

// Dans cette suite d'algorithmes, quelques procédures se retrouvent plusieurs fois (ex : polygone, carré). Logoplus détectera ces redondances si, au hasard de vos copier/coller, vous conservez ces procédures en de multiples exemplaires dans l'éditeur. Comme le texte qui suit est assez long, Logoplus mettra du temps avant de vous le signaler. Il est donc déconseillé de le faire compiler entièrement. //

```
// Des arabesques en forme de fleur. //
eff effxt déroule accélère donne "a 0
répète 364 [
    donne "a :a +10 avance 25
    td :a si :a > 100 donne "a -:a
]
```

```
//----- Autre chose: -----//
// D'autres arabesques florales. //
```

```
pour volute
td 90 avance 10
td 90 avance 10
td 90 avance 15
td 90 avance 15
td 90 avance 20
td 90 avance 15
fin
```

```
eff effxt déroule accélère
donne "a 0
donne "vert 120
donne "bleu 120
donne "rouge 50
lèvecrayon avance 100
baissecrayon fixehasard 1 255
répète 20 [
    donne "rouge hasard +120
    donne "vert hasard +120
    donne "bleu hasard +120
    fcc :rouge +:vert +:bleu
    volute td 10
    répète 36 [ volute td 10 avance 200 ]
    avance 200
]
```

```
//----- Autre chose: -----//
```

```
// Encore d'autres arabesques. //
```

```
eff effxt déroule bc accélère
donne "a 0
répète 720 [
    donne "a :a +1
    avance 10 td :a
]
td 90
répète 720 [
    donne "a :a +1
    avance 10 td :a
]
```

```
//----- Autre chose: -----//
```

```
// Un tore. //
```

```
eff effxt déroule bc accélère
fcc bleue
répète 36 [
    avance 15 tg 10 fcc bleue
    répète 36 [ avance 5 td 10 ]
]
```

```
//----- Autre chose: -----//
```

```
// Un soleil orange. //
```

```
eff effxt déroule bc accélère
ct
```

```
pour polygone :longueur :côtés :xc :yc
donne "angle 360 / :côtés
lc av :longueur td 90 + ( :angle / 2 ) bc
répète :côtés [ av :longueur td :angle ] lc
fin
```

```
fcc bleue
répète 36 [
```

```

    polygone 100 3 0 0
    polygone 100 3 0 0
    fcc rouge avance 30
    tg 10
  répète 36 [
    fcc orange polygone 80 3 0 0
    fcc verte
    avance 40 tg 10
  ]
]

//----- Autre chose: -----//
// Une série de triangles torsadés. //
eff effxt déroule bc accélère
ct
donne "x_tortue 150
donne "a 0 donne "b 0
donne "c 0
répète 1360 [
  donne "b :b +20 donne "a :a +2
  donne "c :c +1
  teste :c = 3
  sivrail [ fcc noire donne "c 0 ]
  sifaux fcc bleue
  avance :a td 115
  si :a > 84 [donne "a 0 td :b ]
]

//----- Autre chose: -----//
// Un drapeau . //
eff effxt déroule bc accélère
trait 5 donne "d 0 donne "x 0 donne "y 0
répète 6 [
  bandeb lèvecrayon
  fcc noire avance -10
  baissecrayon
]
origine fpos :x :y donne "x :x +45
donne "y :y +25 fpos :x :y
fcc rouge avance 0
origine fpos :x :y
donne "x :x -8 donne "y :y -5
fpos :x :y fcc noire
lèvecrayon
avance -4 td 90
avance 5 tg 90
baissecrayon
répète 6 [
  td 90 avance 80
  lèvecrayon avance -80
  td 90 avance 10
  tg 180 baissecrayon
]
trait 1

pour bande_rouge
fcc rouge avance 200
fin

pour bande_blanche
fcc blanche avance 200
fin

pour bandeb
td 90 bande_rouge
avance -200 tg 90
avance -10 td 90
bande_blanche avance -200
tg 90
fin

//----- Autre chose: -----//
// Une suite florale. //
eff effxt déroule bc accélère
ct
donne "a 10 donne "b 0
répète 360 [

