Initiation à la programmation en C

TP n°7

Antoine Miné

5 avril 2007

Site du cours: `http://www.di.ens.fr/~mine/enseignement/prog2006/`

Exercice 1. Type complexe.
Proposez un type structuré `complexe` pour les nombres complexes. Celui-ci comportera deux champs de type `double` : une partie réelle et une partie imaginaire.
Proposez ensuite des fonctions pour additionner $+$, multiplier $\times$, élever au carré un nombre complexe et calculer le carré $|\cdot|^2$ de sa norme. On rappelle que $(a+ib)+(a'+ib') = (a+a')+i(b+b')$, $(a+ib)\times(a'+ib') = (a\times a' - b\times b') + i(a\times b' + b\times a')$ et $|a+ib|^2 = a^2+b^2$. On propose les prototypes suivant pour ces fonctions, en utilisant le mécanisme de l'appel par valeur :

```
complexe add    (complexe x, complexe y);
complexe mult   (complexe x, complexe y);
complexe carre  (complexe x);
double   norme2 (complexe x);
```

Exercice 2. Itération de la fonction $z \mapsto z^2 + c$ dans les complexes.
Proposez une fonction :

```
int itere(double x, double y, int maxiter);
```

qui itère la séquence $z_{n+1} = z_n^2 + c$, dans les nombres complexes, avec $z_0 = 0$ et $c = \mathtt{x} + i\mathtt{y}$. La séquence est itérée à partir de $n = 0$ et jusqu'à avoir soit $|z_n|^2 > 4$, soit $n = \mathtt{maxiter}$. L'entier retourné est la valeur de l'indice n où on s'est arrêté.

Exercice 3. Ensemble de Madelbrot en ASCII-art.
L'ensemble de Mandelbrot est l'ensemble des points c tels que la suite $z \mapsto z^2 + c$ ne diverge pas vers l'infini. En pratique, on se contente de regarder si la suite $z \mapsto z^2 + c$ reste dans le disque $|z|^2 \leq 4$ pendant 50 itérations. On se propose donc d'évaluer `iter(x,y,50)` pour tous les points (`x`,`y`) où `x` varie de -2 à 0.5 avec un pas de 1/30 et `y` de -1 à 1 avec un pas de 1/15. Pour chaque point (`x`,`y`), on affichera un caractère `#` si `iter(x,y,50)` vaut 50 et un espace sinon. L'image est obtenue en affichant, pour chaque `y` de -1 à 1, une ligne de caractères correspondant aux `x` de -2 à 0.5 suivie d'un saut de ligne `\n`.

Exercice 4. Format d'image.
On se donne le type suivant pour représenter une couleur :

```
typedef struct { unsigned char r,v,b; } couleur;
```

où `r`, `v` et `b` représentent le taux de rouge, de vert et de bleu définissant la couleur. 0 signifie l'absence de cette couleur et 255 sa valeur maximale. On peut définir par exemple :

```
couleur noir = { 0, 0 , 0 }, blanc = { 255, 255, 255 }, bleu = { 0, 0, 255 };
```

On propose le type `image` suivant permettant de représenter des images de taille au plus `MAXDIM`× `MAXDIM` *pixels* :

```
#define MAXDIM 512
typedef struct {
  int      largeur;
  int      hauteur;
  couleur  pixels[MAXDIM][MAXDIM];
} image;
```

Proposez une fonction :

```
void init_image(image* img, int largeur, int hauteur);
```

qui initialise l'image pointée par `img` aux dimensions voulues et la peint en noir. Comme les objets de type `image` sont très gros, on préfère les passer par référence.
Proposez également des fonctions :

```
couleur get_pixel(const image* img, int x, int y);
void    set_pixel(image* img, int x, int y, couleur c);
```

qui permettent de connaître ou de changer la couleur d'un pixel. Vous vérifierez dans ces fonctions que les coordonnées `x` et `y` sont licites.

Exercice 5. Ensemble de Mandelbrot graphique.
Proposez une fonction :

```
void mandel(double x1, double y1, double x2, double y2, image* img);
```

qui dessine l'ensemble de Mandelbrot dans l'image `img`. Le coin de coordonnées (0,0) de `img` correspondra à $c = \texttt{x1} + i\texttt{y1}$ et celui de coordonnées maximales correspondra à $c = \texttt{x2} + i\texttt{y2}$. Afin de rendre le résultat plus joli, on affichera un pixel blanc si `iter(x,y,50)` renvoie un résultat impair et un pixel noir sinon.

Exercice 6. Sauvegarde au format `.ppm` binaire.
Proposez une fonction

```
void sauve_image(const char* fichier, const image* img)
```

qui sauvegarde l'image `img` dans le fichier `fichier` au format `.ppm` binaire.
On rappelle qu'un fichier à ce format est composé de trois lignes de textes suivies d'une suite d'octets :
- sur la première ligne, le mot `P6` (le nom du format),
- sur la deuxième ligne, deux entiers séparés par un espace et représentant la largeur l et la hauteur h en pixels de l'image,
- sur la troisième ligne, un entier (généralement 255) représentant l'intensité maximale de référence,
- ensuite, une séquence de $3 \times l \times h$ octets (sans espace ni retour à la ligne) représentant l'intensité de rouge, de vert, puis de bleu de chaque pixel de l'image.

Utilisez cette fonction pour sauvegarder l'ensemble de Mandelbrot calculé à la question précédente. On choisira $x1 = -2$, $x2 = 0.5$, $y1 = -1.25$ et $y2 = 1.25$.