Responsable pédagogique	AF	AM	PB	PM
Période	Sem1	Sem2	Sem3	Sem4
Volume horaire	Cours/TD		TP	
	2			

STRUF

Triangles d'étoiles

Indicateur temporel :

questions	1h	2h	3h	4h	5h	6h
1 .. 4						

Documents à rendre : Sujet du TD

Méthodologie

Tout programme se devant d'être analyser avant la conception et le réalisation(codage) : vous allez établir les algorithmes en pseudo-code avant de coder en langage C.

Partir du source fourni tristar.c ; puis renommez-le tristar1.c, ou tristar2.c ou tristar3.c selon votre avancement dans les questions.

Tout compilation se fera toujours par la ligne de fabrication : gcc -Wall -o <nom_programme> <nom_source(s)>

ex : gcc -Wall -o tristar1 tristar1.c

Principe de l'algorithme

Le principe consiste à créer des triangles (rectangle, isocèle...) constitué d'étoiles sur un nombre impair de lignes et colonnes (on prendra entre 3 et 15).

Pour le saisie du nombre impair on dispose d'une fonction (sous-programme) nommée *Saisir_nombre_impair* qui vérifie 2 conditions essentielles : le nombre doit être entre une borne min et une borne max ; et ce nombre doit être impair (prendre l'opérateur % *modulo* ; qui permet d'extraire le reste d'une division entière).

Exemple : Soient 2 nombres a et b valant respectivement 5 et 2.

quotient (entier) = a / b ; et le reste = a % b

1. On dispose d'une fonction de saisie du nombre d'étoiles

Compléter dans éléments de réponse le code du sous-programme Saisir_nombre_impair ; afin de répondre au Cdc.(cahier des charges)

2. Tristar1

Ecrire en C un programme qui saisisse un nombre entier impair N entre 3 et 15 inclus et affichant en retour un triangle rectangle rempli d'étoiles.

Vous compléterez le pseudo-code en Annexe2 avant codage en C dans tristar1.c

3. Tristar2

Ecrire un programme qui saisisse un nombre entier impair N entre 3 et 15 inclus et affichant en retour un triangle isocèle rempli d'étoiles. Cf exemple d'exécution.

Le principe consiste à remplir chaque ligne par des blancs (caractère espace) puis mettre une astérisque à partir du bout de chaque ligne de blancs sur les "lig" colonnes de la ligne courante

ex 2eme ligne lig = 3 donc 3 astérisques

L'extrémité d'une ligne de blancs est obtenue par calcul de la moitié de max (décrémenté à chaque itération cF figure géométrique).

4. Tristar3

Ecrire un programme qui saisisse un nombre entier impair N entre 3 et 15 inclus et affichant en retour un triangle isocèle creux dont le contour est fait d'étoiles. Cf exemple d'exécution.

On reprend TRISTAR2 en ajoutant les cas particuliers (pour l'affichage de l'astérisque) : 1ere ou dernier ligne ; 1ere ou 2eme diagonale.

Annexe 1 : Exemples d'exécution

```
patrickMAC:correc maylaenderpatrick$ ./tristar1
valeur impaire ? entre 3 et 15
4
valeur impaire ? entre 3 et 15
5
*
***
*****
```

```
patrickMAC:correc maylaenderpatrick$ ./tristar2
valeur impaire ? entre 3 et 15
5
  *
 ***
*****
```

```
patrickMAC:correc maylaenderpatrick$ ./tristar3
valeur impaire ? entre 3 et 15
5
  *
 * *
*****
```

Annexe 2 :Algorithmes des programmes (à compléter)

```
Programme TRISTAR1;
{Debut PROGRAMME principal TRISTAR1 }
var n est un entier ;
        col, lig : entiers ;

        DébutProg
                n <- Saisir_nombre_impair(     ,      );
                lig <- 1;

                {boucle d'affichage    sur n lignes  }
                TANT QUE    ( lig <=      ) FAIRE

                        {boucle d'affichage * sur lig colonnes  }
                        POUR col <- 1 JUSQU'A ..............   FAIRE
                                Afficher(" ");
                        FINP
                Afficher(CRLF);
                lig <-  ..... ; { ligne suivante : de 2 en 2 !}
                FINTQ
        FinProg
```

```
{ Debut  PROGRAMME principal TRISTAR2 }
var n est un entier ;
        col, lig : entiers ;

        DébutProg
                n <- Saisir_nombre_impair(          ,        );
                lig <- 1;

                TANT QUE ( lig <=     ) FAIRE
                        {boucle d'affichage de "blanc" jusqu'a colonne centrale 4, 3, 2 , 1 }
                        POUR col <- 1 JUSQU'A (n -       ) DIV 2 FAIRE
                                Afficher(          );
                        FINP

                        {boucle d'affichage d'* "au bout de la ligne de blanc"}
                        POUR col <- 1 JUSQU'A   ..... FAIRE
                                Afficher("*");
                        FINP

                Afficher(CRLF);
                 lig <-  ..... ;{ ligne suivante : de 2 en 2 !}
                FINTQ
        FinProg
```

```
{ Debut prog principal TRISTAR3}
var n est un entier ;
        col, lig : entiers ;

        DébutProg
                n <- Saisir_nombre_impair(3, 15);
                lig <- 1;

                TANT QUE lig <= ........        FAIRE
                        {boucle d'affichage de "blanc"}
                        POUR col <- 1 JUSQU'A (n -  ......        ) DIV 2 FAIRE
                                Afficher(  ......        );
                        FINP

                        {boucle d'affichage d'* "aux contours"}
                        POUR col <- 1 JUSQU'A  lig  FAIRE
                                {1ere ligne ou derniere }
                                SI ( lig = 1 ) OU ( lig =.....    ) ALORS
                                        Afficher("*");
                                SINON
                                        {1ere arete(diagonale) ou (2eme) diagonale }
                                        SI ( (col = .....       ) OU (col = .....          ) )ALORS
                                                Afficher( ......      );
                                        SINON
                                                Afficher(........       );
                                        FINSI
                                FINSI
                        FINP

                Afficher(CRLF);
                lig <- ..... ;                {ligne suivante de 2 en 2 }
                FINTQ
        FinProg
```