```

```

        donne "b :b +1 donne "a :a +4
        avance :a td 190
        si :a >100 [ donne "a 0 td :b ]
    ]

//----- Autre chose: -----//
// Une grande rosace. //
eff effxt déroule bc accélère
donne "taille 12
donne "b 0 donne "t 0

pour cube
donne "t -:taille
répète 4 [ avance :taille td 90 ]
avance :taille td 45
avance :taille td 45
avance :taille td 135
avance :taille avance :t
tg 45 avance :taille
td 45 avance :taille
origine
fin

répète 32 [
    donne "b :b +1
    cube td 90
    si :b =8 [
        donne "taille :taille *2
        donne "b 0
    ]
]

//----- Autre chose: -----//
// Une petite fleur. //
eff effxt déroule bc accélère
donne "a 1
donne "b 0 donne "r 0
donne "v 0 donne "b 0
fixehazard 1 255
répète 100 [
    donne "r hazard donne "v hazard
    donne "b hazard
    fcc :r + :v +:b donne "a :a +4
    avance :a tg 45
    si :a >25 [ donne "a 1 td 10 ]
]
cachetortue

//----- Autre chose: -----//
// Un cristal de triangles. //
eff effxt déroule bc accélère
fixehazard 1 255 donne "v 0
répète 6 [
    donne "v :v +10 fcc :v + hazard avance 100
    répète 3 [
        donne "v :v +10 fcc :v + hazard
        si :v >250 donne "v 1 avance 50
        répète 3 [
            donne "v :v +10 fcc :v + hazard
            si :v >250 donne "v 1
            avance 25
            répète 3 [
                donne "v :v +10 fcc :v + hazard
                si :v >250 donne "v 1 avance 12
                répète 3 [
                    donne "v :v +10 fcc :v + hazard
                    si :v >250 donne "v 1 avance 6
                    tg 120
                ]
                td 120
            ]
            tg 120
        ]
        td 120
    ]
    tg 60
]

```

//----- Autre chose: -----//

// Une étoile pleine de triangles. //

eff effxt déroule bc accélère

fixehazard 0 255

donne "v 0 donne "a 0

lèvecrayon tg 90

avance 100 tg 90

avance 100 td 180

baissecrayon

répète 6 [

    donne "a :a +1 flocon

    avance 200 td 60

]

pour flocon

répète 3 [

    donne "v :v +10 fcc :v + hazard

    si :v >250 donne "v 1

    avance 100

    répète 3 [

        donne "v :v +10 fcc :v + hazard

        si :v >250 donne "v 1

        avance 50

        répète 3 [

            donne "v :v +10 fcc :v + hazard

            si :v >250 donne "v 1

            avance 25

            répète 3 [

                donne "v :v +10 fcc :v + hazard

                si :v >250 donne "v 1

                avance 12

                répète 3 [

                    donne "v :v +10 fcc :v + hazard

                    si :v >250 donne "v 1

                    avance 6 tg 120

                    ]

                tg 120

                ]

            tg 120

            ]

                tg 120

                ]

                    tg 120

                    ]

fin

//----- Autre chose: -----//

// Une pointe. //

eff effxt déroule ct bc accélère

donne "taille 100

donne "demitaile :taille /2

donne "demi2 100

donne "nb 0

donne "demi2 50 donne "nb 0

répète 8 [ répète 50 tr\_petite\_branche ]

origine donne "demi2 50

td 45

pour tr\_petite\_branche

td 60 av :demi2

re :demi2 tg 120

av :demi2 re :demi2

td 60

si : demi2 >1 [donne "demi2 :demi2 -1 av 4 ]

