

/*

```

-----
Nom du fichier : vehicule.c
Nom du labo   : Laboratoire no. 2
Auteur(s)    : Eric Peronetti, Grégory Rey-Mermet, Célestin Piccin
Date creation : 24.05.2022
Description   : Fichier contenant les définitions nécessaires pour la voiture
Remarque(s)  : -
Compilateur  : Mingw-w64 g++ 11.2.0
-----

```

*/

```

#include <stdio.h>    //printf, PRI
#include <inttypes.h> //uint
#include <string.h>   //strncpy

#include "vehicule.h"

#define FORMAT "%-*s : "

//Les noms des types de véhicule en toutes lettres
const char* const TVEHICULE[] = {"Camionette", "Voiture"};
//Les noms des types de voiture en toutes lettres
const char* const TVOITURE[] = {"Standard", "Haut de gamme"};

//Le nombre de caractères pour le format (le format demande un int et non un size_t)
const int NOMBRE_CARACTERE = 24;

void afficherVehicule(const Vehicule* vehicule) {
    printf(FORMAT "%s\n", NOMBRE_CARACTERE, "Type de vehicule", TVEHICULE[vehicule->tVehicule]);
    printf(FORMAT "%s\n", NOMBRE_CARACTERE, "Plaque", vehicule->numeroPlaque);
    printf(FORMAT "%s\n", NOMBRE_CARACTERE, "Marque", vehicule->marque);

    switch (vehicule->tVehicule) {
        case CAMIONETTE:
            printf(FORMAT "%.2lf\n",
                NOMBRE_CARACTERE,
                "Volume de transport [m3]",
                vehicule->typeVehicule.camionette.volTransport);
            break;
        case VOITURE:
            printf(FORMAT "%" PRIu16 "\n",
                NOMBRE_CARACTERE,
                "Poids [kg]",
                vehicule->typeVehicule.voiture.poids);

            printf(FORMAT "%s\n",
                NOMBRE_CARACTERE,
                "Categorie de voiture",
                TVOITURE[vehicule->typeVehicule.voiture.tVoiture]);

            switch (vehicule->typeVehicule.voiture.tVoiture) {
                case STANDARD:
                    printf(FORMAT "%" PRIu16 "\n",
                        NOMBRE_CARACTERE,
                        "Cylindree [cm3]",
                        vehicule->typeVehicule.voiture.typeVoiture.voitureStd.cylindree);

                    printf(FORMAT "%" PRIu16 "\n",
                        NOMBRE_CARACTERE,
                        "Rejet CO2 [g/km]",
                        vehicule->typeVehicule.voiture.typeVoiture.voitureStd.rejetCO2);
                    break;
                case HAUT_GAMME:
                    printf(FORMAT "%" PRIu16 "\n",
                        NOMBRE_CARACTERE,
                        "Puissance [CV]",
                        vehicule->typeVehicule.voiture.typeVoiture.voitureHg.puissance);
                    break;
            }
            break;
    }
}

```

```

Vehicule voitureStandard(const char* numeroPlaque,
    const char* marque,
    uint16_t poids,

```

```
        uint16_t cylindree,
        uint16_t rejetCO2) {
//Ici il n'est pas nécessaire de mettre tous les noms des paramètres mais pour
//une question de lisibilité nous avons décidé de tous les préciser
Vehicule v = {
    .tVehicule=VOITURE,
    .numeroPlaque="",
    .marque="",
    .typeVehicule={
        .voiture={
            .tVoiture=STANDARD,
            .poids=poids,
            .typeVoiture={
                .voitureStd={
                    .cylindree=cylindree,
                    .rejetCO2=rejetCO2
                }
            }
        }
    }
};

//Recopie du numéro de plaque et de la marque dans le vehicule
strncpy(v.numeroPlaque, numeroPlaque, TAILLE_MAX_NUMERO_PLAQUE);
strncpy(v.marque, marque, TAILLE_MAX_MARQUE);

return v;
}

Vehicule voitureHautDeGamme(const char* numeroPlaque,
                             const char* marque,
                             uint16_t poids,
                             uint16_t puissance) {
//Ici il n'est pas nécessaire de mettre tous les noms des paramètres mais pour
//une question de lisibilité nous avons décidé de tous les préciser
Vehicule v = {
    .tVehicule=VOITURE,
    .numeroPlaque="",
    .marque="",
    .typeVehicule={
        .voiture={
            .tVoiture=HAUT_GAMME,
            .poids=poids,
            .typeVoiture={
                .voitureHg={
                    .puissance = puissance
                }
            }
        }
    }
};

//Recopie du numéro de plaque et de la marque dans le vehicule
strncpy(v.numeroPlaque, numeroPlaque, TAILLE_MAX_NUMERO_PLAQUE);
strncpy(v.marque, marque, TAILLE_MAX_MARQUE);

return v;
}

Vehicule camionette(const char* numeroPlaque, const char* marque, double volTransport) {
//Ici il n'est pas nécessaire de mettre tous les noms des paramètres mais pour
//une question de lisibilité nous avons décidé de tous les préciser
Vehicule v = {
    .tVehicule=CAMIONETTE,
    .numeroPlaque="",
    .marque="",
    .typeVehicule={
        .camionette={
            .volTransport=volTransport
        }
    }
};

//Recopie du numéro de plaque et de la marque dans le vehicule
strncpy(v.numeroPlaque, numeroPlaque, TAILLE_MAX_NUMERO_PLAQUE);
strncpy(v.marque, marque, TAILLE_MAX_MARQUE);
```

```
    return v;
}

const char* obtenirNomTVehicule(TVehicule tVehicule) {
    return TVEHICULE[tVehicule];
}

const char* obtenirNomTVoiture(TVoiture tVoiture) {
    return TVOITURE[tVoiture];
}
```