Week4_Stack Implementation

- **Explanation (algorithm):**

  Firstly, we check character by character in a sentence or line to identify whether it is string or character. In case of character, along the way push if it is kind of the open symbol into the stack, but if the close symbol appears we need to check if it has only close symbol without complement it is the extra closing. Another case is if it complements the top of stack or not, if yes pops it out and continues until the last character. After this repetition, if the stack is not empty yet it proves that the open symbol remains without closing. Lastly just alert user 'OK' if successfully or state the problem if failed.

- **Solution with codes**

```
void Checking(string line){
    ArrayStack<char> array;
    int pos;
    bool insideQuote = false;
    for(int i = 0; i < line.length(); i++){
        char temp = line[i];
        if ( temp == '\'') {
            insideQuote = !insideQuote;
            continue;
        }
        if ( insideQuote )
        {
            continue;
        }
```

Checking char: Using loop to scan whole line and work with each char of the string input until the last char.

```
if( isOpen(temp) ){
    array.push(temp);
    pos = i+1;
}
else if( isClose(temp) )
{
    if( array.isEmpty() ){
        cout<<"ERROR at pos=" << i+1 <<" reason = Extra-closing" << endl;
        return;
    }

    char forCheck = array.peek();
    if( !checkMatching(forCheck,temp) ){
        cout<<"ERROR at pos=" << i+1 << " reason = Mismatch"<< endl;
        return;
    }
    array.pop();
}
```

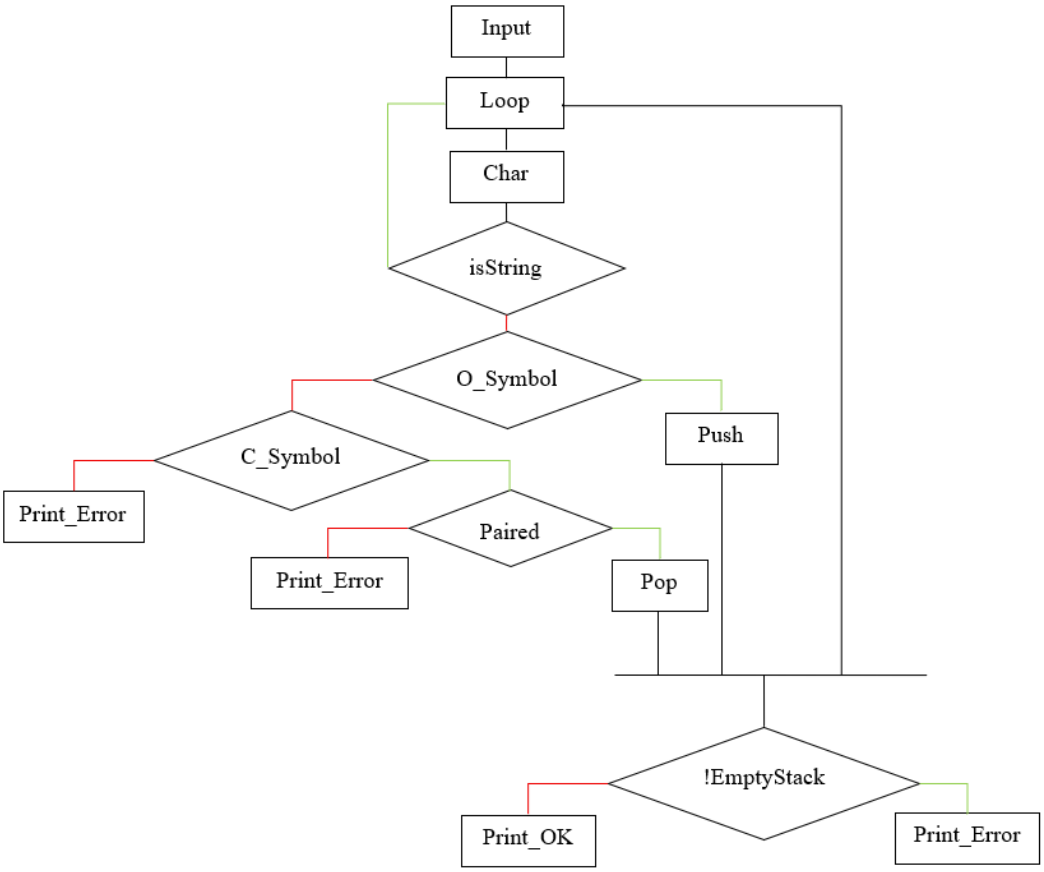
Store & compare: Push each open symbol into the stack and if it is close symbol just check if it complements by using the check function and pop it when already paired.

```
if ( !array.isEmpty() ) {
    cout << "ERROR at pos "<< pos << "reason = Unclosed" << endl;
} else {
    cout << "OK" << endl;
}
```

Result: If the stack still exists with element, it means that the open symbol remains without closing and print to user. Else it completes and compiled successfully when print OK.

Conclusion: This code able to handle most of the special characters so cases can be detected and alert users in separate cases of problem, in addition it can store more character, hence these operations are enough.

- **Flowchart**



Note: for condition
_ Redline: false
_ Greenline: true.

+ O_Symbol: Open_Symbol
+ C_Symbol: Close_Symbol