---

1 Spécifier une structure de données

Une spécification d'une structure de données est une description de la structure qui fait apparaître toutes les informations qu'il est nécessaire de connaître pour programmer des fonctions qui manipulent la structure. Une spécification de la structure suivante

```
struct cercle {
    double abscisse_du_centre;
    double ordonnee_du_centre;
    double rayon;
};
```

serait

> *La structure* `struct cercle` *permet de représenter un cercle. Les champs* `abscisse_du_centre` *et* `ordonnee_du_centre` *contiennent, respectivement, l'abscisse et l'ordonnée du cercle. Le champ* `rayon` *contient son rayon. Il est strictement positif.*

Noter : 1) l'usage du mot « champ » pour distinguer le champ de la structure de la notion géométrique ; 2) l'absence d'intention dans le texte (on écrit « le champ rayon est strictement positif » et pas « il faut que le champ rayon soit strictement positif ») ; 3) l'absence totale de référence à un quelconque algorithme qui utiliserait la structure ; 4) le fait que tous les champs sont décrits.

2 Le jeu SameGnome

La structure de données suivante constitue un choix raisonnable :

```
#define M 15
#define N 10
#define VIDE 0
struct SG {
    int plateau [M][N];
    int nbbilles;
};
```

Le texte ci-dessous a ses qualités (au moins, il fait sourire le correcteur) mais ce n'est pas une spécification. Des variantes proches se rencontrent très fréquemment dans les rapports des étudiants.

1. *Dans le cadre de la formation GIS, nous avons décidé de réaliser le jeu SameGnome en C et pour cela, nous avons défini la structure ci-dessus.*

2. *Dans le but d'assurer un maximum de généralité, nous nous sommes dit que nous allions définir une constante pour les cases vides et que toutes les autres valeurs prises par les autres cases du plateau correspondraient à des billes de différentes couleurs.*

3. *Lorsqu'une case est sélectionnée on la colorie en jaune. Les quatre cases adjacentes sont à leur tour coloriées en jaune et ainsi de suite. Ensuite, les billes situées sur ces cases sont considérées comme retirées du jeu et les billes situées au-dessus « tombent » verticalement pour ne pas laisser de « trou ».*

4. *De même, si une colonne devient vide, toutes les colonnes situées à sa droite sont décalées vers la gauche, de façon à ce que toutes les colonnes occupées soient tassées le plus à gauche possible.*

Question 1. Repérer au moins une phrase qui indique une intention et au moins une référence à un algorithme dans le texte précédent. Tous les champs sont-ils décrits ?

Les questions qui suivent sont destinées à transformer ce texte en une vraie spécification. Lorsque la structure de données est un peu compliquée, il peut être utile de commencer par un paragraphe qui définisse précisément certaines expressions qu'on souhaite utiliser et qui pose quelques notations (un peu comme dans un contrat). Pour vous aider, voici un tel paragraphe :

La structure `struct SG` *permet de représenter un plateau du jeu SameGnome. Chacune des $M \times N$ cases de ce plateau peut soit être vide, soit contenir une bille. Chaque bille a une couleur, représentée par un entier. Les lignes et les colonnes du plateau sont numérotées. La ligne d'indice zéro est la ligne du bas. La colonne d'indice zéro est la colonne de gauche. Chaque case est repérée par ses coordonnées : un couple d'entiers (i, j) formé d'un indice de ligne $i \in I = [0, M-1]$ et d'un indice de colonne $j \in J = [0, N-1]$. La case de coordonnées (i, j) est notée $P_{i,j}$.*

Question 2. Lister les expressions précisées et les notations introduites dans le texte ci-dessus.

On cherche maintenant à spécifier le champ `plateau`. On commence ainsi

Le champ `plateau` *contient le plateau. La case* `plateau[i][j]` *de ce champ contient $P_{i,j}$.*

Question 3. Traduire la partie utile du paragraphe **2** (préciser les valeurs possibles de `plateau[i][j]` et leur interprétation).

Corrigé : *La case $P_{i,j}$ est vide si* `plateau[i][j]` *vaut* `VIDE`*. Si* `plateau[i][j]` *a une valeur c différente de* `VIDE`*, la case $P_{i,j}$ contient une bille de couleur c.*

Question 4. Traduire la partie utile du paragraphe **3** : exprimer l'idée en une phrase, puis reformuler cette même idée de façon rigoureuse au moyen d'une formule quantifiée $(\forall, \exists)$.

Corrigé : *Sur une colonne quelconque du plateau, les billes sont tassées sur les lignes les plus basses. En d'autres termes,*

$$\forall\,(i,j) \in I \times J\,,\ P_{i,j} \neq \texttt{VIDE} \Rightarrow \forall\, i' \in [0, i-1]\,,\ P_{i',j} \neq \texttt{VIDE}\,.$$

Question 5. C'est le bon moment pour expliquer qu'on peut facilement repérer une colonne vide grâce à sa case du bas. Faites-le.

Corrigé : *En particulier si la case du bas d'une colonne est vide alors toute la colonne est vide. En d'autres termes,*

$$\forall\, j \in J\,,\ P_{0,j} = \texttt{VIDE} \Rightarrow \forall\, i \in I\,, P_{i,j} = \texttt{VIDE}\,.$$

Question 6. Traduire la partie utile du paragraphe **4** : exprimer l'idée en une phrase, puis reformuler cette même idée de façon rigoureuse au moyen d'une formule quantifiée $(\forall, \exists)$.

Corrigé : *Les colonnes non vides sont tassées aux indices les plus à gauche. En d'autres termes (et en utilisant la propriété ci-dessus),*

$$\forall\, j \in J\,,\ P_{0,j} = \texttt{VIDE} \Rightarrow \forall\, j' \in [j+1, N-1]\,,\ P_{0,j'} = \texttt{VIDE}\,.$$

Question 7. Spécifier le champ `nbbilles` : exprimer l'idée en une phrase, puis reformuler cette même idée de façon rigoureuse en utilisant la notion de cardinal d'un ensemble.

Corrigé : *Le champ* `nbbilles` *contient le nombre de billes présentes sur le plateau. En d'autres termes,*

$$\texttt{nbbilles} = \left|\{(i,j) \in I \times J \mid P_{i,j} \neq \texttt{VIDE}\}\right|\,.$$