Report

代码及报告可见：C-project/project2 at main · Cheese-Bar/C-project (github.com)

一、思路

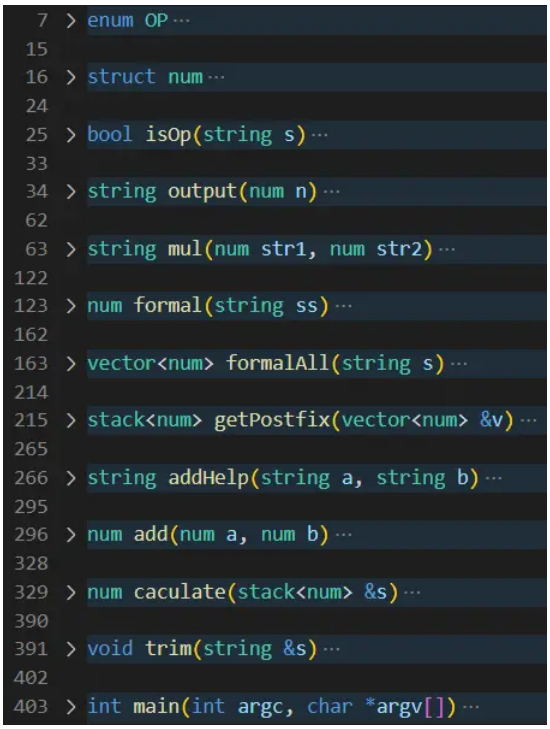
1. 沿用并拓展了project1中定义的num结构体，使其支持高精度表示以及计算符号表示。
2. 对输入的字符串进行处理，用结构体num对数字以及符号进行表示。
3. 利用`stack`将输入的中序表达式处理为后序表达式，以支持复杂计算。
4. 对后序表达式进行高精度运算，并将结果整合为可供输出的字符串。
5. 在过程中，对异常输入进行捕获，并输出提示。

二、Hilight

1. 代码结构清晰，功能强大
2. 支持中序表达式输入，含括号，一次输入即可得到结果
3. 人性化的交互设计&提示信息
4. 良好的代码风格**&**方法分割，**高内聚低耦合**
5. 支持**任意位数**的乘法&加法&括号

三、代码结构

1. 整体结构



2. 主方法

```
int main(int argc, char *argv[])
{
    while (true)
    {
        try
        {
            string s ="";
            cout << "Please enter what you want to calculate; type EXIT to
exit." << endl;
            getline(cin, s);
            trim(s);
            if (s.compare("EXIT") == 0)
            {
                cout << "Bye~" << endl;
            }
            vector<num> temp = formalAll(s);
            stack<num> temp2 = getPostfix(temp);
            num result = caculate(temp2);
            string re = output(result);
            cout << re << endl;
        }
        catch (const std::exception &e)
        {
            printf("The input is illegal, please check!\n");
        }
    }
    return 0;
}
```

3. 中序转后序

```
stack<num> getPostfix(vector<num> &v)
{
    stack<num> nums;
    stack<num> ops;
    for (size_t i = 0; i < v.size(); i++)
    {
        if (v[i].op == NONE)
        {
            nums.push(v[i]);
        }
        else
        {
            if (ops.empty())
            {
                ops.push(v[i]);
            }
            else
            {
                if (v[i].op > 0)
                {
                    while (!ops.empty() && (v[i].op <= ops.top().op))
                    {
                        nums.push(ops.top());
                        ops.pop();
```

```
                    }
                }
                ops.push(v[i]);
            }

            if (ops.top().op == RIGHT)
            {
                ops.pop();
                while (ops.top().op != LEFT)
                {
                    nums.push(ops.top());
                    ops.pop();
                }
                ops.pop();
            }
        }
    }

    while (!ops.empty())
    {
        nums.push(ops.top());
        ops.pop();
    }
    return nums;
}
```

四、效果展示

- 支持中序表达式计算
- 支持空格消除
- 支持高精度运算
- 支持连续计算与退出