**新疆政法学院**

**数据结构与算法**

**实验报告**

班 级： 计算机科学与技术4班

学 号： 2124030160

姓 名： 孙久猛

指导老师： 张家琦

学 期： 2022-2023第一学期

|  |  |  |  |
| --- | --- | --- | --- |
| **实验题目** | 实验三：求前缀表达式的值 | | |
| **实验时间** | **10:00** | **实验地点** | **东训0114** |
| **实验成绩** |  | **实验性质** | **□验证性 √设计性 □综合性** |
| 教师评价：  □算法/实验过程正确； □源程序/实验内容提交 □程序结构/实验步骤合理；  □实验结果正确； □语法、语义正确； □报告规范；  其他：  评价教师签名： | | | |
| 一、实验目的  1.理解前缀表达式、中缀表达式、后缀表达式的区别；  2.掌握前缀表达式、中缀表达式、后缀表达式之间的转换；  3.理解运用栈模型计算前缀表达式的过程；  4.通过对前缀表达式求值的练习，加深对栈、队列等相关线性结构的理解。 | | | |
| **二、实验项目内容（实验题目）**  （一）求前缀表达式的值  1、使用栈模型，对前缀表达式求值算法进行实现；  2、输入正确的前缀表达式；  3、输出该表达式的计算结果。  **三、源程序（实验步骤/实验过程/算法）**  //Deom \_03.c  #include <stdio.h>  #include <stdlib.h> //库函数工具包  #include <string.h>  #include <ctype.h>  /\*变量的值必须是EOF或可用unsigned char类型表示的字符，函数的返回值为int类型\*/  #define N 30  #define TRUE 1  #define FALSE 0  typedef struct {  double data[N];  int top;  } Stack;  void Push( Stack \*ptrs, double item )  {//定义Push（）函数判断栈满继续不满执行指向top  if( ptrs->top == N - 1 ){  printf( "Stack is full.\n" );  return;  } else {  ptrs->data[ ++( ptrs->top ) ] = item; //顶端数据赋值  return;  }  }  double Pop( Stack \* ptrs )  {//定义函数Pop（）判断栈的堆积  if( ptrs->top == -1 ) { //top()取栈顶元素  printf( "Stack is empty.\n" );  return 0; //栈为空时的情况  } else  return ptrs->data[ ( ptrs->top )-- ]; //栈的指针由上到下  }  int IsOperator( char ch )  {//定义函数IsOperator();判断输入的字符有无错误  if( ch == '+' || ch == '-' || ch == '\*' || ch == '/' ) //判断类型“+ —\* / ”  return TRUE;  else  return FALSE;  }  double Calculate( double a, double b,char ch )  {//Calculate()函数用于计算表达式  switch( ch ) {  case '+' : return a + b; break;  case '-' : return a - b; break;  case '\*' : return a \* b; break;  case '/' : return a / b;  }  }  int main()  {  char expr[N]; //定义字符类型的数组  printf("输入需要计算的前缀表达式（注：每个操作数、操作符需空格隔开）：\n");  gets( expr ); //库函数=scanf（）;  //scanf("%[^\n]", expr);  //scanf("%s", expr);  int len = strlen( expr ); //调用strlen()函数判断字符长度  Stack ss; //定义栈ss  ss.top = -1; //初始化ss的top值  double cc = 1;  double tem\_sum = 0;  double operand\_a;  double operand\_b;  double result;  int error = 0; // 记录除数为0的错误情况  int i;  for( i = len - 1; i >= 0; -- i ) {  if( expr[i] >= '0' && expr[i] <= '9' )  {//数组data的值在0~9之间  tem\_sum += ( expr[i] - '0' ) \* cc;  cc \*= 10;  if( expr[i-1] == '+' ) {  //printf( "%d\n", tem\_sum );  Push( &ss, tem\_sum );//Push()函数  tem\_sum = 0;  cc = 1;  i -= 2; // 跳过下一个正号和空格  continue;  } else if( expr[i-1] == '-' ) {  tem\_sum = -tem\_sum;  //printf( "%d\n", tem\_sum );  Push( &ss, tem\_sum );  tem\_sum = 0;  i -= 2; // 跳过下一个负号和空格  continue;  } else if( expr[i-1] == ' ' ) { // 一个数字处理完了  //printf( "%d\n", tem\_sum );  Push( &ss, tem\_sum );  tem\_sum = 0;  cc = 1;  i --;  continue;  }  } else if( expr[i] == '.' ) {  tem\_sum /= cc \* 1.0;  cc = 1;  } else if( IsOperator( expr[i] ) ) {  operand\_a = Pop( &ss );  operand\_b = Pop( &ss );  if( expr[i] == '/' && operand\_b == 0 ) {  error = 1;  break;  } else {  result = Calculate( operand\_a, operand\_b, expr[i] );  //printf( "result:%.1lf\n", result );  Push( &ss, result );  i--; // 跳过下一个空格  }  }  }  if( error != 1 )  printf( "计算结果为: %.1lf\n", Pop( &ss ) );  else  printf( "ERROR\n" );  return 0;  }  **四、运行结果**  **//Deom\_03.c**  **Running statuation:**    **//Picture\_01.exe**    **//Picture\_02.exe**    **//Picture\_03.exe**    **//Picture\_04.exe** | | | |

日 期： 2022年10月12日