

Turbo C 2.0

Librería POLYGON.H

Desarrolladores:

- Marcel Emmanuel Díaz Largaespada
- Gabriel Alejandro Ortiz Amador
- Luis Miguel Pineda Joseph
- Engel Gabriel Reyes Moreno
- Obed Isaí Ríos Largaespada

Grupo 2M1 – CO | Generación 2020

1. Librería POLYGON.h

La librería POLYGON cuenta con un conjunto de funciones que le permiten al usuario dibujar polígonos regulares en posiciones variables de la pantalla, auxiliándose del modo gráfico que ofrece **graphics.h**.

1.1. Funciones de la librería POLYGON.h

1.1.1. Drawregpoly:

Sintaxis: drawregpoly(x, y, radio, lados);

Prototipo: drawregpoly(int x, int y, int radius, int sides)

Descripción: La función dibuja en pantalla la figura vacía de un polígono regular de un número de lados y radio determinados, con centro en (x, y).

Argumento	Descripción
x	La coordenada en x del polígono.
y	La coordenada en y del polígono.
radio	La distancia entre el centro de la figura (indicado por el par ordenado [x,y]) y cualquiera de sus vértices.
lados	El número de lados que posee el polígono (abarca desde un mínimo de 3 lados a un máximo de 30)

Ejemplo:

Entrada

```
#include <graphics.h>
#include <POLYGON.h>
#include <conio.h>

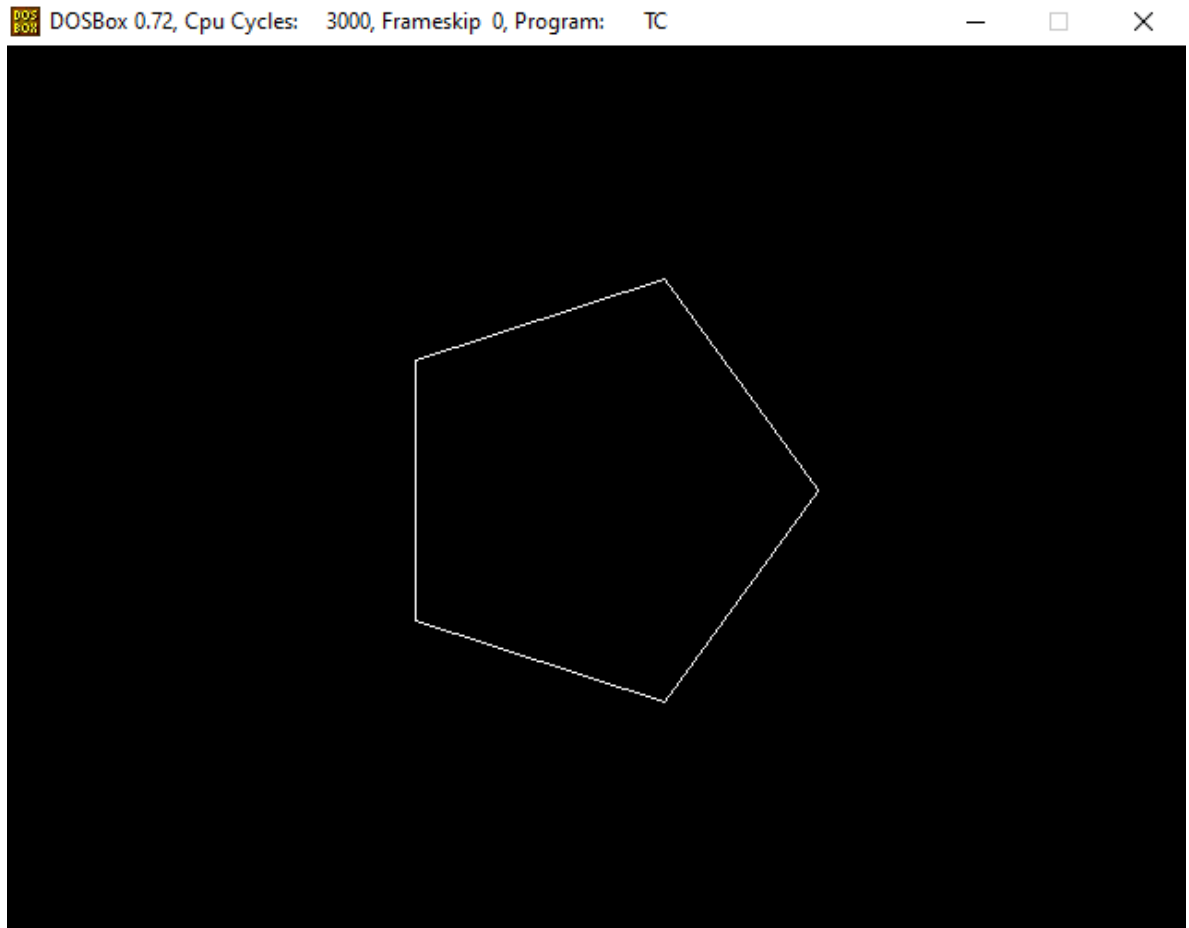
int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver,&g_mode,"...\\bgi");

    drawregpoly(320, 240, 120, 5);

    getch();
    closegraph();
    return 0;
}
```

