

Questionnaire de l'Épreuve de Sélection Préliminaire (sur 100 points)

Problème I – COMBINAISONS DE PRODUITS

20 points

Ecrivez un programme qui trouve et affiche à l'écran de combien de manières différentes on peut écrire un nombre naturel donné comme produit de deux nombres naturels.

Explication

Le nombre naturel 60 peut s'écrire comme produit de deux autres nombres naturels de 6 manières différentes (on ignore les écritures commutatives):

1×60; 2×30; 3×20; 4×15; 5×12; 6×10

Restriction

Le nombre naturel donné est compris dans l'intervalle [2; 1000].
Votre programme n'a pas besoin de vérifier l'observation de cette restriction!

Entrée du programme

Lecture du clavier du nombre naturel.

Sortie du programme

Affichage à l'écran du nombre de produits différents.

Exemple d'exécution

```
Nombre naturel? 60  
Nombre de produits différents: 6
```



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

Problème II – MOYENNES

20 points

Ecrivez un programme qui calcule et affiche les moyennes arithmétiques de différents élèves disposant chacun d'un nombre variable de notes.

On fournira en donnée le nombre d'élèves, puis, pour chacun d'entre eux:

- son nom,
- ses différentes notes suivies d'une validation (elles seront terminées par le nombre -1).

Ce n'est qu'après la lecture de l'ensemble des données que le programme affichera un tableau récapitulatif indiquant le nom et la moyenne arithmétique de chaque élève.

Remarque

Le programme devra traiter convenablement le cas où un élève ne dispose d'aucune note. Le tableau récapitulatif contiendra alors le caractère «*» au lieu de sa moyenne.

Exemple d'exécution

```
Combien d'élèves? 3  
Nom de l'élève 1? Géraldine  
Quelles sont les notes de Géraldine?  
28.5  
30  
36  
-1  
Nom de l'élève 2? Laurence  
Quelles sont les notes de Laurence?  
27  
36.5  
-1  
Nom de l'élève 3? Pierre  
Quelles sont les notes de Pierre?  
-1  
  
Résultats  
-----  
Géraldine: 31.50  
Laurence: 31.75  
Pierre: *
```

Restrictions

- $1 \leq \text{Nombre d'élèves} \leq 40$.
- $0 \leq \text{Nombre de notes par élève} \leq 10$.

Votre programme n'a pas besoin de vérifier l'observation de ces restrictions!

Entrée du programme

Lecture du clavier du nombre d'élèves, du nom et des notes de chaque élève.

Sortie du programme

Affichage à l'écran d'un tableau récapitulatif.



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

Problème III – SOUS-MARIN

25 points

Un sous-marin désire explorer un abîme de la Fosse des Mariannes. Cette fosse est subdivisée en N segments, dont chaque segment a une profondeur différente. L'équipe du sous-marin définit un abîme comme un segment qui a une plus grande profondeur que celle des segments adjacents. Deux segments sont dits adjacents s'ils se trouvent l'un à côté de l'autre, c.-à-d. lorsqu'ils ont un côté commun. Généralement, un segment a donc deux segments adjacents, à l'exception des deux segments au début et à la fin de la fosse (qui ont chacun un seul segment adjacent).

Tâche

Écrivez un programme qui lit les profondeurs des N segments du fichier texte SMARIN.TXT et qui, pour une position de submersion P, détermine et affiche la position de l'abîme le plus proche.

Restrictions

- $1 \leq N \leq 1000$ Le nombre de segments.
- $1 \leq H_k \leq 10000$ La profondeur du segment k.
- $1 \leq P \leq N$ La position de submersion.

Votre programme n'a pas besoin de vérifier l'observation de ces restrictions!

Entrée du programme

- La première ligne du fichier texte SMARIN.TXT contient un seul nombre entier positif: le nombre de segments N.
- La deuxième ligne du fichier texte contient un seul nombre entier positif: la position de submersion.
- Les N prochaines lignes décrivent les différents segments, un segment par ligne. La k-ième ligne contient un seul nombre entier positif H_k qui représente la profondeur du segment k.

Sortie du programme

Le programme doit afficher à l'écran une seule ligne avec un seul nombre entier: la position de l'abîme le plus proche.

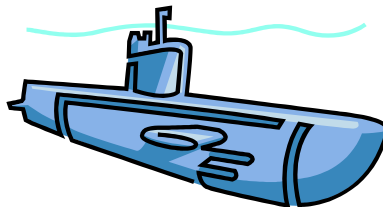
Exemple d'exécution

Contenu du fichier SMARIN.TXT	Affichage sur l'écran
5	3
4	
6	
5	
8	
7	
2	

A partir du segment 4, qui a une profondeur de 7, l'abîme le plus proche est celui à la position 3 qui a une profondeur de 8.



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



Problème IV – LE MOT MYSTERE

35 points

Soit une grille carrée $N \times N$ avec des lettres majuscules non accentuées et soit une série de mots. Ecrivez un programme qui trouve et marque tous les mots de la série dans la grille. Dans cette dernière, un mot peut être lu dans tous les sens, de la droite vers la gauche et inversement, du haut en bas et inversement ainsi que diagonalement du haut en bas et inversement. Finalement le programme découvre et affiche à l'écran le mot mystère. Ce dernier se compose des lettres non marquées de la grille, c.-à-d. qui ne correspondent pas à un mot de la série. Le sens de lecture du mot mystère est du haut en bas et de la gauche vers la droite.



Exemple

Grille 8x8

I	I	C	S	A	J	P	X
N	N	L	A	V	I	E	T
D	P	T	R	X	T	B	U
E	U	A	E	A	I	S	O
X	T	L	L	R	E	U	E
P	A	C	M	A	N	P	M
P	A	S	C	A	L	E	I
L	A	N	I	M	R	E	T

Série de mots

PIXEL
USB
LATEX
INTERNET
PASCAL
OSI
PEARL
INPUT
ASCII
TERMINAL
PACMAN
TIMEOUT
INDEX

I	I	C	S	A	J	P	X
N	N	L	A	V	I	E	T
D	P	T	R	X	T	B	U
E	U	A	E	A	I	S	O
X	T	L	L	R	E	U	E
P	A	C	M	A	N	P	M
P	A	S	C	A	L	E	I
L	A	N	I	M	R	E	T

Le mot mystère est JAVA.

Restrictions

- $4 \leq N \leq 127$.
- La longueur des mots de la série est comprise entre 2 et N.
- Le nombre de mots de la série est compris entre 1 et $2N$.

Votre programme n'a pas besoin de vérifier l'observation de ces restrictions!

Entrées du programme

- Le fichier texte MYSTERY1.TXT correspondant à la grille sous la forme suivante: la première ligne contient un nombre entier naturel correspondant à N, les N lignes suivantes contiennent les lettres (sans espace entre eux)
- Le fichier texte MYSTERY2.TXT correspondant à la série de mots sous la forme suivante: la première ligne contient un nombre entier naturel correspondant au nombre de mots, les lignes suivantes contiennent chacune un mot de la série

Sortie du programme

Affichage à l'écran du résultat.



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