数据结构实验四 堆栈的应用

[ 发布人： [易法令](https://www.scholat.com/flyi)   截止时间：2021-10-14 23:59:59  ]

一、实验目的:

1．掌握堆栈的存储方式和基本操作

2．掌握堆栈后进先出运算原则在解决实际问题中的应用

二、实验内容:

1.利用栈结构，编写程序将十进制数转换成二进制数或八进制数。

说明:十进制数值转换成二进制使用辗转相除法将一个十进制数值转换成二进制数值。即用该十进制数值除以2，并保留其余数；重复此操作，直到该十进制数值为0为止。最后将所有的余数反向输出就是所对应的二进制数值。十进制数值转换成八进制算法类似。转换算法要求用一个函数完成。

2.假设算术表达式中允许包含两种括号:圆括号和方括号，其嵌套的顺序随意，即（[][]）

或[（[]()）]等为正确格式，而[(]或()))或 [())均为不正确的格式。请使用栈结构，写一算法检验某表达式中的括号是否匹配，并测试你的算法是否正确。测试表达式为:

(1)[(1+2)\*3-1]+[((1+2]\*3)-1]

(2) [(1+2)\*3-1]+[(1+2)\*3-1]

三、实验源代码

1.改2进制

#include <malloc.h>

#include <stdio.h>

#include <stdlib.h>

typedef int ElemType;

#define MaxStackSize 100

#include "LinkedStack.h"

int main(void){

    int decimal;

    SequenceStack myStack;

    StackInit(&myStack);

    scanf("%d",&decimal);

    while (decimal) {

        StackPush(&myStack,decimal%2);

        decimal=decimal /2;

    }

    while (StackNotEmpty(myStack)) {

    StackPop(&myStack, &decimal);

    printf("%d",decimal);

    }

}

#ifndef LinkedStack\_H

#define LinkedStack\_H

#include <stdio.h>

#include <stdlib.h>

//typedef int ElemType;

//#define MaxStackSize 100//////////////////////////////////////不能加;

typedef struct{

    ElemType stack[MaxStackSize];

    int top;

} SequenceStack;

void StackInit(SequenceStack \*S){

    S->top=0;

}

int StackNotEmpty(SequenceStack S){

    if(S.top<=0)return 0;

    else

     return 1;

}

int StackPush(SequenceStack \*S,ElemType x){

    if(S->top>=MaxStackSize){

        printf("this stack is overflow!");

        return 0;

    }

    else {

    S->stack[S->top]=x;

    S->top++;

    return 1;

    }

}

int StackPop(SequenceStack \*S,ElemType \*d){

    if(S->top<=0){

        printf("this stack is empty!\n");

        return 0;

    }

    else {

    S->top--;

    \*d=S->stack[S->top];

    return 1;

    }

}

int StackTop(SequenceStack S,ElemType \*d){

    if (S.top<=0) {

    printf("this stack is empty!\n");

    return 0;

    }

    else {

    \*d=S.stack[S.top-1];

    return 1;

    }

}

#endif

 2.确认括号齐

#include <stdio.h>

#include <string.h>

#define Mise 20

typedef char ElemType;

#include "paren.h"

int main(void){

    char line[20];

    int i=0,flag=0;

    SequencePar myList;

    SeqInit(&myList);

    scanf("%s",line);

    while (line[i++]!='\0') {

      //////////////////////////////////////////////////////////////////  printf("%d\n",i);

    if(    line[i-1]=='{'  ||  line[i-1]=='['  ||  line[i-1]=='('  ){

        SeqPush(&myList,line[i-1]);

            }

    if (   line[i-1]==']'  ||  line[i-1]=='}'  ||  line[i-1]==')' ) {

        SeqPull(&myList,line[i-1]);

        }

    }

    return 0;

}

#ifndef paren\_H

#define paren\_H

#include <stdio.h>

 int k=0;

typedef struct {

    ElemType pare[Mise];

    int top;

} SequencePar;

void SeqInit(SequencePar \*S){

    S->top=0;

}

int SeqNot(SequencePar S){

    if(S.top<=0)return 0;

    else

     return 1;;

}

int SeqPush(SequencePar \*S,ElemType x){

    if (S->top>=Mise) {

    printf("The stack is overflew!\n");return 0;

    }

    else {

    if (    x=='{'  ||  x=='['  ||  x=='('  ) {

    S->top++;

    S->pare[S->top]=x;

  //////////////////////////////////////////////////////////////////  printf("in:%c\n",S->pare[S->top]);

    return 1;

    }

    else {}

    }

}

int SeqPull(SequencePar \*S,ElemType d){

    if(S->top<=0){

        printf("this stack is empty!\n");

        return 0;

    }

    else {

         if(S->pare[S->top]=='{'&&d=='}'){

    ///////////////////////////////////////////     printf("on:%c,line[i]:%c",S->pare[S->top],d);

          S->top--;

        return 0;