El código de arriba dibuja un pentágono vacío con centro en (320, 240), cuyo radio es de 120 píxeles.

Salida:



1.1.2. Fillregpoly:

Sintaxis: fillregpoly(x, y, radio, lados);

Prototipo: fillregpoly(int x, int y, int radius, int sides)

Descripción: La función dibuja un polígono regular **relleno** con centro en las coordenadas (x, y). El segundo parámetro indica la distancia en píxeles desde el centro hasta un vértice cualquiera y el tercer parámetro indica el número de lados, hasta un máximo de 30. Cabe destacar que el polígono se rellenará con el color del estilo de relleno actual, y el color del delineado estará dado por el color actual.

Argumento	Descripción
<i>x</i>	La coordenada en x del polígono.
<i>y</i>	La coordenada en y del polígono.
<i>radio</i>	La distancia entre el centro de la figura (indicado por el par ordenado [x,y]) y cualquiera de sus vértices.

<i>lados</i>	El número de lados que posee el polígono (abarca desde un mínimo de 3 lados a un máximo de 30)
---------------------	--

Ejemplo:

Entrada

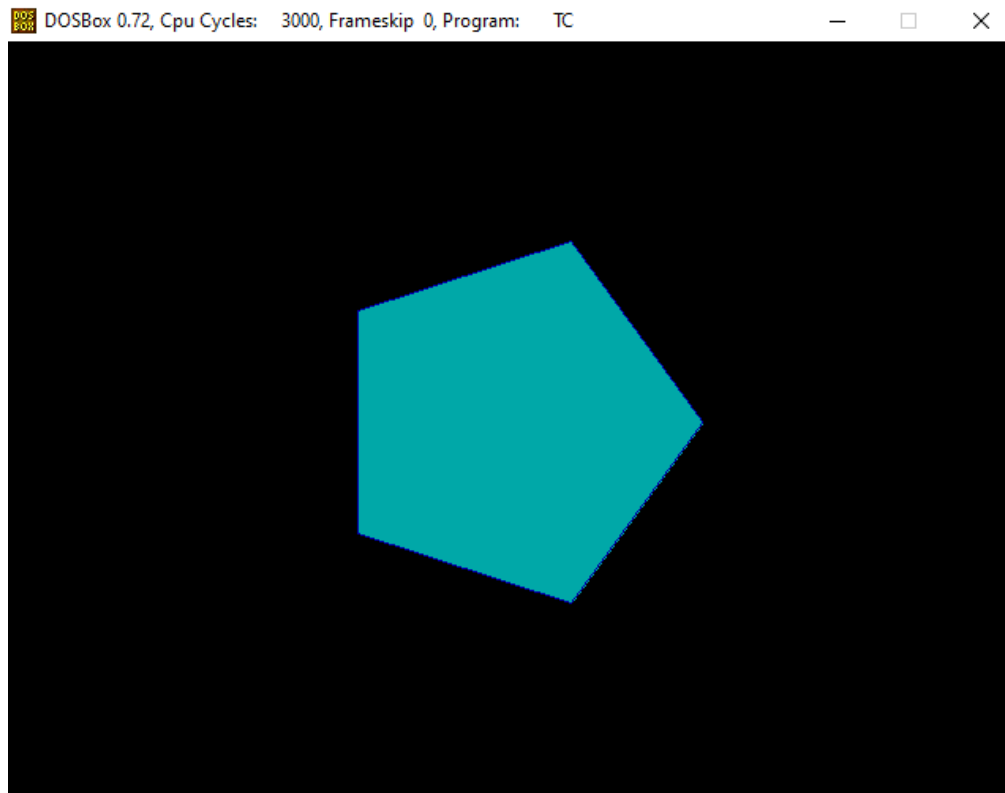
```
#include <graphics.h>
#include <POLYGON.h>
#include <conio.h>

int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver,&g_mode,"...\\bgi");

    setfillstyle(SOLID_FILL,CYAN);
    setcolor(BLUE);
    fillregpoly(320, 240, 120, 5);

    getch();
    closegraph();
    return 0;
}
```