fin

//----- Autre chose: -----//

// Un carré fractale. //

eff effxt déroule ct bc accélère

répète 4 [

    fcc noire avance 30

    répète 4 [

        fcc bleue avance 10

        répète 4 [ fcc orange avance 10 td 90 ]

        avance 20 tg 90

    ]

```

    avance 50 td 90
  ]

//----- Autre chose: -----//
// Encore un carré fractale. //
eff effxt déroule bc ct accélère
répète 4 [
  fcc noire avance 50 td 90
  répète 4 [
    fcc bleue avance 25 tg 90
    répète 4 [
      fcc rouge avance 12 td 90
      répète 4 [
        fcc verte avance 6 tg 90
        répète 4 [ fcc violette avance 3 td 90 ]
      ]
    ]
  ]
]

//----- Autre chose: -----//
// Des oursins aléatoires. //
eff effxt déroule accélère bc mt
//***** initialisation des variables*****//
donne "a 1
donne "rouge 0 donne "vert 0 donne "bleu 0
donne "x 0 donne "y 0 donne "de 0
pour oursin
fixehazard 1 32767
répète 100 [
  donne "rouge hasard +100
  donne "vert hasard +100
  donne "bleu hasard +100
  fcc :rouge +:vert +:bleu
  donne "a :a +1 avance :a
  td 190
]
fin

répète 50 [
  fixehazard -200 200 fpos hasard hasard
  oursin donne "a 1
]

//----- Autre chose: -----//
// Un gigantesque carré fractale. //
eff effxt déroule accélère bc mt
répète 4 [
  avance 50 td 90 avance 50 tg 90 avance 50 td 90
  répète 4 [
    avance 25 tg 90 avance 25 td 90 avance 25 tg 90
    répète 4 [
      avance 12 td 90 avance 12 tg 90 avance 12 td 90
      répète 4 [
        avance 6 tg 90 avance 6 td 90 avance 6 tg 90
        répète 4 [ avance 2 td 90 avance 2 tg 90 avance 2 td 90 ]
      ]
    ]
  ]
]

//----- Autre chose: -----//
// Un triangle façon Sierpinski. //
eff effxt déroule accélère bc mt
td 30 fpos -190 (-80)
répète 3 [
  avance 400 td 120
  répète 3 [
    avance 200 td 120
    répète 3 [
      avance 100 td 120
      répète 3 [avance 50 td 120 ]
    ]
  ]
]

//----- Autre chose: -----//

```

```

// Un autre triangle façon Sierpinski. //
eff effxt déroule accélère bc mt
td 30 fpos -130 (-130)
répète 3 [
  avance 400 td 120
  répète 3 [
    avance 200 td 120
    répète 3 [
      avance 100 td 120
      répète 3 [
        avance 50 td 120
        répète 3 [
          avance 25 td 120
          répète 3 [
            avance 12,5 td 120
            répète 3 [avance 6,25 td 120 ]
          ]
        ]
      ]
    ]
  ]
]

```

```

//----- Autre chose: -----//
// Une spirale. //
eff effxt déroule accélère bc ct
donne "a 0
répète 200 [ donne "a :a +1 avance 10 td :a ]

```

```

//----- Autre chose: -----//
// Une spirale quasi-circulaire. //
eff effxt déroule accélère ct lc av 80 bc
donne "a 1
répète 800 [ donne "a :a +0,04 avance :a td 10 ]

```

```

//----- Autre chose: -----//
// Des spirales concentriques. //
eff effxt déroule accélère ct bc
donne "a 1 fixehasard 0 255
répète 400 [
  donne "rouge hasard + 100
  donne "vert hasard + 100
  donne "bleu hasard + 100
  fcc :rouge + :vert + :bleu
  donne "a :a +1
  répète 3 [avance 10 tg 120 ]
  avance :a
  td 145
]

