#include<iostream.h>

#include<conio.h>

#include<stdlib.h>

#include <iomanip.h>

#include <string>

//using namespace std;

//------------------------------栈的顺序存储结构---------------------------------------------

typedef char ElemType; // 数据元素的类型

const int MAXSIZE=7; // 数组的容量

class SqStack

{ private:

ElemType elem[MAXSIZE];

int top;

public:

SqStack( void);

~SqStack(){};

int SqStack::SetEmpty();

void SqStack::push( ElemType e);

ElemType SqStack::pop();

void SqStack::PrintOut();

int SqStack::IsEmpty(void)const ;

};

//-------------------------------------------------------------

SqStack::SqStack( void):top(0){ }

int SqStack::SetEmpty()

{ return top==0;

}

void SqStack::push( ElemType e)

{

if(top==MAXSIZE){

cout<<"栈满，无法入栈"<<endl;

}else {

elem[top++]=e;

}

}

ElemType SqStack::pop()

{ ElemType e;

if(!IsEmpty()){

e=elem[--top];

}else{

cout<<"栈为空，无法出栈"<<endl;

exit(0);

}

return e ;

}

void SqStack::PrintOut()

{int k;

cout<<"\n PrintOut Data:\n" ;

for(k=top-1; k>=0;k--) cout<<setw(6)<<elem[k];

cout<<endl;

}

int SqStack::IsEmpty(void)const

{ if(top==0) return 1;

else return 0;

}

//1. 从键盘上输入一算术表达式检验该算术表达式中括号是否配对。

//① 编写一程序检验算术表达式中括号是否配对；

//② 从键盘输入一算术表达式，调用你所编写的程序，检验该表达式中括号是否配对，配对输出"yes",否则输出"no"。

//提示：括号配对检查的原则是，对表达式从左向右扫描。当遇到左括号时，左括号入栈；而遇到右括号时，首先将栈中的栈顶元素弹出栈，再比较弹出元素是否与右括号匹配，如果两者匹配，则操作继续；否则，检查出错，打印"no"，并停止操作。当表达式全部扫描完后，如果栈为空，说明括号作用层次嵌套正确，打印"yes"，并停止操作。

//方法是先创建一个存放左括号的空栈，并将'＃'作为栈底元素；顺序扫描表达式，当遇非括号时，继续扫描，遇'（'、'['、'{'时，进栈；而遇'}'、']'、'）'时，与栈顶元

//---------------------------------------------------------------------------

int main(int argc, char\* argv[])

{ int k;

ElemType e,x;

SqStack as;

int left=0;//统计左边的括号

int right=0; //统计右边的括号

cout<<"\n 栈的顺序存储结构实现";

do{

cout<<"\n\n";

cout<<"\n\n 1.输入算书表达式";

cout<<"\n\n 2.配对";

cout<<"\n\n 3..结束程序";

cout<<"\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* ";

cout<<"\n 请输入你的选择(1,2,3,4)"; cin>>k;

switch(k){

case 1:{cout<<"\n 请输入算数表达式[5个字]： ";

for(int i =0;i<MAXSIZE;i++){

cin>>e;

as.push(e);

}

as.PrintOut();

}break;

case 2:{ cout<<"\n 配对";

for(int i=0;i<MAXSIZE;i++){

x=as.pop();

if (x=='('){left++;}

if (x==')'){right++;}

}

//左边的括号等于右边的括号，则匹配成功

if (left==right){cout<<"配对成功，是算术表达式"<<endl;}else{cout<<"配对不成功，不是算术表达式"<<endl;}

}break;

default:break;

} //switch

cout<<"\n--------------------------------- ";

}while(k>=1&&k<3);

cout<<"\n 再见!";

cout<<"\n 按任意键，返回。";

\_getch(); return 0;

}

//-----------------------------------------------