Salida



1.1.3. Drawpolyinellipse:

Sintaxis: drawpolyinellipse(x, y, radiox, radioy, lados);

Prototipo: drawpolyinellipse(int x, int y, int xradius, int yradius, int sides)

Descripción: La función dibuja un polígono regular circunscrito en una elipse con centro en (x, y). El par ordenado del tercer y cuarto parámetro indica el radio que poseerá la elipse en los ejes **x** y **y** respectivamente y, finalmente, el quinto parámetro indica el número de lados que tendrá la figura.

Argumento	Descripción
x	La coordenada en x del polígono.
y	La coordenada en y del polígono.
radiox	El radio del polígono con respecto al eje x .
radioy	El radio del polígono con respecto al eje y .
lados	El número de lados que posee el polígono (abarca desde un mínimo de 3 lados a un máximo de 30)

Ejemplo:

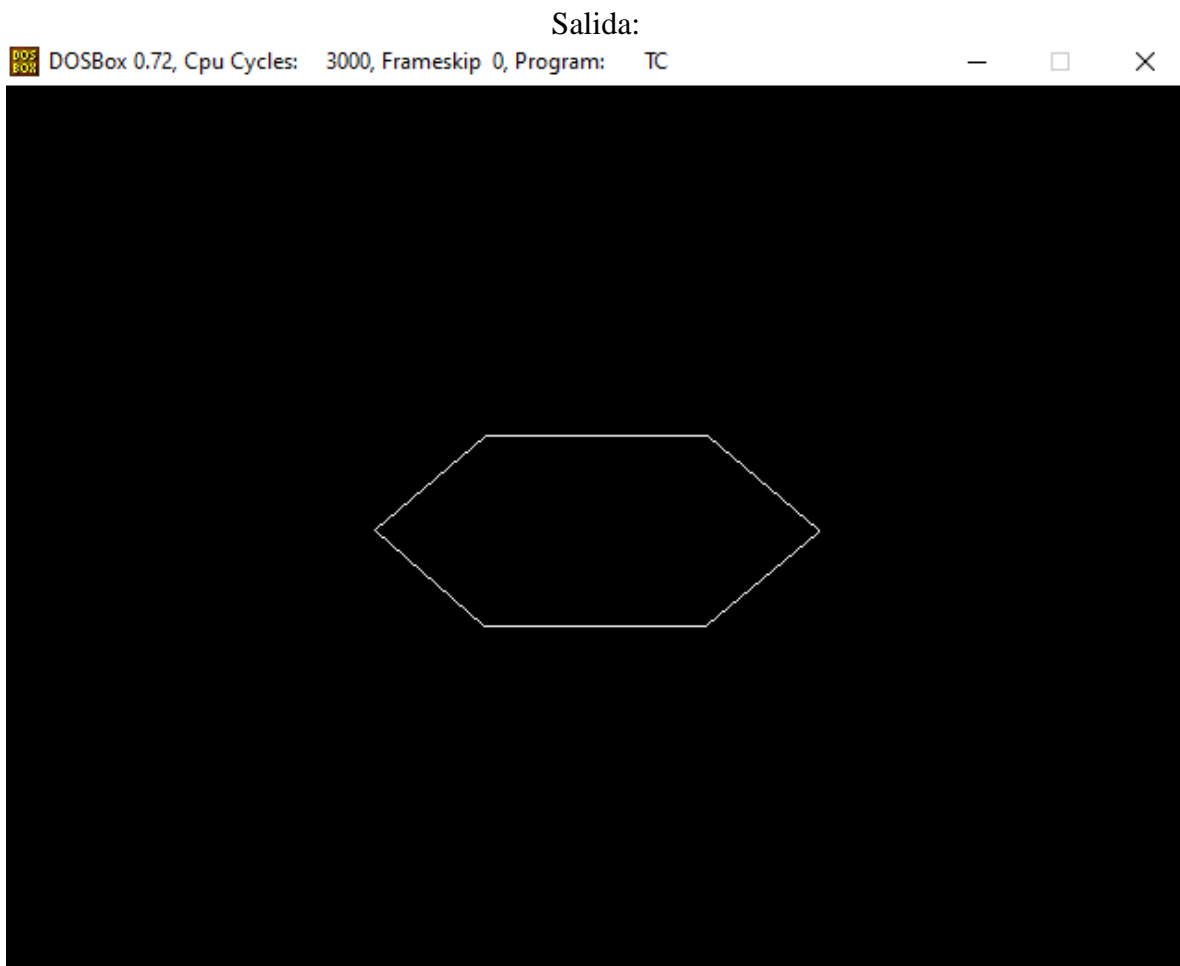
Entrada

```
int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver, &g_mode, "...\\bgi");

    drawpolyinellipse(320, 240, 120, 60, 6);

    getch();
    closegraph();
    return 0;
}
```

