Remove Invalid Parenthesis in C++

```
#include <iostream>
#include <string>
#include <unordered_set>
#include <stack>
using namespace std;

void solution(string str, int mra,
unordered_set<string>& ans);
int getMin(string str);

void solution(string str, int mra,
unordered_set<string>& ans) {
    if (mra == 0) {
        int mrnow = getMin(str);
        if (mrnow == 0) {
            if (ans.find(str) == ans.end()) {
                cout << str << endl;
                ans.insert(str);
            }
        }
        return;
    }
    for (int i = 0; i < str.length(); i++) {
        string left = str.substr(0, i);
        string right = str.substr(i + 1);
        solution(left + right, mra - 1, ans);
    }
}

int getMin(string str) {
    stack<char> st;
    for (int i = 0; i < str.length(); i++) {
        char ch = str[i];
        if (ch == '(') {
            st.push(ch);
        } else if (ch == ')') {
            if (st.empty()) {
                st.push(ch);
            } else if (st.top() == ')') {
                st.push(ch);
            } else if (st.top() == '(') {
                st.pop();
            }
        }
    }
    return st.size();
}

int main() {
    string str = "((((()))";
    unordered_set<string> ans;
    int mra = getMin(str);
    solution(str, mra, ans);
    return 0;
}
```

Goal:

Remove the **minimum number** of parentheses to make the string valid.

🔧 Step 1: getMin("((((()))")

Step	Char	Stack	Action
1	(	(	push
2	(	((	push
3	(	(((	push
4	(	((((	push
5	(	(((((	push
6	)	((((	pop (match)
7	)	(((	pop (match)
8	)	((	pop (match)

□ Final stack size = ((→ **2 unmatched**
✅ So mra = 2 (Minimum Removals Allowed)

🔁 Step 2: Recursive Dry Run Table

We'll track:

Call #	Current String (str)	Removals Left (mra)	Action Taken	Is Valid (getMin=0)?	Output
1	((((()))	2	Start	❌ (getMin=2)	
2	(((())	1	Removed char at index 0	❌ (getMin=1)	
3	((()))	0	Removed char at index 0	✅ (getMin=0)	✅ ((()))
4	(same string)	0	Duplicate path	✅	(skipped by set)
...	many more paths tried	≤ 0	But not valid	❌	

✅ Only ((())) satisfies getMin == 0 **with exactly 2 removals**

	✅ Your unordered_set prevents printing duplicates 📜 **Final Output:** ((()))
Output:- ((()))	