

## Tarea de computación

### Aleatorio (random):

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

int x;
main ()
{
for (x=1; x<6; x++)
printf ("%d\n", rand());
getch();
return (
}
```

```
41
18467
6334
26500
19169
```

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
main()
{
int x, i;
{
srand(1);

for (i=0; i<10; i++)
{
x=rand();
x=x/1000;
printf ("%d\n", x);}}
getch();
return 0;
}
```

```
CA
18
10
11
29
18
13
11
5
30
23
```

## Cuadrato móvil:

```
else
{
    /* OpenGL animation code goes here */

    glClearColor (1.0f, 1.0f, 1.0f, 0.0f);

    glClear (GL_COLOR_BUFFER_BIT);

    glPushMatrix ();
    //glRotatef (theta, 0.0f, 0.0f, 1.0f);
    glBegin (GL_LINES);
    glColor3f (1.0f, 0.0f, 1.0f);    glVertex2f (-0.89f, -0.89f);
    glColor3f (0.0f, 1.0f, 1.0f);    glVertex2f (0.89f, -0.89f);
    glEnd ();
    glBegin (GL_LINES);
    glColor3f (1.0f, 0.0f, 1.0f);    glVertex2f (0.89f, -0.89f);
    glColor3f (0.0f, 1.0f, 1.0f);    glVertex2f (0.89f, 0.89f);
    glEnd ();
    glBegin (GL_LINES);
    glColor3f (1.0f, 0.0f, 1.0f);    glVertex2f (0.89f, 0.89f);
    glColor3f (0.0f, 1.0f, 1.0f);    glVertex2f (-0.89f, 0.89f);
    glEnd ();
    glBegin (GL_LINES);
    glColor3f (1.0f, 0.0f, 1.0f);    glVertex2f (-0.89f, 0.89f);
    glColor3f (0.0f, 1.0f, 1.0f);    glVertex2f (-0.89f, -0.89f);
    glEnd ();
    glPopMatrix ();
glPushMatrix ();
```

```
glPushMatrix ();
//glRotatef (theta, 0.0f, 0.0f, 1.0f);
x=-0.30;
y=0.2;

k=rand();
y=x/100000;
x=k/100000;

glBegin (GL_LINES);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.30f, y+0.5f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.60f, y+0.5f);
glEnd ();
glBegin (GL_LINES);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.60f, y+0.89f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.60f, y+0.2f);
glEnd ();
glBegin (GL_LINES);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.60f, y+0.2f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.30f, y+0.3f);
glEnd ();
glBegin (GL_LINES);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.30f, y+0.3f);
glColor3f (0.0f, 1.0f, 0.0f);    glVertex2f (x-0.30f, y+0.5f);
glEnd ();
glPopMatrix ();

SwapBuffers (hDC);

theta += 1.0f;
```