*La función de arriba dibuja un polígono regular con centro en (320, 240), circunscrito en una elipse con radios de 120 y 60 píxeles con respecto a los ejes **x** y **y** respectivamente.*



1.1.4. Drawpolyinellipse:

Sintaxis: fillpolyinellipse(*x*, *y*, *radiox*, *radioy*, *lados*);

Prototipo: fillpolyinellipse(int *x*, int *y*, int *xradius*, int *yradius*, int *sides*)

Descripción: La función dibuja un polígono regular **relleno** circunscrito en una elipse con centro en (*x*, *y*). El par ordenado del tercer y cuarto parámetro indica el radio que poseerá la elipse en los ejes *x* y *y* respectivamente y, finalmente, el quinto parámetro indica el número de lados que tendrá la figura. Cabe señalar que el color del relleno estará indicado por el patrón de relleno actual y el color de delineado estará dado por el color actual.

Argumento	Descripción
<i>x</i>	La coordenada en <i>x</i> del polígono.
<i>y</i>	La coordenada en <i>y</i> del polígono.
<i>radiox</i>	El radio del polígono con respecto al eje <i>x</i> .

<i>radioy</i>	El radio del polígono con respecto al eje y .
<i>lados</i>	El número de lados que posee el polígono (abarca desde un mínimo de 3 lados a un máximo de 30)

Ejemplo:

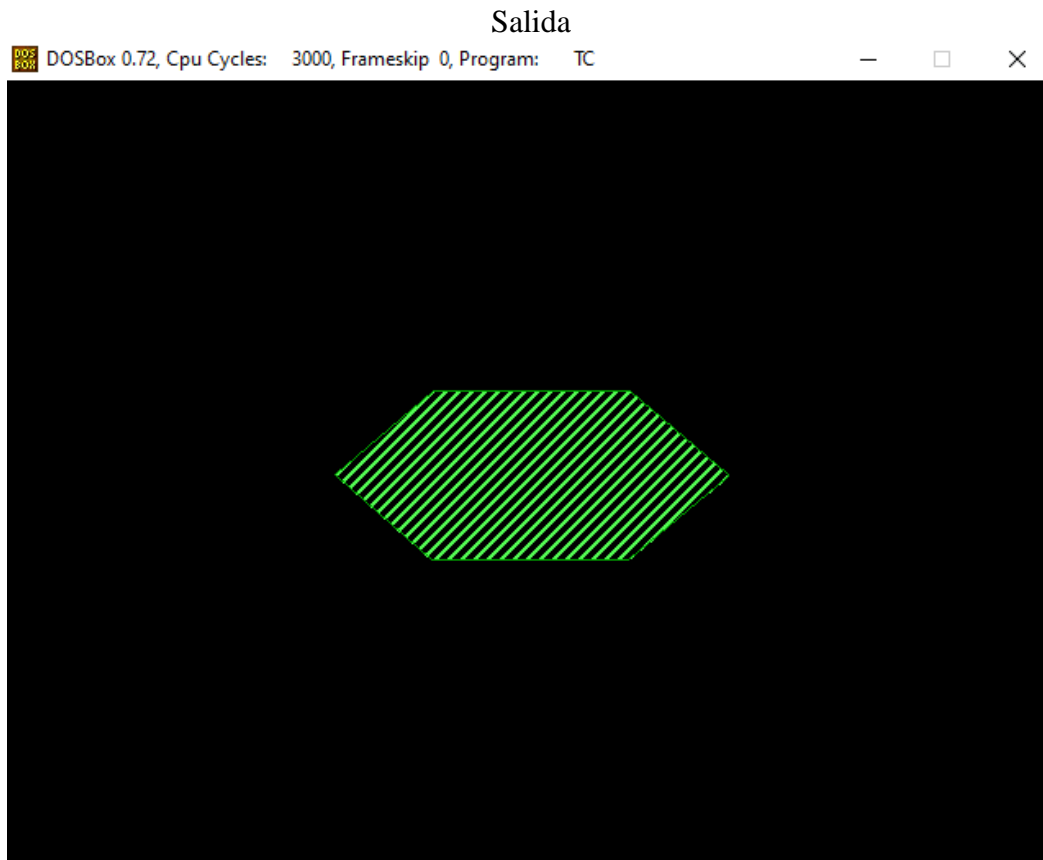
Entrada

```
int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver,&g_mode,"...\\bgi");

    setcolor(GREEN);
    setfillstyle(SLASH_FILL, LIGHTGREEN);
    fillpolyinellipse(320, 240, 120, 60, 6);

    getch();
    closegraph();
    return 0;
}
```

