```
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */
// stack_game.cpp
// A game using my stack implementation in C++.
// Author: Lauren E. Scott
// June 27, 2014
//
/* * * * * * * * * * * * * * * * * * * * * * * * * * * */

#include "stackcpp.h"

int calculate(Stack<string>* s, string input) {
    int result = 0;
    size_t sz;
    if(input == "+" || input == "-" || input == "/" || input == "*") {
        if(!(s->hasTwoElements())) {
            cout << "Stack is empty." << endl;
            return 1;
        } else {
            string top = s->pop();
            cout << "Top :" << top << endl;
            string second = s->pop();
            cout << "Second :" << second << endl;
            if(input == "+") {
                result = stoi(top) + stoi(second);
                cout << "Result: " << result << endl;
                s->push(to_string(result));
            } else if(input == "-") {
                result = stoi(top) - stoi(second);
                cout << "Result: " << result << endl;
                s->push(to_string(result));
            } else if (input == "/") {
                if (second == "0") {
                    cout << "Can't divide by 0. " <<endl;
                    return 1;
                } else {
                    result = stoi(top) / stoi(second);
                    cout << "Result: " << result << endl;
                    s->push(to_string(result));
                }
            } else if (input == "*") {
                result = stoi(top) * stoi(second);
                cout << "Result: " << result << endl;
                s->push(to_string(result));
            }
        }
    } else if(stoi(input) >= 48 || stoi(input) < 58) {                  // I
        s->push(input);                                                 // .
    } else {
        cout << "Please input only an operator (+, -, *, /) or an integer!"
        return 1;
    }
    return 0;
}

void serve_level(Stack<string>* s, string answer, string allowable_numbers)
    string calculated_answer = "0";
    string player_in = "";

//  cout << answer << endl;
//  cout << "Available numbers: " << allowable_numbers << endl;
    cout << "Push integers and operands onto the list to create the solution
    while(calculated_answer != answer) {
        cin >> player_in;
        calculate(s, player_in);
        calculated_answer = s->top();
        cout << "Current stack: Top --> ";
        s->print();
//      cout << "Answer: " << answer << endl;
//      cout << "Top calculated answer: " << calculated_answer << endl;

    }
    cout << "Nice! Answer was " << answer << endl;
}

void initialize_levels(string levels[5]) {
    levels[0] = "12";
    levels[1] = "3";
    levels[2] = "100";
    levels[3] = "56";
    levels[4] = "0";
}

void initialize_available_numbers(string available_numbers[5]) {
    available_numbers[0] = "6214";
    available_numbers[1] = "5643";
    available_numbers[2] = "253";
    available_numbers[3] = "24654";
    available_numbers[4] = "0";
}

```

```
int main() {
    string input, enter;
    string levels[5];
    string available_numbers[5];

    initialize_levels(levels);
    initialize_available_numbers(available_numbers);

    cout << "------------ The Stack Game --------------" << endl;
    cout << "Welcome to the stack game! Use a stack and Reverse Polish Notat
    cout << "calculations to come up with the solution given in each level!"
    cout << "[Press any key and ENTER]" << endl;
    cin >> enter;

    for (int i = 0; i < 5; i++) {
        Stack<string> s(10);
        cout << "----------- Level " << i << "-------------" << endl;
        cout << "Produce this solution: " << levels[i] << endl;;
        serve_level(&s, levels[i], available_numbers[i]);
    }
}
```