

## Código.

```
#include<stdio.h>
#include<windows.h>
#include<stdlib.h>
struct in
{
    int v3[4];
};

void contrasena(FILE *, FILE*, FILE*, int [], struct in []);
void cambio (FILE*, FILE *, struct in []);
void cambio2(FILE*);
void correcto(FILE*, FILE *, int [], struct in [], int );
void incorrecto(FILE *, FILE* ,struct in [], int, int[], int[]);
int main()
{
    struct in *x;
    int *v, b;
    int i;
    FILE *con, *inc, *puk, *incc;
    errno_t err, err2, err3, err4;
    v = (int*)calloc(4, sizeof(int));
    x = (in*)calloc(3, sizeof(int));
    err = fopen_s(&con, "contrasena.txt", "r+");
    err2 = fopen_s(&inc, "incorrecto.txt", "w+");
    err3 = fopen_s(&puk, "puk.txt", "a+");
    err4 = fopen_s(&incc, "incorrectoc.txt", "w+");
    do
    {
        printf("OPCIONES:\n1.INTRODUZCA SU CONTRASENA PARA ABRIR LA
PUERTA.\n2.CAMBIODECONTRASENA\n");
        scanf_s("%d", &b);
        if (b != 1 && b != 2) {

            printf("Opcion no disponible.\nVuelva a elegir una opcion\n");
            getchar();
            getchar();
            system("cls");
        }
    } while (b != 1 && b != 2);
    switch (b)
    {
    case 1:
    {
        contrasena(con, inc, puk, v, x);

        }break;
    case 2:
    {
        cambio(con,incc,x);
    }break;
    }
    getchar();

    getchar();
}
```

```

void contrasena(FILE*pf, FILE*pt, FILE*px, int *v, struct in *x)
{
    int i = 0, k = 0, b, c = 3, j;
    getchar();
    system("cls");

    for (i = 0; i < 4; i++)
    {
        fscanf_s(pf, "%d", (v + i));
    }

    fclose(pf);
    printf("Introduzca su contraseña\n");
    i = 0;
    do
    {
        k = 0;
        for (j = 0; j < 4; j++)
        {
            scanf_s("%d", &(x + i)->v3[j]);
        }
        for (j = 0; j < 4; j++)
        {
            if ((x + i)->v3[j] == *(v + j))
            {
                k++;
            }
        }
        if (k == 4)
            printf("\nContrasena correcta\n");
        else
        {
            printf("\nContrasena incorrecta\n");
            i++;
        }
    } while (k != 4 && i != 3);
    int *v0, *v1;
    v0 = (int*)calloc(4, sizeof(int));
    v1 = (int*)calloc(4, sizeof(int));
    if (i == 3)
        incorrecto(pt, px, x, i, v0, v1);
    if (i < 3)
        correcto(pf, pt, v, x, i);
}

void correcto(FILE *pf, FILE *pt, int *v, struct in *x, int a)
{
    int i, j, c;
    getchar();
    getchar();
    system("cls");
    for (i = 0; i < a; i++)
    {
        fprintf(pt, "Contraseña incorrecta %d:\t", i);
        for (j = 0; j < 4; j++)
        {
            fprintf(pt, "%d ", i, (x + i)->v3[j]);
        }
    }
}

```

```

        fprintf(pt, "\n");
    }
    fclose(pt);
    printf("Puerta abierta\n\n");

do{
    printf("Pulse 1 para cerrar la puerta\nPulse 2 para cambiar la
contrasena\n\n");
    scanf_s("%d", &c);
    if (c != 1 && c != 2)
    {
        printf("Opcion no valida\n");
        getchar();
        getchar();
        system("cls");
    }
}while(c != 1 && c!=2);
if (c == 2) {

    cambio2(pf);
}
else
printf("\n\nCerrando puerta\n");
}
void incorrecto(FILE *pt, FILE *px, struct in *x, int a, int *v0, int *v1)
{
    int i, j, b = 0;
    getchar();
    getchar();
    system("cls");
    for (i = 0; i < a; i++)
    {
        fprintf(pt, "Contraseña incorrecta %d:\t", i);
        for (j = 0; j < 4; j++)
        {
            fprintf(pt, "%d ", (x + i)->v3[j]);
        }
        fprintf(pt, "\n");
    }
    fclose(pt);
    for (i = 0; i < 4; i++)
    {
        fscanf_s(px, "%d", v0 + i);
    }
    fclose(px);
    printf("\nContrasena incorrecta\n\nPuerta bloqueada\n\n");
    printf("\nIntroduzca codigo alternativo");
    for (i = 0; i < 4; i++)
    {
        scanf_s("%d", v1 + i);
    }
    for (i = 0; i < 4; i++)
    {
        if (*(v0 + i) == *(v1 + i))
            b++;
    }
    if (b == 4) {
        getchar();
    }
}

```

```

        getchar();
        system("cls");
        printf("\ncodigo de seguridad correcto\n\n\nbriendo puerta\n");
    }

    else
    {
        getchar();
        getchar();
        system("cls");
        printf("\ncodigo de seguridad incorrecto\n\npuerta
bloqueada\n\nProhibido el uso de la caja\n");
    }
}

void cambio2(FILE *pf)
{
    int i, j, *v, a;
    getchar();
    system("cls");
    v = (int*)calloc(4, sizeof(int));
    errno_t err;
    err = fopen_s(&pf, "contrasena.txt", "w+");
    printf("\nIntroduzca la nueva contraseña\n");
    for (i = 0; i < 4; i++)
    {
        scanf_s("%d", v + i);
    }
    for (i = 0; i < 4; i++)
    {
        fprintf(pf, "%d ", *(v + i));
    }
    fclose(pf);
    printf("Contrasena cambiada\n");

    do {
        printf("Presione 1 para cerrar la puerta\n");
        scanf_s("%d", &a);
        if(a!=1)
        {

            getchar();
            system("cls");

        }
    } while (a != 1);
    if (a == 1)
        printf("\n\nCerrando puerta");
}

void cambio(FILE *pf, FILE *inn, struct in *x)
{
    int i, j, *v, *v1, *v2, a = 0, k = 0;
    getchar();
    getchar();
    system("cls");
    v = (int*)calloc(4, sizeof(int));
    v2 = (int*)calloc(4, sizeof(int));
    errno_t err;
    printf("Introduzca la contraseña anterior\n");

