

暨南大学本科实验报告专用纸(附页)

算数表达式求值 (栈)

课程名称 数据结构 成绩评定
实验项目名称 算数表达式求值 (栈) 指导老师 干晓聪
实验项目编号 17 实验项目类型 设计性 实验地点 数学系机房
学生姓名 郭彦培 学号 2022101149
学院 信息科学技术学院 系 数学系 专业 信息管理与信息系统
实验时间 2024 年 6 月 13 日上午 ~ 2024 年 7 月 13 日中午

1. 实验目的

基于逆波兰式求算术表达式值

2. 实验环境

计算机: PC X64

操作系统: Windows + Ubuntu20.0LTS

编程语言: C++: GCC std20

IDE: Visual Studio Code

3. 程序原理

逆波兰式指的是不包含括号, 运算符放在两个运算对象的后面, 所有的计算按运算符出现的顺序, 严格从左向右进行的运算式。

将前缀表达式转换为逆波兰式, 使用两个栈, 按以下流程:

1. 数字: 入临时栈
2. 左括号: 入符号栈
3. 右括号: 重复弹出符号栈内容插入临时栈, 直到遇到左括号
4. 其他运算符: 重复弹出栈内符号直到栈顶运算符优先级不低于自身, 入符号栈

4. 程序代码

4.1. linnerCaculate.cpp

```
1  #include <iostream>
2  #include <vector>
3  #include <stdlib.h>
4  #include <map>
5  #include <algorithm>
6  #include <stack>
7  #include <math.h>
8
9  using namespace std;
10 using pii = pair<int,int>;
11
12 #define int long long
13 #define pb push_back
14 #define F first
15 #define S second
16 #define all(x) x.begin(),x.end()
17 #define loop(i,n) for(int i = 0; i < n; i++)
18
19 const int mod = 1e9 + 7;
20 const int INF = 1e18;
21
22 bool ifdigit(string &s){
23     if(s.size() >= 2) return 1;
24     else return (s[0] > '0' && s[0] < '9');
25 }
26
27 int str2int(string &s) {
28     int t = 0;
29     for(int i = 0;i < s.size();i++) {
30         t += pow(10LL,i) * (s[s.size()-1-i] - '0');
31     } return t;
32 }
33
34 int getPri(string s){
35     if(s[0] == '+' || s[0] == '-') return 0;
36     else return 1;
37 }
38
39 //
40 void solve()
41 {
42     string s;
43     cout << ">>>>";
44     getline(cin,s);
45     // cin.get();
46     s.push_back('\n');
47     vector<string> RES;
```

暨南大学本科实验报告专用纸(附页)

```
48     stack<string> sign,tmp1,rbc;
49     string tmp;
50     for(auto x:s) {
51         if(x == ' ' || x == '\n'){
52             if(tmp.size()) RES.push_back(tmp);
53             tmp.clear();
54         } else tmp.push_back(x);
55     }
56     for(auto x:RES){
57         if(isdigit(x)) tmp1.push(x);
58         else{
59             if(x[0] == '(') sign.push(x);
60             else if(x[0] == ')') {
61                 while(sign.top()[0] != '(' && sign.size()){
62                     tmp1.push(sign.top());
63                     sign.pop();
64                 }
65                 sign.pop();
66             } else {
67                 while(1) {
68                     if (sign.size() == 0 || sign.top()[0] == '(')
69                     { sign.push(x);break;}
70                     else {
71                         if(getPri(x) > getPri(sign.top()))
72                         { sign.push(x);break;}
73                         else {tmp1.push(sign.top());sign.pop();}
74                     }
75                 }
76             }
77             while(sign.size()) {
78                 tmp1.push(sign.top());
79                 sign.pop();
80             }
81             vector<string> gt;
82             while(tmp1.size()){
83                 gt.push_back(tmp1.top());
84                 tmp1.pop();
85             }
86             vector<string> cacu;
87             for(int i = 0;i < gt.size();i ++){cacu.push_back(gt[gt.size()-1-i]);}
88
89             stack<int> nums;
90             for(auto x:cacu) {
91                 if(isdigit(x)) nums.push(str2int(x));
92                 else {
93                     int a = nums.top();
94                     nums.pop();
95                     int b = nums.top();
```

```
96         nums.pop();
97         char c = x[0];
98         switch (c)
99         {
100             case '+':
101                 nums.push(a+b);
102                 break;
103             case '-':
104                 nums.push(a-b);
105                 break;
106             case '*':
107                 nums.push(a*b);
108                 break;
109             case 'x':
110                 nums.push(a*b);
111                 break;
112             case '/':
113                 nums.push(a/b);
114                 break;
115
116             default:
117                 break;
118         }
119     }
120 }
121
122 cout << nums.top() << "\n";
123
124 }
125
126 signed main()
127 {
128     // std::ios::sync_with_stdio(false);
129     // std::cin.tie(nullptr);
130     // std::cout.tie(nullptr);
131
132     // int T = 1;
133     // cin >> T;
134     while(1)
135         solve();
136     return 0;
137 }
```

5. 测试数据与运行结果

运行上述 `_PRIV_TEST.cpp` 测试代码中的正确性测试模块, 得到以下内容:

```
>>>>1 + 1
```

暨南大学本科实验报告专用纸(附页)

```
2
>>>>2 * 3 + 1
7
>>>>2 * ( 3 + 1 )
8
>>>>115 * 514
59110
>>>>( 1 + 1 ) / 2 * 2 + 2
4
>>>>( 1 + 1 ) / 2 * ( 2 + 2 + 2 )
4
>>>>( 1 + 1 ) / 2 * ( 2 + 2 + 2 )
6
```

可以看出，代码运行结果与预期相符，可以认为代码正确性无误。