1. Comience por escribir la plantilla de clase genérica Stack que implemente una pila de elementos. Incluya los métodos necesarios para agregar y eliminar elementos de la pila y para verificar si la pila está vacía.

template <typename T>

class Stack {

public:

Stack() { ... }

void push(T element) { ... }

T pop() { ... }

bool empty() { ... }

};

1. A continuación, agregue la especialización de la plantilla Stack para el tipo char, que implemente una pila de caracteres con una capacidad máxima de 100 elementos.

template<>

class Stack<char> {

private:

static const int max\_size = 100;

char elements[max\_size];

int top;

public:

Stack() : top(-1) { ... }

void push(char element) { ... }

char pop() { ... }

bool empty() { ... }

};

1. Luego, escriba un pequeño programa de prueba que instancie la plantilla Stack con diferentes tipos de datos, incluyendo char, y muestre el comportamiento de la clase con diferentes tipos de datos.

int main() {

Stack<int> intStack;

Stack<char> charStack;

...

return 0;

}

1. Pida a los estudiantes que compilen y ejecuten el programa para verificar que la plantilla Stack se comporta correctamente con diferentes tipos de datos.
2. Finalmente, pida a los estudiantes que agreguen más tipos de datos y más instancias de la plantilla Stack en el programa de prueba para asegurarse de que la plantilla Stack funciona correctamente con todos los tipos de datos que desean usar.