```

```

for (i = 0; i < 4; i++)
{
    fscanf_s(pf, "%d", (v + i));
}
printf("\n");
fclose(pf);
i = 0;
do
{
    k = 0;
    for (j = 0; j < 4; j++) {
        scanf_s("%d", &(x + i)->v3[j]);
        if ((x + i)->v3[j] == *(v + j))
        {
            k++;
        }
    }
    if (k == 4)
    {
        getchar();
        system("cls");
        err = fopen_s(&pf, "contrasena.txt", "w+");
        printf("Introduzca la nueva contraseña\n");
        getchar();
        for (i = 0; i < 4; i++)
            scanf_s("%d", v2 + i);
        getchar();
        for (i = 0; i < 4; i++)
            fprintf(pf, "%d ", *(v2 + i));
        fclose(pf);
        printf("\n\nContrasena cambiada correctamente\n\n");
    }
    else {
        printf("\nContrasena incorrecta\n");
        i++;
        a++;
    }
}
while (k != 4 && a < 2);
for (i = 0; i < a; i++)
{
    fprintf(inn, "contrasena incorrecta %d:", i);
    for (j = 0; j < 4; j++)
        fprintf(inn, "%d ", (x + i)->v3[j]);
    fprintf(inn, "\n");
}
fclose(inn);
if (a == 2)
    printf("\n\nImposible cambiar contrasena");
}

```

## Funciones.

`int main()`

- En el main declaramos un vector de números enteros que se utilizará en la función “contrasena”, los ficheros, el vector de estructuras y hacemos la llamada a las funciones “contrasena” y “cambio”.
- Nos da las opciones de introducir la contraseña para abrir la puerta, pasando a la función “contrasena” o realizar el cambio de la contraseña pasando a la función “cambio”.

`void contrasena(FILE *, FILE*, FILE*, int [], struct in []);`

- Con esta función pedimos al usuario que introduzca la contraseña. Un vez introducida la comparamos con la correcta, si esta es correcta pasamos a la función correcto, en caso contrario, y tras tres intentos, a la función incorrecto.
- Para realizar estas el análisis de la contraseña introducida, pasamos a la memoria RAM la contraseña que se encuentra en el fichero “contrasena”.
- En el vector de números enteros almacenamos la contraseña del fichero, para compararla con la que el usuario introduzca.
- Los otros dos ficheros se utilizan para llamarlos desde la función “incorrecto” y “correcto”.
- En el vector de estructuras guardamos las contraseñas erróneas.
- Declaramos dos vectores, mediante asignación dinámica de memoria, que se usarán en la función “incorrecto”.

`void correcto(FILE*, FILE *, int [], struct in [], int );`

- Pasamos a la función “correcto” una vez introduzcamos la contraseña correcta en la función anterior.
- Esta función se encarga de abrir la puerta, y nos da la opción de cerrar la puerta o cambiar la contraseña sin pedirnos la anterior ya que la puerta está abierta, pasando a la función “cambio2”..
- Introducimos los ficheros “incorrecto” y “contrasena”.
- En “incorrecto” guardamos los intentos erróneos de introducción de la contraseña mediante el vector de estructuras, imprimiendo en la memoria ROM lo que escribimos en la RAM.
- El fichero “contrasena” será utilizado en “cambio2”

`void cambio2(FILE*);`

- En esta función realizamos el cambio de contraseña.
- Declaramos un vector, usando asignación dinámica de memoria, que almacene la nueva contraseña y eventualmente esta reescribirá la anterior en el fichero “contrasena”

```
void incorrecto(FILE *, FILE* ,struct in [], int, int[], int[]);
```

- Esta función se ejecuta una vez fallemos los tres intentos de introducción de la contraseña en la función “contrasena”.
- Aquí se guardarán las contraseñas erróneas al igual que en la función correcto, usando el vector de estructuras y almacenándolas en el fichero “incorrecto”.
- Al fallar las tres veces, la caja fuerte te pedirá la contraseña de seguridad, almacenada en el fichero “puk”, en caso de que la contraseña que introduzcamos sea errónea se bloqueará la caja fuerte, por otra parte, si es correcto la puerta se abrirá.
- Para comparar la contraseña introducida con la del fichero, usaremos los vectores declarados en la función “contrasena”. Uno se usará para guardar la contraseña introducida y el otro para guardar la almacenar la del fichero. Al compararlas descubriremos si la introducida es correcta o errónea.

```
void cambio (FILE*, FILE *, struct in []);
```

- Entramos en esta función al elegir la opción en el main.
- Esta se encarga del cambio de contraseña tras habernos pedido la contraseña inicial, si es correcta, se nos permite cambiarla, en caso contrario y tras dos intentos no se permite el cambio,
- Las contraseñas erróneas se almacenarán en el fichero “incorrectoc” de la misma manera que lo hemos explicado con anterioridad.
- Cambiaremos la contraseña al introducir correctamente la inicial de la misma manera que en la función “cambio2”.

# Flujograma.





