

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

typedef int TipoDato;

#include "pilaarray.h"

void main()
{

Pila P;
int x,aux;

CrearPila(&P);
scanf("%d",&x);

Insertar(&P,x);
printf("%d \n",Cima(P));

if(!PilaVacía(P))
aux=Quitar(&P);

printf("%d \n",aux);

LimpiarPila(&P);

}
```

```
#include <stdio.h>
#include <stdlib.h>
#define MaxTamaPila 100

typedef struct
{
    TipoDato listapila[MaxTamaPila];
    int cima;
}Pila;

/* Operaciones sobre la Pila */

void CrearPila(Pila* P);
void Insertar(Pila* P, const TipoDato elemento);
TipoDato Quitar(Pila *P);
void LimpiarPila(Pila* P);
void Pop(Pila* P, TipoDato elemento);

/* Operacion de acceso a Pila */

TipoDato Cima(Pila P);

/* verificacion estado de la Pila */

int PilaVacía(Pila P);
int PilaLlena(Pila P);

/* Funciones */

void CrearPila(Pila* P)
{
    P -> cima = -1;
}

void Insertar(Pila* P, const TipoDato elemento)
{
    if (P->cima == MaxTamaPila-1)
    {
        puts("Desbordamiento pila");
        exit (1);
    }
    P->cima++;
    P->listapila[P->cima] = elemento;
}

void Pop(Pila* P, TipoDato elemento)
{
    Insertar(P, elemento);
}

TipoDato Quitar(Pila *P)
{
    TipoDato aux;

    if (P->cima == -1)
```

```
{
    puts("Se intenta sacar un elemento en pila vacía");
    exit (1);
}
aux = P->listapila[P->cima];
P->cima--;
return aux;
}

TipoDato Cima(Pila P)
{
    TipoDato aux;
    if (PilaVacía(P))
    {
        puts("Se intenta sacar un elemento en pila vacía");
        exit (1);
    }
    aux = P.listapila[P.cima];
    return aux;
}

void LimpiarPila(Pila* P)
{
    P->cima = -1;
}

int PilaVacía(Pila P)
{
    return P.cima == -1;
}

int PilaLlena(Pila P)
{
    return P.cima == MaxTamaPila-1;
}
```

# Gráfica 3D

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

main()
{
    clrscr();

    char opcion;
    float x, y, z, ax, ay, az, tx, ty, tz, gx, gy, cx, cy;
    int m, n, a;
    m=9; n=2;
    x=0; y=0; z=0; ax=0; ay=0; az=0; tx=0; ty=0; tz=0; gx=0; gy=0; cx=0; cy=0;
    printf("\tPresiona x, y ó z para manipular el gráfico, s para terminar.")
;

    getch();
    initgraph(&m, &n, "c:\\tc\\bgi");
    do{
        for(x=-3; x<=3; x+=10.0/640.0)
        {
            y=cos(x);
            tx=cos(az)*(x*cos(ay)+(y*sin(ax)+z*cos(ax))*sin(ay))-sin(az)*(y*cos(ax)
)-z*sin(ax));
            ty=sin(az)*(x*cos(ay)+(y*sin(ax)+z*cos(ax))*sin(ay))+cos(az)*(y*cos(ax)
)-z*sin(ax));
            tz=-x*sin(ay)+(y*sin(ax)+z*cos(ax))*cos(ay);
            gx=ty-tx; gy=tz-tx;
            cx=(320/10)*gx+320;
            cy=240-(240/10)*gy;
            putpixel(cx, cy, WHITE);
        }
        // opcion=getch();
        switch(opcion=getch())
        {
            case 'x': ax=ax+0.5;
                break;
            case 'y': ay=ay+0.5;
                break;
            case 'z': az=az+0.5;
                break;
            case 's': a=1;
                break;
            default: printf("No existe tal cosa.");
                break;
        }
    }while(a != 1 );
    closegraph();
    return ;
}
```