

Solution exercice 3.6

```
#include <inttypes.h>
#include <stdio.h>
#include <stdlib.h>

#define TAILLE_MAX_NOM 20

typedef char Nom[TAILLE_MAX_NOM + 1];
typedef enum {SUISSE, ETRANGER} Nationalite;
typedef enum {A, B, C} TypePermis;

const char* const NATIONALITES[] = {"Suisse", "Etranger"};
const char* const TYPES_PERMIS[] = {"A", "B", "C"};

typedef union {
    uint8_t tauxActivite;
    TypePermis typePermis;
} Specificites;

typedef struct {
    Nom nom;
    Nationalite nationalite;
    Specificites specificites;
} Personne;

void afficher(const Personne* p);

int main(void) {
    Personne
        p1 = {"Toto", SUISSE, {.tauxActivite = 80}},
//      p1 = {"Toto", SUISSE, {80}}, // possible
        p2 = {"Titi", ETRANGER, {.typePermis = C}};
//      p2 = {"Titi", ETRANGER, {C}}; // pas correct !

    afficher(&p1); printf("\n");
    afficher(&p2); printf("\n");

    return EXIT_SUCCESS;
}

void afficher(const Personne* p) {
    printf("Nom          : %s\n", p->nom);
    printf("Nationalite   : %s\n", NATIONALITES[p->nationalite]);
    if (p->nationalite == SUISSE)
        printf("Taux activite : %" PRIu8 "%%\n", p->specificites.tauxActivite);
    else
        printf("Type permis   : %s\n", TYPES_PERMIS[p->specificites.typePermis]);
}
```

Autre variante possible :

```
#include <inttypes.h>
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define TAILLE_MAX_NOM 20

typedef char Nom[TAILLE_MAX_NOM + 1];
typedef enum {SUISSE, ETRANGER} Nationalite;
typedef enum {A, B, C} TypePermis;

const char* const NATIONALITES[] = {"Suisse", "Etranger"};
const char* const TYPES_PERMIS[] = {"A", "B", "C"};

typedef struct {
    uint8_t tauxActivite;
} Suisse;

typedef struct {
    TypePermis typePermis;
} Etranger;

typedef union { // Variante plus claire et plus évolutive que précédemment
    Suisse suisse;
    Etranger etranger;
} Specificites;

typedef struct {
    Nom nom;
    Nationalite nationalite;
    Specificites specificites;
} Personne;

Personne suisse(const char* nom, uint8_t tauxActivite);
Personne etranger(const char* nom, TypePermis typePermis);
void afficher(const Personne* p);

int main(void) {
    Personne
        p1 = suisse("Toto", 80),
        p2 = etranger("Titi", C);

    afficher(&p1); printf("\n");
    afficher(&p2); printf("\n");

    return EXIT_SUCCESS;
}

Personne suisse(const char* nom, uint8_t tauxActivite) {
    Personne p = {"", SUISSE, {.suisse = {tauxActivite}}};
    // Autres variantes possibles :
    // Personne p = {"", SUISSE, {{tauxActivite}}};
    // Personne p = (Personne) {"", SUISSE, {.suisse = {tauxActivite}}};
    // Personne p = (Personne) {"", SUISSE, {{tauxActivite}}};
    strncpy(p.nom, nom, TAILLE_MAX_NOM);
    return p;
}
```

```
Personne etranger(const char* nom, TypePermis typePermis) {
    Personne p = {"", ETRANGER, {.etranger = {typePermis}}};
    strncpy(p.nom, nom, TAILLE_MAX_NOM);
    return p;
}

void afficher(const Personne* p) {
    printf("Nom          : %s\n", p->nom);
    printf("Nationalite   : %s\n", NATIONALITES[p->nationalite]);
    if (p->nationalite == SUISSE)
        printf("Taux activite : %" PRIu8 "%%\n",
            p->specificites.suisse.tauxActivite);
    else
        printf("Type permis   : %s\n",
            TYPES_PERMIS[p->specificites.etranger.typePermis]);
}
```

Solution exercice 3.8

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

// Si l'utilisateur saisit une chaîne correspondant à l'une des
// chaînes figurant dans le tableau tab, alors la fonction renvoie le
// rang de cette chaîne. Renvoie -1, sinon.
int lireChaine(const char* const tab[], size_t n, const char* msgInvite);

int main(void) {

    typedef enum
        {LUNDI, MARDI, MERCREDI, JEUDI, VENDREDI, SAMEDI, DIMANCHE} Jour;

    const char* const JOURS[] = {"lundi", "mardi", "mercredi", "jeudi",
                                   "vendredi", "samedi", "dimanche"};

    const size_t NB_JOURS = sizeof(JOURS) / sizeof(char*);

    const char* const MSG_INVITE
        = "Donnez un jour de la semaine en toutes lettres et en minuscules : ";

    // Saisie, par l'utilisateur, d'un jour de la semaine.
    // Attention! Saisie non contrôlée.
    Jour jour = lireChaine(JOURS, NB_JOURS, MSG_INVITE);

    if ((int)jour != -1)
        // Affichage (cyclique) des jours de la semaine à partir du jour saisi
        for (size_t i = 0; i < NB_JOURS; ++i)
            printf("%s\n", JOURS[(jour + i) % NB_JOURS]);
    else
        printf("Saisie incorrecte. Desole!\n");

    printf("\n");
    return EXIT_SUCCESS;
}

int lireChaine(const char* const tab[], size_t n, const char* msgInvite) {
    #define TAILLE_MAX_CHAINE 8
    char chaineControle[4];
    sprintf(chaineControle, "%%ds", TAILLE_MAX_CHAINE);
    char chaine[TAILLE_MAX_CHAINE + 1];
    printf(msgInvite);
    scanf(chaineControle, chaine);
    // vider le buffer
    fflush(stdin); // ou fseek(stdin, 0, SEEK_END);
    for (size_t i = 0; i < n; ++i)
        if (strcmp(chaine, tab[i]) == 0)
            return (int)i;
    return -1;
}
```