*El código de arriba dibuja un hexágono regular circunscrito en una elipse con centro en (x, y). Sus radios con respecto a los ejes **x** y **y** son de 120 y 60 píxeles respectivamente. El color del delineado está indicado por la función **setcolor** y el patrón de relleno por la función **setfillstyle**.*



1.1.5. Drawpolyinrectangle:

Sintaxis: drawpolyinrectangle(xinicial, yinicial, xfinal, yfinal, lados, inclinación, puntoInicio, puntoFinal);

Prototipo: drawpolyinrectangle(int x1, int y1, int x2, int y2, int sides, double inclination, int startPoint, int endPoint)

Descripción: La función dibuja un polígono regular circunscrito en un rectángulo. Los dos primeros pares ordenados de parámetros indican la posición inicial y final del rectángulo en el plano coordenado. El resto de parámetros indican el número de lados que contiene la figura, así como la inclinación de la misma.

Argumento	Descripción
<i>xinicial</i>	La coordenada en <i>x</i> del punto inicial del rectángulo.
<i>yinicial</i>	La coordenada en <i>y</i> del punto inicial del rectángulo.
<i>xfinal</i>	La coordenada en <i>x</i> del punto final del rectángulo.
<i>yfinal</i>	La coordenada en <i>y</i> del punto final del rectángulo.

<i>lados</i>	El número de lados que posee el polígono (abarca desde un mínimo de 3 lados a un máximo de 30).
<i>inclinación</i>	La inclinación que poseerá la figura, expresada en grados.
<i>puntoInicio</i>	El punto del que partirá el trazo de la figura.
<i>puntoFinal</i>	El punto en que terminará el trazo de la figura

Ejemplo:

Entrada

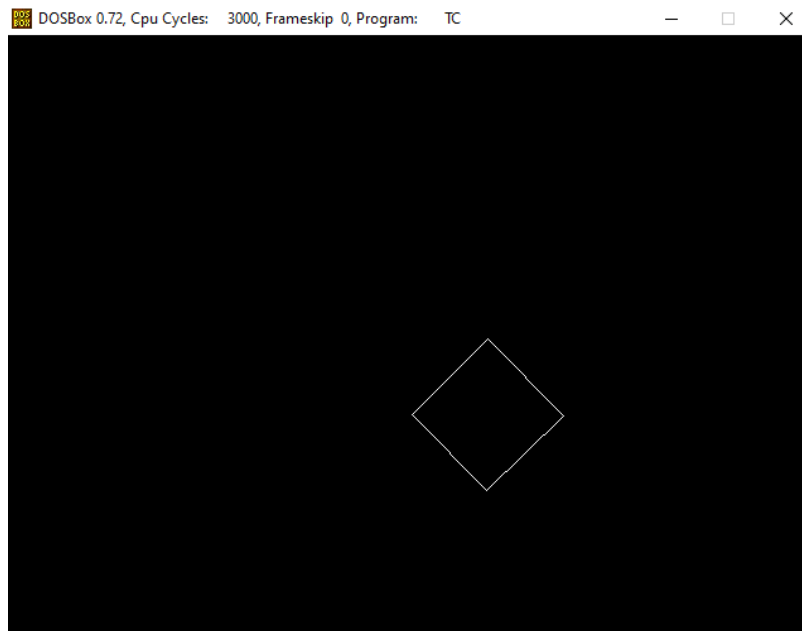
```
int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver,&g_mode,"...\\bgi");

    drawpolyinrectangle(320, 240, 440, 360, 4, 90, 0, 5);

    getch();
    closegraph();
    return 0;
}
```

El código de arriba dibuja un cuadrado regular con una inclinación de 90 grados. Los primeros dos pares ordenados de parámetros lo delimitan dentro de un cuadrado.

