Licence Informatique 2012-2013 **Programmation 1**
Université Bordeaux 1 **Devoir Surveillé 1h20**

Barème indicatif :

1 Question de cours (répondre en 8 lignes maximum)

Donner une définition de l'interface et de l'implémentation d'un module en language C.
Pourquoi ne faut-il pas inclure des informations concernant l'implémentation dans l'interface ? Justifier.

2 Chaînes de caractères

1. Écrire la fonction **int compteEspace(char chaine[])** qui compte les espaces et les tabulations d'un chaîne de caractères.
2. Écrire la fonction **void supprimeEspacePtr(char* chaine)** qui supprime les espaces et les tabulations d'un chaîne de caractères, en utilisant les pointeurs (sans utiliser aucun indice).

3 Carré magique

Un carré magique d'ordre n est un tableau carré de dimension $n \times n$ dont les éléments sont les nombres entiers de 1 à n^2. Ces nombres sont disposés de manière à ce que leurs sommes sur chaque ligne, sur chaque colonne et sur chaque diagonale principale soient égales.

8	1	6
3	5	7
4	9	2

FIGURE 1 – Exemple de carré magique d'ordre 3

Le but de cet exercice est de construire et d'afficher un carré magique dont l'ordre (c'est-à-dire la taille) est un argument donné en ligne de commande. Plusieurs algorithmes existent pour générer un carré magique. En voici un, nommé méthode siamoise, permettant de créer des carrés magiques d'ordre impair :

3.1 description de l'algorithme

1. Placer d'abord le nombre 1 dans la case située au milieu de la première ligne.
2. Placer les autres nombres en se déplaçant d'une case vers la droite et d'une case vers le haut. Si la nouvelle case n'appartient pas au carré, revenir de l'autre côté, comme si le carré était enroulé sur un tore.
   En résumé, si l'ancienne colonne a pour coordonnées `(l,c)`
   la nouvelle case aura pour coordonnées `((l-1+n)%n,(c+1)%n)`.
3. Si la prochaine case est occupée, décaler d'une case vers le bas par rapport à la position précédente.

Pour tout l'exercice, le carré magique sera stocké dans un tableau d'entiers à deux dimensions alloué dynamiquement. **Ce tableau est un tableau de tableaux.**

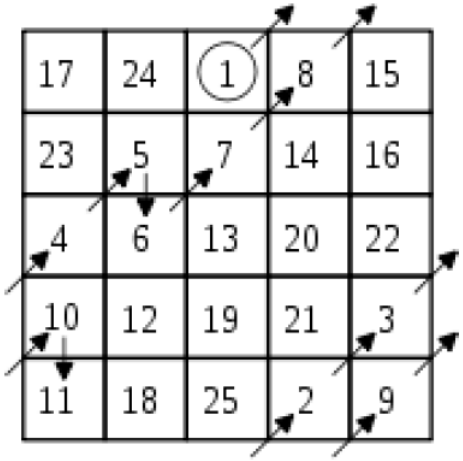

FIGURE 2 – Construction d'un carré magique d'ordre 5

3.2 Implémentation du module

On considère l'interface du module `carreMagique`

```
#ifndef _CARRE_MAGIQUE_H_
#define _CARRE_MAGIQUE_H_

typedef struct carreMagique *carreMagique;

carreMagique creerCarre(int n);
// cree un carre magique de dimension n

void libererCarre(carreMagique c);
// libere la memoire utilisee par le carre magique c

void afficherCarre(carreMagique c);

#endif
```

1. Expliquer brièvement pour quelle(s) raison(s) on utilise l'abstraction de pointeurs.
2. Pour l'implémentation, la structure carreMagique contient l'ordre du carré et le tableau d'entiers à deux dimensions correspondant au carré magique. Expliciter cette structure.
3. Écrire la fonction `creerCarre` qui crée le carré magique d'ordre `taille`, initialise toutes les cases à 0 puis appelle une méthode statique **static void remplirCarre(carreMagique c)** qui remplit le carré magique avec les bons nombres (cette méthode sera implémentée dans la suite de l'exercice).
4. Écrire la fonction `libererCarre` qui libère tout l'espace mémoire alloué par la fonction `creerCarre`
5. Écrire la fonction `afficheCarre` qui affiche le carré magique à l'écran.
6. Écrire la fonction **static void remplirCarre(carreMagique c)** qui remplit le carré c en utilisant l'algorithme décrit précédemment.

3.3 Programme de test

Écrire le programme qui crée et affiche un carré magique dont l'ordre est donné en argument de la ligne de commande. Attention, l'ordre doit être impair. Vous écrirez donc une fonction `usage`.