```
  void printDataPoint(struct dataPoint dp) {
    printf("timeSeconds = %d, value = %f\n", dp.timeSeconds, dp.value);
  }
```

Pour modifier les valeurs dans une fonction et les conserver, il faut se servir des pointeurs :

```
  void clearDataPoint(struct dataPoint *dp) {
    dp->timeSeconds = 0;
    dp->value = 0.;
  }
```

où la syntaxe dp->timeSeconds est équivalente à (*dp).timeSeconds. Différent du raccourci "->", le comportement est exactement le même que pour les variables abordées dans le n°5.

Fichiers d'entête

Pour illustrer l'utilisation des en-têtes, l'exemple suivant définit des fonctions et une structure de données d'histogramme. Ceux-ci sont très utiles pour surveiller des résultats sur la durée, fournir des schémas récapitulatifs ou pour le stockage des données elles-mêmes.

Les en-têtes doivent être inclus pour pouvoir utiliser les fonctions des bibliothèques standards ou celles implémentées dans d'autres sources C. Ils contiennent des structures de données et les déclarations des fonctions, mais pas leur implémentation. Celle-ci est faite dans les fichiers .c qui sont compilés pour produire des bibliothèques statiques ou dynamiques, ou pour les lier avec. D'autres en-têtes peuvent être écrits pour contenir des structures de données ou des définitions de fonctions de manière identique aux en-têtes standards.

```
#ifndef HISTOGRAM_H
#define HISTOGRAM_H
#define MAX_BINS 1000
/* Définit une structure pour stocker les données de l'histogramme. */
struct histogram {
  unsigned int nBins;
  float xMin ;
  float xMax;
  float binContents[MAX_BINS];
} ;
/* Définit la structure en tant que nouveau type, comme un raccourci. */
typedef struct histogram Histogram;
/* Remplit un histogramme. */
int fillHist(Histogram *, float value, float weight);
/* Enregistre un histogramme dans un fichier. */
int saveHist(Histogram *, FILE *);
#endif
```

est un fichier d'en-tête appelé histogram.h qui définit une structure et déclare des fonctions, mais n'implémente pas les fonctions qu'il définit. Si le fichier est inclus dans un autre fichier d'en-tête, il pourrait être inclus plusieurs fois dans un programme. Pour empêcher cette double déclaration, la directive de précompilation ifndef est employée. La condition est remplie lors de la première inclusion, et fausse pour les suivantes. La directive define définit des valeurs qui seront remplacées à l'exécution du précompilateur, juste avant la compilation du code. Comme l'allocation dynamique de mémoire n'a pas encore été vue, une taille de tableau fixe est utilisée pour binContents. Enfin, typedef sert à simplifier la déclaration de la variable Histogram. Le fichier d'en-tête doit être inclus dans le programme avant d'utiliser les fonctions ou les structures :

```
#include "histogram.h"
#include <stdio.h>
#include <stdlib.h>
int main() {
unsigned int i;
Histogram h ; /* Crée une structure d'histogramme */
init Hist(&h,10,0.,10.); /* Initialise l'histogramme */
for(i =0;i<1000;i++) { /* Génère 1000 points aléatoires */
 fillHist(&h,10*(float)rand()/RAND_MAX,1.0); /* Chaque valeur de l'histogramme. */
```