```

```

//----- Autre chose: -----//
// Une étoile grossissante. //
eff effxt déroule accélère ct bc
donne "a 1 fixehasard 0 255
répète 800 [
  donne "rouge hasard + 100
  donne "vert hasard + 100
  donne "bleu hasard + 100
  fcc :rouge + :vert + :bleu
  donne "a :a +0,05
  répète 3 [avance 10 tg 90 ]
  avance :a td 10
]

```

```

//----- Autre chose: -----//
// Une spirale de cercles. //
eff effxt déroule accélère ct bc
donne "a 1 fixehasard 0 255
répète 800 [
  donne "rouge hasard + 100
  donne "vert hasard + 100
  donne "bleu hasard + 100
  fcc :rouge + :vert + :bleu
  donne "a :a +0,05
  répète 36 [avance 2 tg 10 ]
  avance :a td 10
]

```

```
//----- Autre chose: -----//
// Une spirale carrée. //
eff effxt déroule accélère ct bc
donne "a 10
répète 200 [ donne "a :a +2 avance :a td 90 ]
```

```
///----- Autre chose: -----//
// Une spirale hexagonale. //
eff effxt déroule accélère ct bc
donne "a 10
répète 200 [ donne "a :a +1 avance :a td 45 ]
```

```
//----- Autre chose: -----//
// Une spirale hexagonale plus petite. //
eff effxt déroule accélère ct bc
donne "a 1
répète 200 [
    donne "a :a +0,08
    répète 2 [ avance :a td 45 ]
    av 10
]
```

```
//----- Autre chose: -----//
// Une spirale circulaire parsemée de petits triangles. //
eff effxt déroule accélère ct bc
donne "a 1
donne "rouge 0 donne "vert 0 donne "bleu 0
fixehazard 0 255
répète 800 [
    donne "rouge hasard + 100
    donne "vert hasard + 100
    donne "bleu hasard + 100
    fcc :rouge + :vert + :bleu
    donne "a :a +0,05
    répète 3 [avance 10 td 120 ]
    avance :a
    td 10
]
```

```
//----- Autre chose: -----//
// Une spirale triangulaire. //
eff effxt déroule accélère ct bc
donne "a 10
répète 200 [ donne "a :a +4 avance :a td 120 ]
```

```
//----- Autre chose: -----//
// Une courbe spiralée. //
eff effxt déroule accélère ct bc
donne "a 10
répète 200 [ donne "a :a +4 avance :a td 145]
```

```
//----- Autre chose: -----//
// Une autre courbe spiralée. //
eff effxt déroule accélère ct bc
donne "a 10 donne "b 0
répète 200 [
    donne "b :b +1 fcc :b * 1000
    donne "a :a +4
    avance :a td 145
]
```

```
//----- Autre chose: -----//
// Une toile d'araignée. //
eff effxt déroule accélère ct bc

pour branche :taille :demitaille
av :taille
td 60 av :demitaille
re :demitaille
tg 120
donne "taille :taille -4
donne "seuil :seuil +1
teste :seuil >25
sivrai retourne
sifaux branche :taille :demitaille
fin
```

```
ct
donne "taille 100
donne "demitaille :taille /2
donne "seuil 0
branche :taille :demitaille
```

```
///----- Autre chose: -----//
// Trois duodécagones qui se croisent. //
eff effxt accélère déroule ct bc
répète 3 [
    répète 12 [ répète 4 av 12 td 30 ]
    av 24 tg 125
]
```

```
//----- Autre chose: -----//
// Un simple cercle. //
eff effxt accélère déroule ct bc
donne "a 0
donne "taille 100
donne "nbpoint 360
répète 360 [
    donne "a :a +1
    donne "x :taille *(cos :a )+100
    donne "y :taille *(sin :a )+100
    point ph :x :y
]
```

```
//----- Autre chose: -----//
// Une série de rosaces. //
eff effxt accélère déroule ct bc
pour polygone :longueur :côtés :xc :yc
