

Algorithmique et programmation

Master 2 CCI

26 octobre 2007

1 Exercices

1.1 La fonction sinus

La fonction `sinus` peut être écrite comme une somme de terme polynômial :

$$\sin(x) = \sum_{i=0}^n (-1)^i \frac{x^{2*i+1}}{(2*i+1)!}$$

Écrire une fonction qui prend x et n en paramètres. x représente le point à calculer et n le nombre d'éléments à calculer. Par exemple, si $x = 1.0$ et $n = 5$, calcule la somme des 5 premiers termes de $\sin(1.0)$.

1.2 Les horaires

1.2.1 La structure

Définir la structure horaire.

1.2.2 Affichage

Écrire une fonction affichage qui prend en argument un horaire et l'affiche à l'écran.

1.2.3 Somme

Écrire une fonction qui retourne la somme de deux horaires. En faisant **attention** à gérer correctement le dépassement de 23h59m59s.

1.2.4 La boucle

Écrire une fonction qui prend en argument un horaire *depart*, un horaire *intervalle* et un entier n . Cette fonction devra afficher les n horaires partant de *depart* et ajoutant *intervalle*.

Enfin, écrire une fonction *main* qui appelle cette fonction. Par exemple, le code suivant :

```
int main(void)
{
    struct horaire dep;
    struct horaire inter;

    dep.h = 1;
    dep.m = 0;
    dep.s = 0;

    inter.h = 0;
    inter.m = 4;
    inter.s = 24;

    suitehoraire(dep,inter, 5);
    return 0;
}
```

doit afficher ceci :

```
Horaire 1h 0m 0s
Horaire 1h 4m 24s
Horaire 1h 8m 48s
Horaire 1h 13m 12s
Horaire 1h 17m 36s
```

1.3 Les polynômes du deuxième degré

Nous voulons écrire une fonction qui retourne une structure représentant les racines d'un polynôme du 2^{ème} degré.

1.3.1 La structure

Un polynôme du deuxième degré peut avoir 0, 1, 2 ou une infinité de solutions. La structure doit refléter cela. La structure que nous proposons est :

```
struct spoly {
    int nbsol;
    float x1, x2;
};
```

Nous définissons les conventions suivantes :

- Si $nbsol == 0$, alors il n'y a pas de solutions ;
- Si $nbsol == 1$, alors il y a une solution et le champs $x1$ la contient ;
- Si $nbsol == 2$, alors il y a deux solutions et les champs $x1$ et $x2$ les contiennent ;
- Si $nbsol == -1$, alors il y a une infinité de solutions.

1.3.2 La fonction delta

Écrire une fonction *delta* qui prend en paramètre les coefficients du polynôme et retourne la valeur *delta*. Nous rappelons que : $delta = b^2 - 4 * a * c$.

1.3.3 La fonction de calcul

Ecrire une fonction *reso_poly* qui remplit correctement et retourne une structure *spoly*.

Nous rappelons que :

- Si $delta < 0$, il n'y a pas de solutions ;
- Si $delta == 0$, il y a une solution qui vaut $x = \frac{-b}{2*a}$;
- Si $delta > 0$, il y a deux solutions qui valent $x1 = \frac{-b+\sqrt{delta}}{2*a}$ et $x2 = \frac{-b-\sqrt{delta}}{2*a}$.

1.3.4 Le main

Écrire la fonction *main* qui demande à l'utilisateur les coefficients a , b et c et calcule les racines du polynôme associé. Ensuite, le programme doit afficher les solutions s'il en existe.

Exemple :

Coefficients ?

1 2 1

Il y a une seule solution : x = 1

Coefficients ?

1 2 4

Il n'y a pas de solutions

Coefficients ?

1 5 0

Il y a deux solutions : x1 = 0, x2 = -5