

Pratique de la programmation et projet

TP 1 : prototypes et fichiers de déclaration

Frédéric Vivien

Découpage d'un programme

On veut réaliser un programme de calcul de la norme d'un vecteur tridimensionnel : étant donné un vecteur (x, y, z) on veut calculer sa norme : $\sqrt{x^2 + y^2 + z^2}$.

- Écrivez un programme, en un seul fichier, réalisant ce calcul au moyen des fonctions :
 - `somme` : qui prend en entrée trois entiers et renvoie leur somme ;
 - `carre` : qui prend en entrée un entier et renvoie son carré ;
 - `conversion` : qui prend en entrée un entier et renvoie le réel de même valeur ;
 - `racine_carre` : qui prend en entrée un entier et renvoie sa racine carrée (réelle) ;
 - `norme` : qui prend les coordonnées d'un vecteur sous la forme de trois entiers et renvoie sa norme (qui est un nombre réel).
- Même question que précédemment, les fonctions devant apparaître dans l'ordre alphabétique dans le fichier.
- Même question que précédemment, les prototypes étant cette fois-ci déclarés dans un fichier d'entêtes (« `.h` »).
- Même question que précédemment, le fichier contenant le code (« `.c` ») étant stocké dans un répertoire `src`, et le fichier d'entêtes dans un répertoire `include`.
- Même question que précédemment, chaque fichier source (« `.c` ») ne contenant plus que le code d'une seule fonction.

Réversivité mutuelle

Les fonctions `sinus` et `cosinus` sont définies par les formules :

$$\begin{cases} \cos(nx) = \cos((n-1)x)\cos(x) - \sin((n-1)x)\sin(x) \\ \sin(nx) = \sin((n-1)x)\cos(x) + \cos((n-1)x)\sin(x) \end{cases}$$

Écrivez un programme

- prenant en entrée un entier n et deux réels y et z tels qu'il existe un angle x vérifiant $y = \cos(x)$ et $z = \sin(x)$;
- renvoyant la valeur de $\cos(nx)$.

Votre programme ne devra pas faire appel aux fonctions prédéfinies de calcul du sinus et du cosinus, mais se servir de la définition rappelée ci-dessus et implanter des fonctions récursives de calcul de $\cos(nx)$ et $\sin(nx)$.

Affichage du nombre d'invocations de fonctions

On veut réutiliser ici les fonctions `sinus` et `cosinus` de l'exercice précédent, mais on veut en plus qu'à chaque fois qu'elle sont exécutées, les fonctions `sinus` et `cosinus` affichent le nombre total d'invocations des fonctions `sinus` et `cosinus` depuis le début de l'exécution du programme. Par contre, on ne veut pas changer la signature de ces fonctions (elles prennent les mêmes arguments et renvoient les mêmes valeur que précédemment).

1. Réalisez le comportement demandé en utilisant une variable globale.
2. Réalisez le comportement demandé en utilisant une variable globale, les sources des fonctions `sinus` et `cosinus` résidant dans des fichiers différents.
3. On veut maintenant que seule la fonction `cosinus` affiche le nombre d'invocations, et que le nombre affiché soit juste celui du nombre d'invocations de la fonction `cosinus`. Réalisez le comportement demandé sans utiliser de variables globales.

Fonctions homonymes

On veut écrire un programme qui gère des objets géométriques.

Vous écrirez dans un fichier `carre.c` des fonctions :

- `aire` : qui prend en entrée un réel (la longueur du côté du carré) et qui renvoie l'aire du carré ;
- `périmètre` : qui prend en entrée un réel (la longueur du côté du carré) et qui renvoie le périmètre du carré ;
- `caracteristique_carre` : qui prend en entrée un réel (la longueur du côté du carré) et qui affiche (en appelant les fonctions ci-dessus) l'aire et le périmètre du carré.

Vous écrirez dans un fichier `cercle.c` des fonctions :

- `aire` : qui prend en entrée un réel (le diamètre du cercle) et qui renvoie l'aire du cercle ;
- `périmètre` : qui prend en entrée un réel (le diamètre du cercle) et qui renvoie le périmètre du cercle ;
- `caracteristique_cercle` : qui prend en entrée un réel (le diamètre du cercle) et qui affiche (en appelant les fonctions ci-dessus) l'aire et le périmètre du cercle.

Dans un troisième fichier source, vous écrirez un programme principal qui appellera les deux fonctions `caracteristique_carre` et `caracteristique_cercle`.