

# Problème I

## Recherche d'un stage [50 points]

**N** élèves d'une classe sont à la recherche d'un stage. L'école leur a fourni une liste de **N** entreprises susceptibles de les accueillir. Chaque élève est obligé d'envoyer une lettre de motivation à chacune des entreprises. Par la suite, chacune des entreprises effectue un classement des élèves par ordre de préférence décroissant. Les élèves de leur côté doivent aussi établir une liste des entreprises par ordre de préférence décroissant.

Le professeur de l'école, responsable des stages, doit trouver une solution au problème suivant: il doit affecter chaque élève à une entreprise, mais en tenant compte des listes de préférences établies: celle des élèves et celle des entreprises.

### Tâche

Écrivez un programme qui, demande d'abord à l'utilisateur d'entrer le nom du fichier contenant les paramètres du problème à résoudre. Il génère ensuite une solution possible en veillant à ce que dans aucun cas, il existe un couple (élève, entreprise) se préférant dans les listes de préférences mutuellement davantage que dans la solution proposée par le programme. Un tel couple est appelé «couple instable».

Exemple d'un couple instable:

- Él1 fait le stage dans En2.
- En1 accueille le stagiaire Él2.
- Él1 préfère En1 à En2.
- En1 préfère Él1 à Él2.

### Restrictions

$1 \leq N \leq 30$       Le nombre d'élèves et d'entreprises.

### Entrée du programme

- La première ligne du fichier de paramètres contient un seul nombre entier positif: le nombre d'élèves et d'entreprises **N**.
- Les **N** lignes suivantes contiennent chacune **N** entiers compris entre 1 et **N** représentant pour la  $i^{\text{ème}}$  de ces lignes l'ordre décroissant de préférence des entreprises pour le  $i^{\text{ème}}$  étudiant. Les **N** entiers sont séparés par un espace.
- Ces **N** lignes sont suivies par une ligne vide.
- Les **N** lignes suivantes contiennent chacune **N** entiers compris entre 1 et **N** représentant pour la  $i^{\text{ème}}$  de ces lignes l'ordre décroissant de préférence des étudiants pour la  $i^{\text{ème}}$  entreprise. Les **N** entiers sont séparés par un espace.

## Sortie du programme

- Le programme affiche une solution pour laquelle il n'existe aucun couple (élève, entreprise) se préférant dans les listes de préférences mutuellement davantage que dans la solution proposée par le programme.
- La première colonne cadrée sur trois espaces représente les numéros des étudiants, la seconde colonne cadrée sur cinq espaces représente les numéros des entreprises.

## Exemples d'exécution

### Exemple 1:

```
IN1 .TXT
  1   2
  2   1
  3   3
```

*Le nom du fichier à traiter est IN1.TXT. Le résultat affiché indique que l'élève 1 devra faire son stage dans l'entreprise 2, l'élève 2 devra faire son stage dans l'entreprise 1 et l'élève 3 devra faire son stage dans l'entreprise 3.*

Voici le contenu du fichier IN1.TXT:

```
3
2 1 3
3 1 2
3 2 1

1 2 3
2 3 1
1 3 2
```

*L'entreprise préférée de l'élève 1 est l'entreprise 2, puis vient l'entreprise 1 et finalement 3.*

*L'entreprise préférée de l'élève 2 est l'entreprise 3, puis vient l'entreprise 1 et finalement 2.*

*L'entreprise préférée de l'élève 3 est l'entreprise 3, puis vient l'entreprise 2 et finalement 1.*

*L'élève préféré de l'entreprise 1 est l'élève 1, puis l'élève 2 et finalement l'élève 3.*

*L'élève préféré de l'entreprise 2 est l'élève 2, puis l'élève 3 et finalement l'élève 1.*

*L'élève préféré de l'entreprise 3 est l'élève 1, puis l'élève 3 et finalement l'élève 2.*

### Exemple 2:

```
IN2 .TXT
  1   4
  2   1
  3   2
  4   3
```

Voici le contenu du fichier IN2.TXT:

```
4
3 2 4 1
2 1 3 4
2 4 1 3
3 1 4 2

1 2 4 3
3 1 4 2
3 2 4 1
2 1 3 4
```

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

## Problème II

### Tracteur

**[50 points]**

Le temps de la moisson (*allemand: die Ernte*) est arrivée! Pendant quelques semaines déjà, l'agriculteur Bob a préparé son tracteur et aujourd'hui il peut commencer à moissonner son champ rectangulaire.

