实验四：栈和队列子系统

1.实验目的

1. 掌握栈和队列的特点及其描述方法；
2. 掌握用链式存储结构和顺序存储结构实现一个栈/队列；
3. 掌握栈的各种等基本操作及几个典型应用的算法。
4. 掌握队列的建立、出队、入队等基本操作和算法实现；
5. 递归与断点调试。

## 2.实验内容及要求

**说明：本次实验学生可以自由选择一个子系统实现即可。编程基础好的同学可以两个都做。**

1. 分段读懂程序，敲入代码，逐句注释清楚。
2. 调试程序，记录实验问题及解决办法，写实验报告。

**A.栈子系统内容：**

1. 设计一个字符型的栈；
2. 编写进栈、出栈、显示栈中全部元素的程序；
3. 设计一个选择式菜单。
4. 编写一个把十进制整数转换成二进制数的应用程序（**选做**）；
5. 编写一个把中缀表达式转换成后缀表达式（逆波兰式）的应用程序（**选做**）；

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* 1---------进 栈 \*

\* 2---------出 栈 \*

\* 3---------显 示 \*

\* 4---------数制转换 \*

\* 5---------逆波兰式 \*

\* 0---------返 回 \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**B.队列子系统内容：**

1. 设计一个字符型的链队列；
2. 编写队列的进队、出队、读队头元素、显示队列元素的程序；
3. 设计一个选择式菜单。

## 3.参考程序及实验结果

### A.栈子系统（需自行调试加注释）

#include <stdio.h>

#include <stdlib.h>

#define STACKMAX 100

typedef struct stacknode

{

int data;

struct stacknode \*next;

}StackNode;

typedef struct

{

StackNode \*top;

}LinkStack;

void Push (LinkStack &s,int x)

{

StackNode \*p=new StackNode;

p->data=x;

p->next=s.top;

s.top=p;

}

int Pop(LinkStack &s,int &x)

{

StackNode \*p;

if(s.top!=NULL)

{

p=s.top;

x=p->data;

s.top=p->next;

delete p;

return 1;

}

else

return 0;

}

void ShowStack (LinkStack s)

{

StackNode \*p=s.top;

if (p==NULL)

printf("\n\t\t栈为空。 ");

else

{

printf("\n\t\t栈元素为： ");

while (p!=NULL)

{

printf("%6d",p->data);

p=p->next;

}

printf("\n");

}

}

void Conversion(int n)

{

LinkStack s;

int x;

s.top=NULL;

do

{

x=n%2;

n=n/2;

Push(s,x);

}while(n);

printf("\n\t\t转换后的二进制数值： ");

while (Pop(s,x))

printf("%d",x);

printf("\n");

}

void Suffix()

{

char str[STACKMAX];

char stack[STACKMAX];

char exp[STACKMAX];

char ch;

int sum,i,j,t,top=0;

printf("\n\t\t输入算术表达式(算术符只能包含+,-,\*,/),以#结束:\n\t\t");

fflush(stdin);//清空缓存

i=0;

do

{

i++;

scanf("%c",&str[i]);

}while(str[i]!='#'&& i!=STACKMAX);

sum=i;

t=1;

i=1;

ch=str[i];

i++;

while (ch!='#')

{

switch (ch)

{

case '(':

top++;

stack[top]=ch;

break;

case ')':

while (stack[top]!='(')

{

exp[t++]=stack[top--];

exp[t++]=',';

}

top--;

break;

case '+':

case '-':

while (top!=0 && stack[top]!='(')

{

exp[t++]=stack[top--];

exp[t++]=',';

}

stack[++top]=ch;

break;

case '\*':

case '/':

while(stack[top]=='\*' || stack[top]=='/')

{

exp[t++]=stack[top--];

exp[t++]=',';

}

stack[++top]=ch;

break;

case ' ':

break;

default:

while (ch>='0'&&ch<='z')

{

exp[t++]=ch;

ch=str[i++];

}

i--;

exp[t++]=',';

}

ch=str[i++];

}

while (top!=0)

{

exp[t++]=stack[top--];

if(top!=0)

exp[t++]=',';

}

printf("\n\t\t输入的中缀表达式是: ");

for(j=1;j<sum;j++)

printf("%c",str[j]);

printf("\n\n\t\t后缀表达式是: ");

for(j=1;j<t;j++)

printf("%c",exp[j]);

printf("\n");

