

Série I : Manipulation des sous-programmes

Exercice 1

Soit donnée une matrice d'entiers $A[N, N]$.

- Écrire un sous-programme qui initialise cette matrice à zéro :
 - Ligne par ligne.
 - Colonne par colonne.
- Donner un sous-programme qui met à 1 les éléments de la diagonale principale.
- Donner un sous-programme qui met dans $A[i, j]$ la valeur $i + j$ si $i \neq j$.
 La matrice étant symétrique, le sous-programme doit calculer $A[i, j]$ et $A[j, i]$ en même temps.
- Donner un sous-programme qui calcule la somme des éléments de A .
- Donner un sous-programme qui transforme A en A' telle que chaque ligne de A' est obtenue en inversant la ligne correspondante de A .

Exercice 2

Soit les matrices suivantes :

1	0	0	0	0
1	1	0	0	0
1	2	1	0	0
1	3	3	1	0
1	4	6	4	1

- Écrire un sous-programme qui trouve les éléments de ces matrices.
- Donner un sous-programme qui calcule la somme des éléments de ce type de matrice.
- Existe-t-il une propriété qui permet de déduire la somme de ces matrices sans faire le calcul complet?

Exercice 3

Donner un sous-programme qui réalise la transformation suivante :

1	4	7	10		1	4	2	7
2	5	8	11	→	5	3	10	8
3	6	9	12		6	11	9	12

Problème : JEU MASTERMIND

But : dans ce jeu, 2 joueurs s'affrontent : un codificateur et un décodeur.

Le décodeur doit deviner, par déductions successives, la couleur et la position de 5 pions cachés derrière un écran placé par le codificateur.

Jeu : le codificateur place 5 pions de son choix, pris parmi 8 couleurs, derrière l'écran.

C'est au décodeur de jouer : il met 5 pions. Si l'un des pions de sa proposition correspond "par sa couleur et sa position" à un pion caché, le codificateur l'indique en signalant "un pion bien placé". Si l'un des pions correspond uniquement par sa "couleur", le codificateur l'indique en signalant "un pion mal placé". Le décodeur propose de nouvelles combinaisons jusqu'à ce que le codificateur réponde par "5 pions bien placés" ou que le nombre d'essais maximal soit atteint.

Travail demandé

Avec la structure de données suivante :

```
const int NB_PIONS      = 5;
const int NB_COULEURS  = 8;
const int NB_ESSAIS    = 10;

enum couleur_t{ROUGE, VERT, BLEU, JAUNE, NOIR, BLANC, GRIS, VIOLET};

typedef couleur_t tab[NB_PIONS];
```

- 1) Écrire la fonction `recuperer_proposition()` qui récupère la proposition du décodeur.

```
void recuperer_proposition(tab proposition);
```

- 2) Écrire la fonction `evaluer()` qui compare la combinaison du codificateur et la proposition du décodeur et déduit le nombre de pions bien placés et le nombre de pions mal placés.

```
void evaluer(tab combinaison, tab proposition, int &nb_Bien, int &nb_Mal);
```

- 3) Écrire la fonction `jeu()` qui déroule le jeu jusqu'à ce que le décodeur devine la combinaison du codificateur, ou que le nombre d'essais maximal soit atteint.

```
void jeu();
```