PISTAS.

1. Pista 1: Crea un objeto std::deque vacío para almacenar los datos y pide al usuario que ingrese los elementos. Puedes utilizar un bucle while para leer los elementos hasta que el usuario ingrese una cadena vacía (es decir, presione enter sin ingresar ningún valor).
2. Pista 2: Implementa una función para insertar un elemento en una posición específica del deque. Para hacerlo, utiliza el método insert() de std::deque. Recuerda que la posición de un elemento en el deque se cuenta desde cero.
3. Pista 3: Implementa una función para eliminar un elemento de una posición específica del deque. Para hacerlo, utiliza el método erase() de std::deque. Al igual que en el caso anterior, recuerda que la posición de un elemento se cuenta desde cero.
4. Pista 4: Implementa una función para ordenar los elementos del deque en orden ascendente. Para hacerlo, puedes utilizar la función std::sort() de la biblioteca estándar de C++. Ten en cuenta que std::sort() requiere que el rango a ordenar sea un contenedor con iteradores aleatorios, por lo que es posible que necesites convertir tu deque en un vector temporalmente.
5. Pista 5: Implementa una función para mostrar el contenido del deque por pantalla. Puedes utilizar un bucle for para recorrer el deque y mostrar cada elemento. También puedes utilizar la función std::copy() de la biblioteca estándar de C++ para copiar los elementos del deque en un flujo de salida, como std::cout.