

Feuille de Travaux Pratiques n°3  
Tableaux d'entiers

### 1. Gestion d'ensemble par des tableaux

On considère le type suivant, avec  $n$  prédéfini :

```
typedef int ensemble[n];
```

Ce type permet de représenter des ensembles d'au plus  $n$  entiers, appartenant à l'intervalle  $[0..n-1]_{\mathbb{N}}$ .

Si une variable  $E$  est définie dans ce type, alors on aura :

$\forall i$ , tel que  $0 \leq i \leq n-1$ ,

si  $i$  est élément de l'ensemble alors  $E[i] = 1$

sinon  $E[i] = 0$

Ecrire un programme contenant les différentes fonctionnalités suivantes, avec un menu permettant de les manipuler :

- 1.1. **Initialisation à vide** : affectation à vide d'un ensemble  $E$ .
- 1.2. **Initialisation** : affectation d'un ensemble  $E$  à un ensemble d'entiers saisis au clavier.
- 1.3. **Test de vacuité** : est-ce qu'un ensemble  $E$  est vide ?
- 1.4. **Cardinal** : donner le nombre d'éléments d'un ensemble  $E$ .
- 1.5. **Test d'appartenance** : est-ce qu'un entier  $x$  appartient à un ensemble  $E$  ?
- 1.6. **Ajout d'élément** : ajout d'un entier  $x$  dans l'ensemble  $E$ .
- 1.7. **Enumération** : afficher les éléments d'un ensemble  $E$ .
- 1.8. **Fonctions C** : reprendre ces questions en les programmant sous forme de fonctions C.

### 2. Gestion de relations binaires par des tableaux à deux dimensions

On s'intéresse à la gestion de relations binaires entre entiers. Pour représenter une relation binaire, nous utilisons des tableaux "à deux dimensions", en considérant le type suivant, avec  $n$  prédéfini :

```
typedef int relation[n][n];
```

Ce type permet de représenter des relations au sein d'un ensemble d'au plus  $n$  entiers, appartenant à l'intervalle  $[0..n-1]_{\mathbb{N}}$ . Si une variable  $R$  est définie dans ce type, qui représente une relation  $\rho$ , alors on aura :

$\forall i \in [0..n-1]_{\mathbb{N}}, \forall j \in [0..n-1]_{\mathbb{N}}$ ,

si la relation  $\rho(i,j)$  existe, alors  $R[i][j] = 1$

sinon  $R[i][j] = 0$

Ecrire un programme contenant les différentes fonctionnalités suivantes, avec un menu permettant de les manipuler :

- 2.1. **Initialisation** : affectation à vide d'une relation  $R$ .
- 2.2. **Initialisation** : affectation d'une relation  $R$  à un ensemble de couples saisis au clavier.
- 2.3. **Test de réflexivité** : est-ce qu'une relation  $R$  est réflexive, c'est à dire  
 $\forall i \in [0..n-1]_{\mathbb{N}}, \rho(i,i)$  existe
- 2.4. **Test de transitivité** : est-ce qu'une relation  $R$  est transitive, c'est à dire :  
 $\forall i \in [0..n-1]_{\mathbb{N}}, \forall j \in [0..n-1]_{\mathbb{N}}, \rho(i,j) \text{ et } \rho(j,k) \Rightarrow \rho(i,k)$