TD/TP 2 : Premiers programmes, fonctions

Gilles Peskine

4 octobre 2002

1 Aire d'un triangle

Écrire un programme qui calcule et affiche l'aire d'un triangle. On rappelle que l'aire d'un triangle de côtés a, b, c est

$$A = \sqrt{s(s-a)(s-b)(s-c)}$$

où

$$s = \frac{a+b+c}{2}$$

2 Frais de port

Une société de vente par correspondance applique la règle suivante pour calculer les frais de port : ils s'élèvent à 5% du montant de la commande avec un minimum de 10 €, et sont offerts pour les commandes de plus de 500 €. Le programme suivant est censé calculer les frais de ports, mais il contient plusieurs erreurs et maladresses. Repérez-les et corrigez-les.

```
int main () {
    float montant, frais;
    printf("Montant de la commande (en euros) : ");
    scanf("%1.2f", &montant);
    frais = 5 / 100 * montant;
    if (frais < 10){
        frais = 10;
        printf("Frais de port : 10 euros\n");
    } else if (montant > 500) {
        frais = 0;
        printf("Pas de frais de port.\n");
    } else {
        printf("Frais de port (5 %) : %1.2f euros\n", frais);
    }
    printf("Somme à payer : %1.2f euros", frais + montant);
}
```

3 Expression d'une durée

Écrire un programme qui lit un nombre de secondes et affiche le nombre de jours, heures, minutes et secondes correspondants. Par exemple, si le nombre de secondes est 309639, alors le nombre de jours est 3, le nombre d'heures est 14, le nombre de minutes est 0 et le nombre de secondes est 39.

4 Résolution d'une équation du second degré

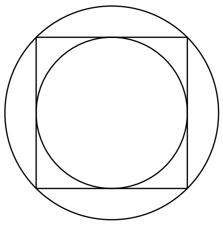
Écrire un programme qui lit trois réels a, b, c et affiche les solutions de l'équation $ax^2 + bx + c = 0$, lorsqu'elles existent. On rappelle que les solutions sont données par l'expression

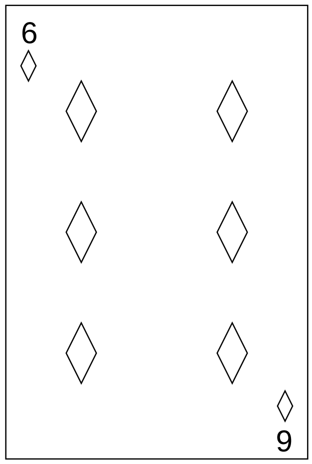
$$\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

Lorsqu'il y a des solutions x_1 et x_2, calculer $ax_1^2 + bx_1 + c$ et $ax_2^2 + bx_2 + c$. Trouve-t-on bien 0 ?

5 Premiers dessins

Pour chacune des figures suivantes, écrire un programme qui dessine la figure dans une fenêtre graphique. Le programme devra se terminer lorsque l'utilisateur clique dans la fenêtre.





6 Déplacement d'une tortue

Dans le langage Logo, on peut afficher des dessins à l'aide d'une *tortue*. À tout instant, la tortue a une *position* et un *cap*. Pour dessiner, on utilise deux fonctions :

1. `avance(float l)` avance d'une longueur `l` en laissant une trace ;
2. `tourne_droite(float a)` tourne sur place d'un angle `a` (en degrés) vers la droite.

Écrire ces deux fonctions, ainsi qu'une fonction `tourne_gauche`. Utiliser trois variables globales `x`, `y`, `cap` pour stocker la position et le cap courants de la tortue.

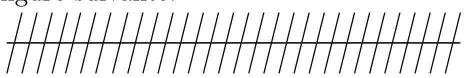
Afin de pouvoir dessiner des figures qui nécessitent plus d'un trait de crayon, ajouter une variable globale entière `crayon`. Modifier la fonction `avance` pour qu'elle trace un trait si `crayon` vaut 1, et qu'elle ne fasse que déplacer la tortue sans dessiner si `crayon` vaut 0.

Utiliser ces fonctions pour dessiner les figures de l'exercice 5.

Les programmes suivants utilisent des boucles. Ils sont plutôt destinés au TD/TP 3.

7 Premières boucles

Écrire un programme qui dessine la figure suivante.



8 Polygones réguliers

Écrire un programme qui lit un nombre entier n et dessine un polygone régulier à n côtés. Utiliser au choix la bibliothèque graphique directement ou la tortue de l'exercice 6.

Note : avant de se lancer dans le programme, il faudra savoir calculer les coordonnées des sommets. Les fonctions mathématiques habituelles `sin`, `cos`, etc. sont déclarées dans `math.h`.