Son champ possède **C** colonnes et **L** lignes et compte donc un total de **C\*L** parcelles. Avant de commencer, Bob inspecte attentivement son champ et il remarque que le rendement **R<sub>p</sub>** de blé des différentes parcelles varie fortement (surement en fonction de la qualité du sol de cette parcelle).



Après son inspection, alors qu'il était justement en train de s'asseoir sur son tracteur pour commencer, Bob remarque qu'il a oublié de faire le plein ! Dans le bac à essence de son tracteur ne restent que **N** litres qui ne suffiront jamais pour moissonner la totalité de son champ. Comme il doit vendre rapidement le plus de blé possible, il doit profiter au maximum des **N** litres qui lui restent: il doit trouver un chemin à travers son champ tel que la quantité de blé moissonnée soit maximale.

### **Tâche**

Écrivez un programme qui lit à partir du fichier TRACTEUR.TXT le champ, la position de départ du tracteur et le nombre de litres restant dans le bac à essence et qui détermine et affiche la quantité maximale de blé que Bob peut moissonner. Le champ rectangulaire compte **C** colonnes et **L** lignes qui forment **C\*L** parcelles. Chaque parcelle possède un certain rendement **R<sub>p</sub>** de blé.

Lorsque le tracteur se trouve sur une parcelle, Bob moissonne tout le blé qui s’y trouve sans consommer d’essence. Le tracteur peut se déplacer horizontalement et verticalement et chaque transition d’une parcelle à une autre coûte exactement 1 litre d’essence.

## Restrictions

$1 \leq C \leq 100$	Le nombre de colonnes du champ.
$1 \leq L \leq 100$	Le nombre de lignes du champ.
$1 \leq P_c \leq C$	La colonne de départ de Bob.
$1 \leq P_l \leq L$	La ligne de départ de Bob.
$1 \leq R_p \leq 10$	Le rendement d'une parcelle.

### Sous-tâche 1 [25 points]

Pour 25 points, la quantité **N** de litres d'essence ne dépassera pas 12.

### Sous-tâche 2 [25 points]

Pour 25 points, la quantité **N** de litres d'essence ne dépassera pas 15.

## Entrée du programme

- La première ligne du fichier texte TRACTEUR.TXT contient un seul nombre entier positif: le nombre de colonnes **C** du champ.
- La deuxième ligne du fichier texte contient un seul nombre entier positif: le nombre de lignes **L** du champ
- Les **L** lignes suivantes contiennent chacune exactement **C** nombres entiers positifs: le rendement **R<sub>p</sub>** des parcelles de cette ligne.
- La ligne suivante contient exactement un seul nombre entier positif: la colonne de départ **P<sub>c</sub>** du tracteur. La première colonne a le numéro 1 et la dernière colonne a le numéro **C** (les **C** colonnes sont donc numérotées de 1 à **C**).
- La ligne suivante contient exactement un seul nombre entier positif: la ligne de départ **P<sub>l</sub>** du tracteur. La première ligne a le numéro 1 et la dernière ligne a le numéro **L** (les **L** lignes sont donc numérotées de 1 à **L**).
- La ligne suivante contient exactement un seul nombre entier positif: le nombre de litres **N** d'essence.

## Sortie du programme

Le programme doit afficher à la console une seule ligne contenant exactement un seul nombre entier positif: la quantité maximale de blé que Bob peut moissonner.

## Exemples d'exécution

### Exemple 1

Contenu du fichier TRACTEUR.TXT:

```
4
5
1 5 5 4
2 5 5 6
2 3 6 3
1 1 8 2
1 1 9 2
3
3
2
```

Affichage à la console: 23

#### Explication:

*Le champ a 4 colonnes et 5 lignes. Le tracteur commence dans la 3<sup>ème</sup> colonne et la 3<sup>ème</sup> ligne dans la parcelle qui a un rendement de 6. Le tracteur a 2 litres d'essence. Bob moissonne d'abord la parcelle dans laquelle il commence (6), puis il roule verticalement vers le bas dans la parcelle avec le rendement 8 (ceci coûte 1 litre d'essence) et finalement il roule verticalement vers le bas dans la parcelle avec le rendement 9 (ceci coûte de nouveau 1 litre d'essence).*

*Rendement total:  $6+8+9 = 23$ .*

### Exemple 2

Contenu du fichier TRACTEUR.TXT:

```
5
5
1 2 2 2 2
1 2 2 2 2
1 2 2 2 2
1 2 2 2 2
9 1 1 1 1
1
1
4
```

Affichage à la console: 13

#### Explication:

*Le meilleur rendement avec 4 litres d'essence à partir de la position de départ est:*

*$1+1+1+1+9 = 13$ .*

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