

## CIL 2009 • Questionnaire de l'Epreuve Finale (sur 100 points)

### Problème I - EXCLUSION

20 points

Le fichier d'entrée EXCLUIN.TXT contient 4 nombres entiers positifs: P, Q, M et N, séparés par un espace. P et Q délimitent un intervalle [P;Q]. M et N servent à exclure des nombres de cet intervalle. Les nombres à exclure sont:

- les nombres divisibles par M et/ou N
- les nombres contenant les chiffres M et/ou N
- les nombres premiers (exemple: 2, 11, 19, ...)
- les nombres dont la somme des chiffres est première (exemple: 25, car  $2 + 5 = 7$ )

Ecrire un programme qui écrit les nombres restants dans le fichier de sortie EXCLUOUT.TXT.

#### Exemple

```
EXCLUIN.TXT:    1 50 3 7
EXCLUOUT.TXT:  1 4 8 10 22 26 40 44 46
```

#### Restrictions

```
0 < Q < 32767
0 < P < 32767
0 < M < 10
0 < N < 10
```



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

### Problème II - CODAGE

30 points

a	s
b	p
c	i
d	o
e	n
f	a
g	g
h	e
i	
j	
k	
l	
m	
n	
o	
p	
q	
r	
s	
t	
u	
v	
w	
x	
y	
z	

Pour ce problème, on demande d'écrire un programme pour coder ou décoder des messages secrets. On utilise un mot secret connu seulement par le codeur et le décodeur. Ce mot ne peut pas contenir de répétitions de caractères.

Le principe est le suivant: on note les lettres de l'alphabet de 'a' à 'z' dans une colonne et les lettres du mot secret à côté. Considérer l'exemple ci-contre pour le mot secret '*spionage*'. Le reste de la deuxième colonne est rempli par les lettres restantes de l'alphabet, donc celles qui ne figurent pas dans le mot secret, mais en commençant en bas. Par exemple, la dernière position, à côté de la lettre 'z', reçoit la lettre 'b', vu que la lettre 'a' est contenue dans le mot secret '*spionage*'.

Le tableau ainsi obtenu sert au codage ainsi qu'au décodage. Par exemple, le message

"attaque demain a l'aube"

se code en

"skksqjn onvszu s w'sjpn"

On remarque que les espaces et les signes de ponctuation ne sont pas codés.

Ecrire un programme qui permet de coder ou de décoder un message, selon le choix de l'utilisateur. On demande également de vérifier si le mot secret dans le fichier d'entrée est valide, c'est-à-dire s'il ne contient pas de répétitions de lettres. Dans le cas contraire, le fichier de sortie devra contenir le message "*Mot secret non valide*".

### Entrée

Le fichier d'entrée CODEIN.TXT contient le mot secret dans la première ligne et le texte dans la deuxième ligne. Au clavier, l'utilisateur entre la lettre majuscule 'C' ou 'D' selon qu'il veut coder ou décoder le texte.

### Sortie

Le fichier de sortie CODEOUT.TXT contient soit le texte codé soit le texte décodé, selon le choix de l'utilisateur.

### Restrictions

La longueur du texte dans CODEIN.TXT ne dépasse pas 255 caractères.

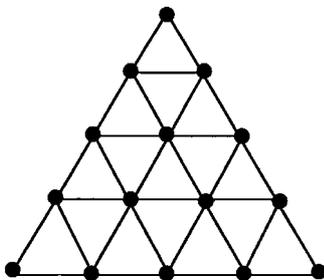
Les messages à coder ou à décoder ne contiennent que des lettres minuscules non-accentuées.



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

## Problème III - Solitaire

**50 points**



Le jeu se joue sur la planche triangulaire que voici. Chaque emplacement, sauf un seul, le trou initial, est occupé par un pion. Il s'agit d'enlever tous les pions sauf le dernier. Les coups s'effectuent en sautant selon les lignes indiquées. Le pion, qui effectue un coup, doit sauter par-dessus un des ses collègues et aboutir dans un trou. Le collègue "sauté" est alors enlevé de la planche. A la fin de la partie, le dernier pion doit se retrouver à l'emplacement du trou initial.

Votre programme doit lire l'emplacement du trou initial dans le fichier d'entrée et noter la suite des coups à effectuer pour aboutir à la solution dans le fichier de sortie.

### Données en entrée

Le fichier d'entrée SOLITIN.TXT contient deux entiers séparés par un espace qui indiquent les coordonnées du trou initial. Le premier entier représente la rangée et le deuxième l'emplacement dans cette rangée.

Exemple: 5 1

Le trou initial se trouve alors dans la cinquième rangée, premier emplacement.

### Données en sortie

Le fichier SOLITOUT.TXT contient la suite des coups à effectuer pour aboutir à la solution de la façon suivante: *coordonnées du point de départ -> coordonnées du point d'arrivée.*

Pour notre exemple:

```
3 1 -> 5 1
3 3 -> 3 1
1 1 -> 3 3
2 1 -> 4 1
4 4 -> 2 2
```

etc.

Si aucune solution n'est possible, le fichier de sortie contient le message:

```
Pas de solution
```



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