

Ejercicios 4-Iniciación a la Programación

1. Un programa de edición de imágenes permite que el usuario diseñe iconos utilizando la siguiente interfaz de usuario:

Introduzca anchura/altura del icono en píxeles [anchura,alto]:

Introduzca el nivel de gris del píxel (x,y):

El programa acepta enteros para la anchura y la altura del icono, y solicita al usuario los niveles de gris (brillo) de cada píxel (valores entre 0 y 255, ambos incluidos).

Escriba dicho programa en lenguaje C. Utilice una función para imprimir los niveles de gris de los píxeles de la imagen.

2. Escribe un programa que utilice una estructura de datos, de nombre *clientes*, que incluya tres campos:

- Entero con la edad de un clientes
- Puntero a carácter para almacenar hasta las primeras 20 letras del nombre del cliente
- Puntero adecuado para generar una lista de clientes

El programa ha de pedir al usuario la edad del cliente, así como las 20 primeras letras de su nombre (`scanf("%s", (char *))`) y almacenarlos en la estructura de datos. A continuación, se muestra un mensaje como el siguiente, con el nombre y edad del cliente:

Mi cliente ***** tiene *** años

Nota: no olvides inicializar el puntero para la lista de forma adecuada.

3. Escribe una función, *producto_escalar*, que reciba como parámetros de entrada dos punteros a float, que indican la dirección del primer elemento de los vectores X e Y, respectivamente, así como un entero con la longitud de esos vectores. La función devolverá el valor del producto escalar de ambos vectores. Nota: recuerde comprobar que los punteros estén inicializados (`!=NULL`).

4. Escribe una función, *mayor_que*, similar a la del ejercicio 2, pero que devuelva un vector, Z, (puntero a entero), cuyos elementos son 1 si es mayor el correspondiente de X que el de Y, y 0 en caso contrario.

Ejercicios 4-Iniciación a la Programación

5. Escribe una función, *reserva_matriz_mxn*, que reciba dos números enteros, *m* y *n*, como parámetros de entrada y que devuelva un puntero a una zona de memoria en la que se pueden almacenar *m* vectores de *n* elementos enteros cada uno (es decir, una matriz de *m* filas y *n* columnas). El valor de los elementos de cada fila se debe inicializar en esa función a un valor igual al número de fila.

6. Escribe un programa que gestione una lista enlazada de números complejos ordenados de mayor a menor módulo. El programa permitirá insertar y borrar nodos, así como visualizar el contenido de toda la lista.