donne "angle 360 / :côtés
lc av :longueur td 90 + ( :angle / 2 ) bc
répète :côtés [ av :longueur td :angle ] lc
fin
```

```
// polygone à 4 côtés .
On fait pivoter le polygone dans le même sens que la tortue. //
répète 36 [ polygone 100 4 45 0 av 10 td 10 ]
attends 1
eff fcc noire
// on fait pivoter le polygone dans le sens inverse de la tortue //
```

```
répète 36 [ polygone 100 4 45 1 av 10 td 10 ]
attends 1
eff fcc bleue
```

```
//polygone à 3 côtés.
On fait pivoter le polygone dans le même sens que la tortue. //
```

```
répète 36 [ polygone 100 3 45 0 av 10 td 10 ]
attends 1
eff fcc verte
```

```
// On fait pivoter le polygone dans le sens inverse de la tortue. //
```

```
répète 36 [ polygone 100 3 45 1 av 10 td 10 ]
```

```
//----- Autre chose: -----//
// Une explosion de cercles. //
eff effxt accélère déroule ct bc
// *****appel de procédures avec paramètres *****//
```

```
pour carré :taille :angle :tour
répète :tour [ avance :taille td :angle ]
fin
```

```
// On appelle la procédure en lui passant 3 paramètres:
la taille du trait (nombre de pas de la tortue)
l'angle, la répétition. //
```

```
//***** déclaration des variables utilisées: *****//
donne "taille 0
donne "angle 0
donne "tour 1
```



```

// *****//

carré 25 10 36

// Les 3 paramètres peuvent être des variables.
// Pour tester ci-dessous ne pas oublier de désactiver
// l'appel carré 25 10 36 ci-dessus.//

répète 20 [
    donne "taille :taille +2
    carré :taille 10 36
]

//----- Autre chose: -----//
// Une spirale carrée. //
eff effxt accélère déroule ct bc

// Exemple de récursivité:
// On va créer une procédure dont le but est de dessiner un carré.
// On augmentera sa taille de 2 pas à chaque tour.
// A l'intérieur, un test fera sortir de la procédure lorsqu'elle aura
// été parcourue 100 fois. //

// **** initialisation des variable employées****//
donne "a 0
donne "taille 2
//*****//

// Création d'une procédure qui va "s'appeler elle-même"//

pour carré
avance :taille
td 90 donne "a :a +1
// Le test qui permet de sortir de la boucle infinie. //

teste :a >100
sivrai retourne
sifaux [
    // Tant qu'on reste dans la procédure, on change la taille du carré //
    donne "taille :taille +2 // Ceci est un appel récursif //
    carré
]
fin
carré

//----- Autre chose: -----//
// Un carré entouré par 4 triangles sur chaque côté. //
eff effxt accélère déroule ct bc

// Exemple de récursivité:
// on va créer une procédure dont le but est de dessiner un carré
// on augmentera sa taille de 2 pas à chaque tour, ce qui créera une spirale "carrée"
// on va créer une procédure triangle.
// A l'intérieur, un test fera sortir de la procédure carré lorsqu'elle aura
// été parcourue 100 fois. //

// **** initialisation des variable employées****//
donne "a 0
donne "taille 4
donne "u 0
// *****//

pour triangle
fcc bleue
répète 3 [ avance :taille tg 120 ]
fin

// Création d'une procédure qui va s'appeler elle-même. //

pour carré
fcc noire
avance :taille td 90
donne "a :a +1 donne "u :u +1
teste :a <100
sivrai [
// Tant qu'on reste dans la procédure on change la taille du carré. //