    }

    else if(S->pare[S->top]=='['&&d==']'){

   ////////////////////////////////////////////      printf("on:%c,line[i]:%c",S->pare,d);

          S->top--;

        return 0;

    }

    else if(S->pare[S->top]=='('&&d==')'){

  //////////////////////////////////////////       printf("on:%c,line[i]:%c",S->pare[S->top],d);

          S->top--;

        return 0;

    }

    else {

  printf("eorroer%d\n",k);

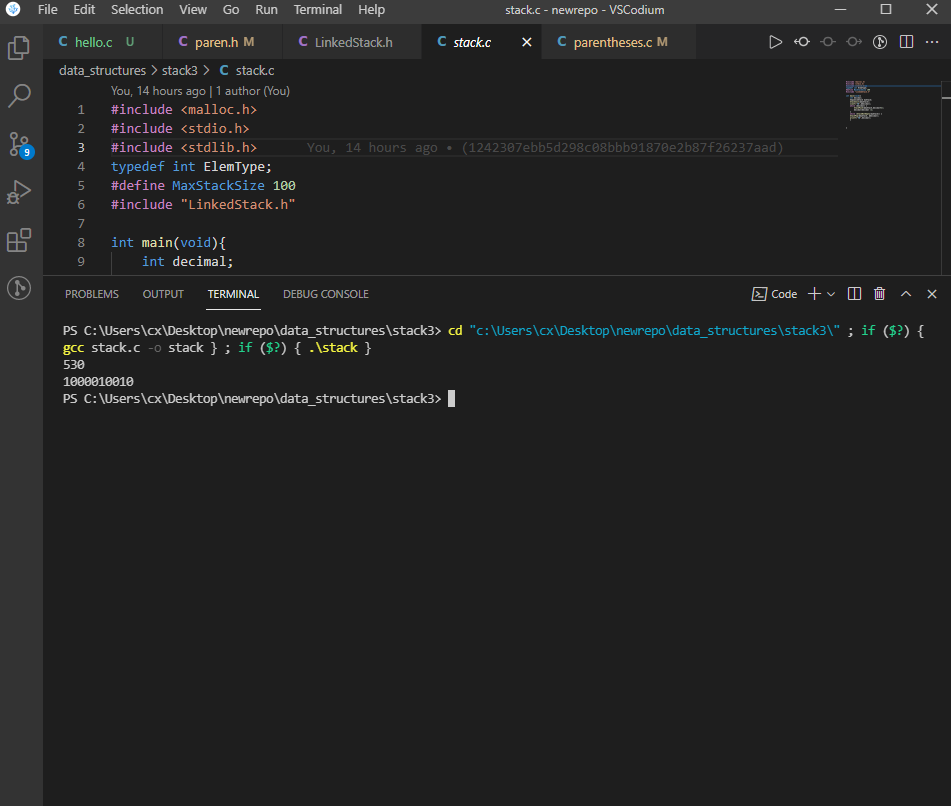
  //////////////////////////////////////////  printf("on:%c,line[i]:%c",S->pare[S->top],d);

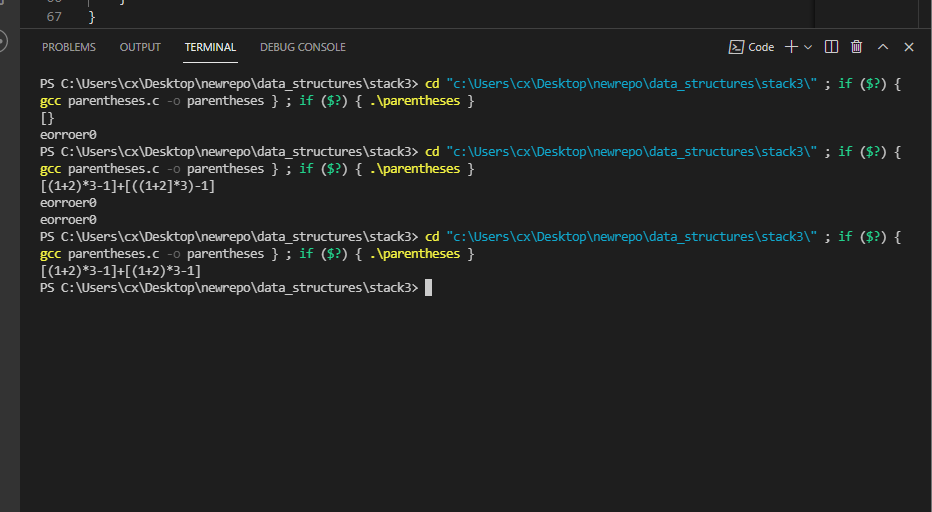
    }

    }

}

#endif

四、实验结果



五、实验心得

1.一定要分清楚出栈和取栈顶元素

2typedef struct {

    ElemType pare[Mise];/////////////////这一定要是数组！！！！！！不然就不是栈了

    int top;

} SequencePar;