Salida



1.1.5. Fillpolyinrectangle:

Sintaxis: fillpolyinrectangle(xinicial, yinicial, xfinal, yfinal, lados, inclinación, puntoInicio, puntoFinal);

Prototipo: drawpolyinrectangle(int x1, int y1, int x2, int y2, int sides, double inclination, int startPoint, int endPoint)

Descripción: La función dibuja un polígono regular **relleno** circunscrito en un rectángulo. Los dos primeros pares ordenados de parámetros indican la posición inicial y final del rectángulo en el plano coordenado. El resto de parámetros indican el número de lados que contiene la figura, así como la inclinación de la misma. Cabe señalar que el color del relleno de la figura está indicada por el patrón de relleno actual y el color del delineado por el color actual.

Argumento	Descripción
<i>xinicial</i>	La coordenada en <i>x</i> del punto inicial del rectángulo.
<i>yinicial</i>	La coordenada en <i>y</i> del punto inicial del rectángulo.
<i>xfinal</i>	La coordenada en <i>x</i> del punto final del rectángulo.
<i>yfinal</i>	La coordenada en <i>y</i> del punto final del rectángulo.
<i>lados</i>	El número de lados que poseerá el polígono (abarca desde un mínimo de 3 lados a un máximo de 30).
<i>inclinación</i>	La inclinación que poseerá la figura, expresada en grados.

Ejemplo:

Entrada

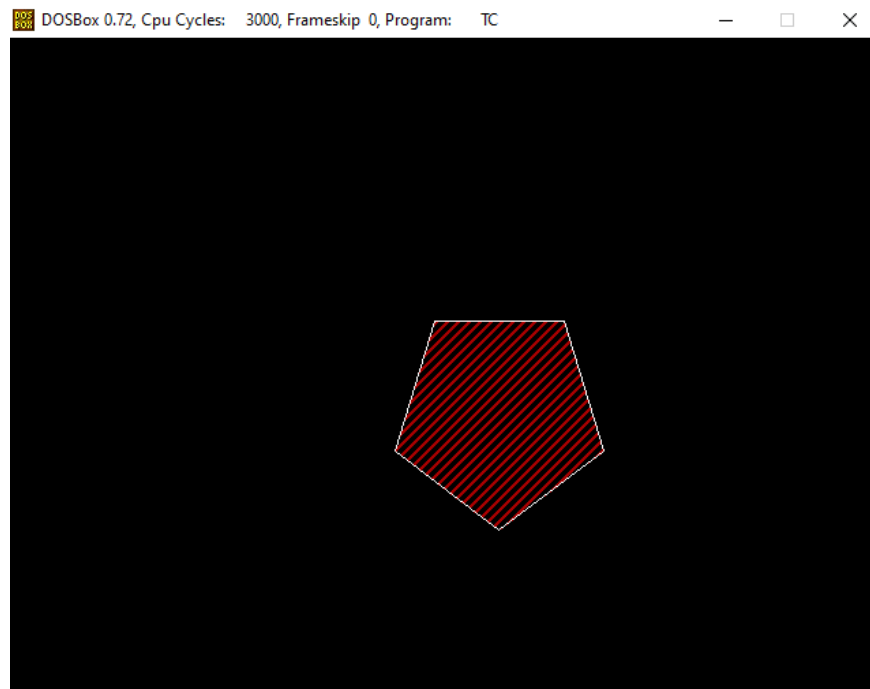
```
int main()
{
    int g_driver = DETECT, g_mode = DETECT;
    initgraph(&g_driver,&g_mode,"...\\bgi");

    setcolor(WHITE);
    setfillstyle(4, RED);

    fillpolyinrectangle(280, 190, 440, 360, 5, 180);

    getch();
    closegraph();
    return 0;
}
```

Salida



Contáctanos: <https://discord.gg/2qd63aq>