```

```
donne "taille :taille +2
// Ceci est un appel récursif puisque nous sommes toujours dans la procédure carré. //
```

```
    triangle carré
  ]
sifaux retourne
fin
```

```
// on appelle la procédure carré.//
```

```
carré
```

```
//----- Autre chose: -----//
// Une courbe articulée autour de carrés. //
```

```
eff effxt accélère déroule ct bc
```

```
// retourne est une commande qui permet de sortir
d'une procédure lors d'un appel récursif,
car la poursuite du programme a lieu sur la ligne après l'appel.
Dans un premier temps, la procédure est "apprise" par Logoplus
puis à l'exécution, au bout de 4 appels récursifs, retourne est exécuté. //
```

```
pour arbre :ordre :taille
teste :ordre <1
sivrai [ donne "ordre 4 retourne ]
sifaux [
  teste :taille >200
  sivrai [ donne "taille 2 retourne ]
  sifaux [
    avance :taille
    tg 90 donne "ordre :ordre -1
    arbre :ordre :taille
    tg 90 donne "taille :taille +2
    avance :taille tg 90
    donne "ordre :ordre -1
    arbre :ordre :taille
    tg 90 avance :taille
  ]
]
fin
```

```
donne "ordre 24
donne "taille 2
arbre :ordre :taille
```

```
//----- Autre chose: -----//
// Belle comme un coeur ! //
```

```
eff effxt accélère déroule ct
```

```
// récursivité avec paramètres. //
donne "taille 0
donne "angle 0
donne "rt 16
donne "tour 0
```

```
// On appelle la procédure en lui passant 3 paramètres:
la taille du trait , l'angle et la répétition. //
```

```
fixey -100
tg 65
procédure 50 10 16
// *****La procédure ***** //
```

```
pour procédure :taille :angle :rt
avance :taille
td :angle
donne "taille :taille -2
teste :tour <51
sivrai [donne "tour :tour +1 procédure :taille :angle :rt ]
sifaux stop
fin
```

```
//----- Autre chose: -----//
// Une roue pleine de carrés. //
eff effxt déroule accélère ct
```

```
pour carré
répète 4 [av 15 td 90 ]
fin
```

```
pour branche
av 30 td 90 av 30
td 90 av 30
td 90 av 15
td 90
fin
```

```
pour cube
carré av 15 carré
branche
fin
```

```
//Appel de la macro-procédure cube (qui fait elle-même appel à la procédure carré )://
```

```
répète 12 [ tg 30 cube origine ]
```

```
//----- Autre chose: -----//
```

```
// Un cercle. //
```

```
eff effxt accélère déroule ct
```

```
donne "taille 100
```

```
donne "nbpoint 360
```

```
fcc bleue
```

```
donne "a 0
```

```
répète 360 [
```

```
    donne "a :a +1
```

```
    donne "x :taille *cos(:a) +100
```

```
    donne "y :taille *sin(:a) +100
```

```
    donne "z :taille *cos(:a) +100
```

```
    zpoint :x :y :z
```

```
]
```

```
//----- Autre chose: -----//
```

```
// Une succession de carrés concentriques un peu flous. //
```

```
eff effxt déroule accélère ct
```

```
donne "a 10
```

```
donne "compteur 0
```

```
répète 200 [
```

```
    donne "compteur :compteur +1
```

```
    teste :compteur = 5
```

```
    sivrai [
```

```
        donne "compteur 0
```

```
        lèvecrayon av 10
```

```
        baissecrayon
```

```
    ]
```

```
    donne "a :a +2
```

```
    avance :a
```

```
    td 90
```

```
]
```

```
//----- Autre chose: -----//
```

```
// Un carré en perspective. //
```

```
eff effxt déroule accélère ct
```

```
pour carré :taille
```

```
répète 4 [av :taille td 90 ]
```

```
fin
```

```
donne "a 10
```

```
donne "compteur 0
```

```
donne "taille 2
```

```
répète 20 [
```

```
    donne "taille :taille +10
```

```
    carré :taille lèvecrayon
```

```
    av 5 tg 135 av 6 td 135
```

```
    baissecrayon
```

```
]
```

```
//----- Autre chose: -----//
```

```
// Un célèbre automate cellulaire. //
```

```
eff effxt déroule accélère ct
```

```
// Ce principe est tiré du principe des turmites (Turing et termites). on donne une règle simple à la tortue:
```

