

Interrogation du 17/09/2007

Corrigé

Exercice I

1. Pour répondre à cette question on utilise la réciproque du théorème de Pythagore : si un côté a une longueur dont le carré est égal à la somme des carrés des deux autres côtés alors le triangle est rectangle.

Une première méthode consiste à identifier le plus grand côté par un jeu de tests et à appliquer la réciproque du théorème de Pythagore. Cette méthode conduit à la fonction ci-après

```
int estRectangle (int AB, int AC, int BC)
/*
  Predicat de rectangularite d'un triangle
  ENTREE : AB, AC et BC, les longueurs des cotes du triangle
  SORTIE : 1 si (ABC) est rectangle, 0 si (ABC) n'est pas rectangle
*/
{
  int cote1, cote2, grandcote;

  /* Afin d'appliquer la reciproque du theoreme de Pythagore, on cherche
     a identifier le plus grand cote (grandcote). Les deux autres cotes
     seront cote1 et cote 2 */

  if (AB<AC) {

    /* AB n'est pas le plus grand cote */

    cote1=AB;

    if (AC<BC)
      { grandcote=BC; cote2=AC; }
    else
      { grandcote=AC; cote2=BC; }
  }

  else {

    /* AC n'est pas le plus grand cote */

    cote1=AC;

    if (AB<BC)
      { grandcote=BC; cote2=AB; }
    else
      { grandcote=AB; cote2=BC; }
  }
}
```

```

/* On applique la reciproque du theoreme de Pythagore */

return(grandcote*grandcote==cote1*cote1+cote2*cote2);
}

```

Une seconde méthode consiste à tester les trois cas possibles et à déclarer le triangle rectangle des que l'un des trois tests est vrai. Cette méthode conduit à la fonction suivante.

```

int estRectangle (int AB, int AC, int BC)
/*
  Predicat de rectangularite d'un triangle
  ENTREE : AB, AC et BC, les longueurs des cotes du triangle
  SORTIE : 1 si (ABC) est rectangle, 0 si (ABC) n'est pas rectangle
*/
{
  /* On applique la reciproque du theoreme de Pythagore aux trois
  configurations possibles */

  return((AB*AB==AC*AC+BC*BC) || (AC*AC==AB*AB+BC*BC) || (BC*BC==AB*AB+AC*AC));
}

```

2.

```
#include <stdio.h>
```

LA FONCTION CREE DANS LA QUESTION 1 EST A INSERER ICI

```

int main()
{
  int AB=3, AC=4, BC=5;

  printf("Un triangle dont les cotes ont pour longueurs %d, %d, %d ",AB,AC,BC);

  if (estRectangle2(AB,AC,BC)) printf("est rectangle\n");
  else printf("n'est pas rectangle\n");

  return 0;
}

```

Exercice II

La variable i est incrémentée et la variable j est décrémentée. Leurs valeurs seront donc respectivement 1 et 2.

Le `--` est placé avant le `j` donc la décrémentaion à lieu avant l'affectation. En revanche l'incrémentaion de i à lieu après puisque le `++` est placé après. Ainsi k reçoit la valeur 0.

Comme k est nul, il est faux. Comme i est non nul, il est vrai. Or faux ou vrai est vrai donc $(k||i)$ est vrai. Comme j est non nul, il est vrai. Comme $i+j$ est non nul, il est vrai. Or vrai ou vrai est vrai donc $(j||(i+j))$ est vrai. Comme vrai et vrai est vrai, $(k||i)&&(j||(i+j))$ est vrai. Il a donc la valeur 1.

L'affichage est $i=1, j=1, j=2, k=0, test=1$. On notera que pour la seconde égalité, j apparaît à gauche du signe égal, mais c'est le contenu de la variable i qui est affiché à sa droite.