

<소스코드> main.cpp

```
#include <iostream>
#include <stack>
#include <vector>
#include <cctype>
#include <stdexcept>
#include <sstream>
#include <cmath>

using namespace std;

//구조체 생성
struct CharIntConstruct
{
    int type;//0 == 숫자 , 1 == 연산자
    int number; //피연산자
    char symbol; //연산자
};

int symbolPrecedence(char op) //우선순위 판단
{
    if (op == '+' || op == '-') { // 덧셈 뺄셈 우선순위
        return 1;
    }
    if (op == '*' || op == '/' || op == '%') { // 곱셈 나눗셈 나머지 우선순위
        return 2;
    }
    if (op == '^') { //가장 높은 순위
        return 3;
    }
    return 0;
}

vector<CharIntConstruct> make_postfix(istream& ins) { //중위 표기법 > 후위 표기법
    vector<CharIntConstruct> postfix; //후위 변형 저장할 벡터
    stack<char> operators; //연산자 저장 스택
    string input;
    getline(ins, input);
    stringstream S(input); //입력 스트림 변환
    char C;
```

```

while (S >> C) {
    if (isdigit(C)) { //숫자 입력이었을 때
        S.putback(C);
        int num;
        S >> num;
        postfix.push_back({ 0, num, 0 }); //백터에 숫자 추가
    }
    else if (C == '(') {
        operators.push(C);
    }
    else if (C == ')') {
        while (!operators.empty() && operators.top() != '(') { //괄호 전까지 숫자 추가
            postfix.push_back({ 1, 0, operators.top() });
            operators.pop();
        }
        operators.pop(); //(제거
    }
    else { //연산자일 때
        while (!operators.empty() && symbolPrecedence(operators.top()) >=
symbolPrecedence(C)) { //우선순위
            postfix.push_back({ 1, 0, operators.top() }); // 우선순위 높은 연산자 추가
            operators.pop();
        }
        operators.push(C); // 연산자 스택에 추가
    }
}

while (!operators.empty()) { //연산자가 남았을 때
    postfix.push_back({ 1, 0, operators.top() });
    operators.pop();
}

return postfix; //후위표기 반환
}

bool evaluate_stack(stack<int>& numbers, const char symbol) { //스택 피연산자 꺼내 연산 수행
    if (numbers.size() < 2) {
        throw runtime_error("Error!\n");
    }

    int num2 = numbers.top();
    numbers.pop(); // 두 번째 피연산자

```

```

int num1 = numbers.top();
numbers.pop(); // 첫 번째 피연산자

switch (symbol) {
case '+':
    numbers.push(num1 + num2); // 덧셈
    break;
case '-':
    numbers.push(num1 - num2); // 뺄셈
    break;
case '*':
    numbers.push(num1 * num2); // 곱셈
    break;
case '/':
    if (num2 == 0) {
        throw runtime_error("Error!\n"); // 나머지 오류
    }
    numbers.push(num1 / num2); // 나눗셈
    break;
case '%':
    if (num2 == 0) {
        throw runtime_error("Error!\n"); // 나머지 오류
    }
    numbers.push(num1 % num2); // 나머지
    break;
case '^':
    numbers.push(pow(num1, num2)); // 거듭제곱
    break;
default:
    throw runtime_error("Error!\n"); // 연산자 오류
    break;
}
return true;
}

int evaluate_postfix(vector<CharIntConstruct> postfix) { // 변환된 수식 스택 이용 계산
    stack<int> num; //피연산자 저장 스택
    for (const auto& token : postfix) {
        if (token.type == 0) { //숫자
            num.push(token.number); //스택에 추가
        }
        else { //연산자
            evaluate_stack(num, token.symbol); //연산

```

```

    }
}
return num.top(); //결과 반환
}

int main(void)
{
    while (1)
    {
        try {
            string input;
            cout << "=====Wn" << endl;
            cout << "input:";
            getline(cin, input); //입력 받은 것 저장
            if (input == "EOI") { // 프로그램 종료 조건
                break;
            }

            stringstream sS(input); //스트림 변환
            vector<CharIntConstruct> postfix = make_postfix(sS); //중>후 변환
            int result = evaluate_postfix(postfix); //계산
            cout << "result : " << result << endl;
        }
        catch (const exception& e) {
            cout << e.what() << endl; /오류 메시지 출력
        }
    }

    return 0;
}

```

<주석 외 코드 부가 설명>

1. 잘못 입력한 경우에 대한 예외처리가 부족함.
2. 헤더파일로 나누어서 작성해보고 싶었으나 그러지 못하였음.

<실행결과>

****우분투에서 캡처하는 방법을 찾지 못해서 결과 화면을 별도로 첨부하지 못하였습니다. 죄송합니다.**