Si elle lit un carré noir, alors elle doit dessiner un carré blanc, tourne à droite et avancer.  
Si elle lit un carré blanc, alors elle doit dessiner un carré noir, tourne à gauche et avancer.  
Pendant un certain temps on a l'impression que la tortue dessine de façon chaotique,  
cependant au bout d'un certain temps, elle se met à dessiner une "échelle" très régulière. //

trait 2

pour déplace  
lc av 2 bc  
fin

donne "r 0 donne "v 0  
donne "b 0

pour début  
teste (couleurpoint? lieu ) = eau  
sivrai [ fcc noire point lieu td 90 déplace ]  
sifaux [ fcc eau point lieu tg 90 déplace ]  
début  
fin

début

//----- Autre chose: -----//  
// Deux hexagones collés l'un à l'autre. //  
eff effxt déroule accélère ct  
obéis 1 td 180  
obéis 2 td 180  
// et on dessine !//  
répète 8 [  
    obéis 1 bc av 30 td 45  
    obéis 2 bc av 30 td 45  
    obéis 3 bc av 30 td 45  
]

//----- Autre chose: -----//  
// Un carré bleu et l'autre jaune inclinés différemment. //  
eff effxt déroule accélère ct

// On prépare la tortue 1 avec une couleur jaune. //  
obéis 1 fcc jaune

// On prépare la tortue 2 avec une couleur bleue clair  
et on la pivote de 45 degrés vers la droite. //  
obéis 2 td 45 fcc bleue

// On demande à la tortue 1 de faire un carré. //  
répète 4 [  
    obéis 1 av 100 td 90  
    // et à la tortue 1 de faire un carré  
    mais n'oublions pas que cette tortue est tournée  
    de 45 degrés vers la droite et de couleur bleue clair.//  
    obéis 2 av 100 tg 90  
]

//----- Autre chose: -----//  
// Encore des rosaces. //  
eff effxt déroule bc accélère

pour polygone :longueur : côtés :xc :yc  
donne "angle 360 / :côtés  
lc av :longueur td 90 + ( :angle / 2 ) bc  
répète :côtés [ av :longueur td :angle ] lc  
fin

// Polygone à 4 côtés.  
On fait pivoter le polygone dans le même sens que la tortue. //  
obéis 1 fixex 50 fcc violette  
obéis 2 fixex 150

répète 36 [  
    obéis 1 polygone 100 4 45 0  
    av 10 td 10  
    obéis 2  
    polygone 100 4 45 1  
    av 10 td 10  
]

```
attends 1
eff fcc bleue
```

```
// Polygone à 3 côtés. On fait pivoter le polygone dans le même sens que la tortue. //
```

```
obéis 1
répète 36 [
    polygone 100 3 45 0
    av 10 td 10
]
attends 1
```

```
fcc orange
```

```
// On fait pivoter le polygone dans le sens inverse de la tortue. //
```

```
obéis 1
répète 36 [
    polygone 100 3 45 1
    av 10 td 10
]
```

```
//----- Autre chose: -----//
```

```
// Un grand triangle et un carré entouré par un cercle. // eff efftx déroule bc accélère
```

```
// On peut préparer les paramètres position et couleur de chaque tortue
```

```
à l'avance. //
```

```
obéis 1 fpos 50 50 bc fcc bleue
obéis 2 fpos -60 20 bc fcc verte
obéis 3 fpos 120 20 bc fcc orange
obéis 1 répète 4 [ td 90 av 50 ]
obéis 2 répète 3 [ tg 120 av 200 ]
obéis 3 répète 36 [ tg 10 av 10 ]
```