**1. STACK Class**

1.1 Private Members

* **char a[max]**: An array to store characters in the stack.
* **int top**: An integer to keep track of the top of the stack.

1.2 Constructor

* Initializes **top** to -1.

1.3 **push** Method

* Adds a character to the stack.

1.4 **reverse** Method

* Reverses the characters in the stack and prints the reversed string.

1.5 **convert** Method

* Converts the characters in the given string to lowercase (if they are uppercase) and removes non-alphabetic characters.
* Modifies the input string in-place and prints the converted string.

1.6 **palindrome** Method

* Checks whether the original string is a palindrome (reads the same backward as forward).
* Prints whether the string is a palindrome or not.

**2. main Function**

2.1 Stack Initialization

* Creates an instance of the **STACK** class named **stack**.

2.2 Input

* Accepts a string input from the user (limited to **max** characters).

2.3 String Conversion and Palindrome Check

* Calls the **convert** method on the stack to convert the string.
* Pushes each character of the string onto the stack.
* Calls the **palindrome** method to check if the string is a palindrome.

2.4 String Reversal

* Calls the **reverse** method on the stack to print the reversed string.

**Note:**

* The program uses a stack to reverse a string, convert it to lowercase, and check if it's a palindrome.
* The stack is implemented using an array.
* The **convert** method removes non-alphabetic characters and converts uppercase characters to lowercase.
* The **palindrome** method checks whether the original string and its reversed version are the same.
* The program provides a simple demonstration of string manipulation using a stack.

Algorithm:

1. \*Include Header Files:\*

- Include necessary header files like iostream and string.h.

2. \*Define Constants:\*

- Define a constant max with a value of 50 using #define.

3. \*\*Create Class STACK:\*\*

- Declare a class named STACK with private members:

- char a[max]: Array to store characters.

- int top: Top of the stack.

- Public member functions:

- push(char): Push a character onto the stack.

- reverse(): Reverse the characters in the stack and print the reversed string.

- convert(char[]): Convert the input string to lowercase and remove non-alphabetic characters.

- palindrome(): Check if the string is a palindrome.

4. \*Define Class Member Functions:\*

- push(char c): Increment top, assign the character to a[top], and null-terminate the string.

- reverse(): Create a temporary array str to store the reversed string, print the reversed string, and terminate with a newline.

- convert(char str[]): Convert the input string to lowercase and remove non-alphabetic characters. Print the converted string.

- palindrome(): Check if the reversed string is the same as the original string. Print whether the string is a palindrome or not.

5. \*\*Define main() Function:\*\*

- Create an instance of the STACK class named stack.

- Declare an array str[max] to store the input string.

- Prompt the user to enter a string to be reversed and checked for palindrome.

- Read the string using cin.getline().

- Call the convert function to convert the string to lowercase and remove non-alphabetic characters.

- Use a loop to push each character onto the stack.

- Call the palindrome function to check if the string is a palindrome.

- Call the reverse function to print the reversed string.

6. \*End of Program:\*

- End the program.