sortie du programme

Entrée

La première ligne contient un nombre entier positif N (sachant que $1 \leq N \leq 1000$) indiquant le nombre de messages à traiter.

La deuxième ligne contient la clé.

Chacune des N lignes suivantes contient un message à encrypter ou à décrypter. Un message à encrypter est précédé du caractère « + », un message à décrypter est précédé du caractère « – ».

Sortie

Pour chaque message, le programme écrit le message correspondant encrypté ou décrypté, puis laisse une ligne vide avant de passer éventuellement au message suivant.

Exemple d'exécution

Données (entrée)	Résultats (sortie)
2 LIO +BONNECHANCE -MWBYMQSIBNM	MWBYMQSIBNM BONNECHANCE

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