

Structures

Arnaud Bannier
Nicolas Bodin
Aurélien Texier

1. La théorie

Exercice 1. Les structures

- 1.1) Définissez une structure `str1` contenant deux entiers `a` et `b` et un caractère `c`, puis déclarez-la et initialisez-la aux valeurs 10, 15 et `'c'`.
- 1.2) Modifiez la structure précédente pour faire une permutation circulaire des variables `a`, `b` et `c` (`a ← b`, `b ← c`, etc.). Attention aux casts !
- 1.3) Déclarez une structure `Etudiant` comportant les quatre champs suivants : Nom, Prenom, Numero et moyenne. Attention aux types que vous utiliserez !
- 1.4) Déclarez un pointeur sur une structure `Etudiant` et utilisez-le pour affecter la valeur 7 à la moyenne d'un étudiant.
- 1.5) Comment allouer un tableau de 10 `Etudiant` de façon statique ? De façon dynamique ?

Exercice 2. Les autres types structurés

- 2.1) Déclarez une énumération `JourSemaine` qui liste les jours de la semaine en partant du dimanche, puis déclarez une variable `aujourd'hui` et affectez lui la valeur du jour actuel.
- 2.2) Créez une fonction `AfficheJour` prenant une variable du type `JourSemaine` en paramètre et affichant le jour de la semaine correspondant à la valeur de la variable. L'instruction `if` vous est interdite.
- 2.3) Comment créer une union `Calcul` contenant deux entiers courts `s1` et `s2` et un entier classique `n` dont la modification d'un des entiers courts ne doit pas impacter l'autre entier court, mais peut impacter la valeur de `n` ?
- 2.4) Déclarez une union similaire à l'union `Couleur` du cours sans utiliser de tableau.

Exercice 3. La mémoire

Considérez la définition de structure suivante puis complétez l'état de la mémoire à la fin du programme ci-dessous.

```
1  typedef struct Coord_s
2  {
3      short x,y;
4  }Coord;
5
```

```
1  typedef union Couleur_u
2  {
3      char comp[4];
4      unsigned int RVBA;
5  }Couleur;
6
```

```
1  Couleur rouge = 0xFF000000;
2  Coord points[3];
3  int i;
4  for (i=0; i<3; i++)
5  {
6      points[i].x = i*2;
7      points[i].y = -i/2;
8  }
9
```

Adresse	Valeur de var.	Nom de var.
0x0160		
0x0161		
0x0162		
0x0163		
0x0164		
0x0165		
0x0166		
0x0167		
0x0168		
0x0169		
0x016A		
0x016B		
0x016C		
0x016D		
0x016E		
0x016F		
0x0170		
0x0171		
0x0172		
0x0173		
0x0174		
0x0175		
0x0176		
0x0177		
0x0178		
0x0179		
0x017A		
0x017B		
0x017C		
0x017D		
0x017E		
0x017F		
0x0180		
0x0181		
0x0182		
0x0183		
0x0184		
0x0185		

2. La pratique

Exercice 4. C'est bientôt Noël !

L'objectif de cet exercice est de calculer le nombre de jours restant avant Noël. Les structures vont nous être particulièrement utiles pour nous stocker notre date actuelle de façon appropriée et réaliser un code propre.

- 4.1) Définissez une structure `Date` contenant trois entiers indiquant une `Année`, un `Mois` et un `Jour` dans l'année.
- 4.2) Pour plus de clarté dans le code, modifiez le champ `Mois` de votre structure `Date` pour lui attribuer une énumération listant les différents mois dans l'année (`JAN`, `FEV`, `MAR` ...).
- 4.3) Créez la fonction `SaisirDate()` demandant à l'utilisateur la date actuelle et permettant de renvoyer une structure de type `Date` correctement remplie.
- 4.4) Créez la fonction `JoursAvantNoel()` prenant en paramètres une `Date` et retournant le nombre de jours avant Noël¹. Pour simplifier, nous pourrions supposer que l'année courante n'est pas bissextile².
- 4.5) Créez une fonction `Ajouter()` permettant d'ajouter un certain nombre de jours à une date et retournant cette nouvelle date.
- 4.6) Sachant qu'une année bissextile comporte 366 jours au lieu de 365, et que depuis 1904, les années bissextiles sont les années multiples de 4, créez un programme appelant la fonction `time(NULL)` et retournant une structure `Date` à la date actuelle.

Exercice 5. Les pointeurs, c'est fantastique !

Reprenez les fonctions de l'exercice précédant pour réaliser des passages de variable par adresse et non par valeur lorsque c'est nécessaire.

1. *Indice 1* : Noël ne tombe pas le 21 août cette année.

2. *Indice 2* : vous aurez sûrement besoin de stocker quelque part le nombre de jours présents dans chaque mois pour réaliser correctement l'exercice.