Programmation 3

L2 Informatique 2012-2013

Fiche de TP 3

Notions abordées : les opérateurs binaires ; représentations des tableaux à deux dimensions.

Un célèbre problème d'échecs consiste à placer 8 dames sur un échiquier sans qu'aucune n'en menace une autre. Un échiquier est un quadrillage de taille 8x8 et une dame menace toutes les cases situées sur la même colonne, ligne ou diagonales.

Exercice 1. (Représentations des données)

Le but de cet exercice est de définir la façon dont l'échiquier ainsi que la présence ou l'abscence de pièce vont être codés en mémoire. Un soin particulier sera apporté à réduire le coût des différents tests (savoir si une dame en menace une autre, si un ensemble de dame menace une position *etc.*).

Indication : : Figure 1 et Figure 2.

Exercice 2. (Opérations bit à bit)

Dans cet exercice, on va utiliser les opérateurs binaires pour modifier des mots binaires de façon rapide. Toutes les questions de l'exercice doivent être résolues en une ligne.

1. Créer un entier possédant son 15^{eme} bit à 1 et les autres à 0.

2. Créer un entier possédant son 15^{eme} bit à 0 et les autres à 1.

3. Mettre le 15^{eme} bit d'un entier stocké dans une variable x à 1.

	7	6	5	4	3	2	1	0
7	63	62	61	60	59	58	57	56
6	55	54	53	52	51	50	49	48
5	47	46	45	44	43	42	41	40
4	39	38	37	36	35	34	33	32
3	31	30	29	28	27	26	25	24
2	23	22	21	20	19	18	17	16
1	15	14	13	12	11	10	9	8
0	7	6	5	4	3	2	1	0

FIGURE 1 – Indexation des cases de l'échiquier.

63	62	61	...	2	1	0

FIGURE 2 – Représentation en mémoire de l'échiquier.

4. Mettre le 15^{eme} bit d'un entier stocké dans une variable x à 0.

5. Tester si le 15^{eme} bit d'un entier est un 1.

6. Tester l'égalité du 15^{eme} bit de deux entiers x et y (on souhaite seulement savoir si les bits sont identiques, on ne s'intèresse pas à leur valeurs).

Exercice 3. (Cœur du programme)

Dans cet exercice, on va écrire les fonctions de base permettant de faire les différents tests nécessaires à la résolution du problème en utilisant la structure de donnée définie dans l'exercice 1 ainsi que les outils de l'exercice 2.

1. Définir le type `Plateau` en fonction de la discussion de l'exercice 1.

2. Écrire une fonction

```
Plateau placer_dame(int x, int y)
```

qui retourne un plateau contenant uniquement une dame en position (`x`, `y`).

3. Écrire une fonction

```
void afficher_plateau(Plateau p)
```

qui réalise un affichage du plateau dans le terminal. La présence d'une pièce sera notée par un 1 et l'abscence de pièce sera notée par un 0.

4. Écrire une fonction

```
Plateau calculer_cases_attaquees(int x, int y)
```

qui crée un plateau avec l'ensemble des cases attaquées par la case (`x`, `y`).

5. Écrire une fonction

```
int est_attaquee(Plateau p, int x, int y)
```

qui renvoie 1 si la case (`x`, `y`) fait partie des cases attaquées de `p`.

6. Écrire une fonction

```
void calculer_ensemble_cases_attaquees(Plateau P[64])
```

qui mettra à la position `x * 8 + y` du plateau P l'échiquier représentant l'ensemble des cases attaquées par une dame située en position (`x`, `y`)

7. Écrire une fonction

```
Plateau ajouter_reine_plateau(Plateau p, int x, int y)
```

qui crée un nouveau plateau contenant le plateau p, une nouvelle dame à la position $(\mathtt{x}, \mathtt{y})$ ainsi que les cases qu'elle menace.

Exercice 4. (Interface graphique)

Le but de cet exercice est de réaliser une interface graphique utilisant la LibMlV permettant de s'essayer au problème des 8 dames.

1. Écrire une fonction prenant un plateau en argument et affichant ce plateau.

2. Écrire une fonction permettant de récupérer les coordonnées de la case sélectionnée par l'utilisateur.

3. Écrire une fonction qui permette à l'utilisateur d'essayer de résoudre le problème. L'interface indiquera par un moyen de votre choix (visuel, sonore, etc.) lorsqu'un placement illégal a été fait.

   Optionnel : On permettra à l'utilisateur d'annuler un nombre de coups arbitraire en utilisant un bouton *précédent*. Cette fonction peut par exemple utiliser un tableau de 8 plateau représentant les différentes étapes de la résolution.