}

void main()

{

LinkStack s;

int i=1,j=1,val,n;

char choice;

s.top=NULL;

while(1)

{

printf("\n");

printf("\n\t\t 栈子系统 ");

printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t\* 1--进 栈 \*");

printf("\n\t\t\* 2--出 栈 \*");

printf("\n\t\t\* 3--显 示 \*");

printf("\n\t\t\* 4--数制转换 \*");

printf("\n\t\t\* 5--逆波兰式 \*");

printf("\n\t\t\* 0--退出程序 \*");

printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");

printf("\n\t\t 请选择菜单号(0--5): ");

fflush(stdin);

choice=getchar();

switch (choice)

{

case '1':

while (1)

{

printf("\n\t\t输入一个整数('0'表示结束)并按回车: ");

scanf("%d",&val);

if (val!=0)

Push (s,val);

else

break;

}

break;

case '2':

if (Pop(s,val))

printf("\n\t\t出栈元素为: %6d",val);

else

printf("\n\t\t栈为空，没有元素可以出栈! \n");

break;

case '3':

ShowStack(s);

break;

case '4':

printf("\n\t\t请输入一个十进制正整数: ");

scanf("%d",&n);

Conversion(n);

break;

case '5':

Suffix();

break;

case '0':

exit(0);

default:

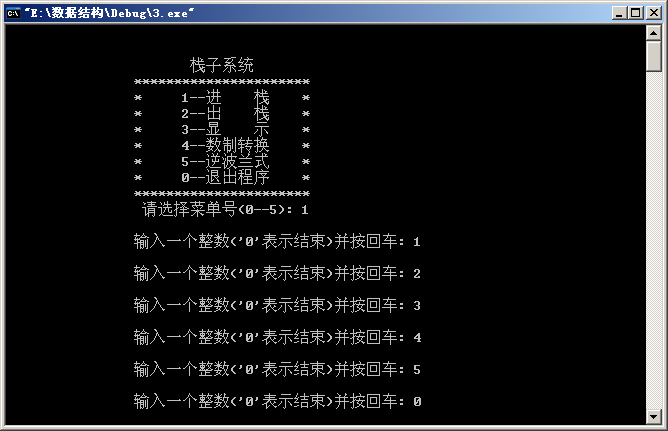
printf("\n\t\t输入的菜单错误，请重新输入! \n");

}

}

}

4. 运行结果



### B.队列子系统（（需自行调试加注释））

#include<stdio.h>  
#include<malloc.h>  
//链队列结点定义  
typedef struct queuenode  
{   
char data;  
struct queuenode \*next;  
}queuenode;  
  
//链队列结构  
typedef struct  
{  
queuenode \*front,\*rear; //头、尾指针  
}linkqueue;  
  
  
void InQueue(linkqueue \*q) // 入队函数   
{ char x;   
queuenode \*p;  
//申请一个新结点  
   p=(queuenode \*)malloc(sizeof(queuenode));  
printf("\n\t\t请键入一个进队的元素：");  
scanf("%c",&x);  
getchar();  
p->data=x;  
p->next=NULL;  
  
//新结点入队  
if(q->front==NULL) //新结点进入一个空队列  
q->front=p;  
else  
q->rear->next=p;  
q->rear=p; //入队后尾指针后移  
if(p)  
printf("\n\t\t%c入队成功！",x);  
}  
  
  
int OutQueue(linkqueue \*q,int \*v) // 出队函数  
{ queuenode \*p;  
if(q->front==NULL) //空队列不能做出队操作，下溢错误  
return 0;  
else  
{ p=q->front;  
\*v=p->data;  
//出队后如果队列为空，需要重设头、尾指针取值  
q->front=p->next;  
if(q->front==NULL)  
q->rear=NULL;  
    free(p);  
return 1;  
}  
}  
  
  
void ShowQueue(linkqueue \*q) // 显示队列函数  
{ queuenode \*p=q->front;  
if(p==NULL)  
printf("\t\t队列为空!\n");  
else  
  { printf("\t\t队列为：");  
while(p!=NULL)  
{ printf("%6c",p->data);  
p=p->next;  
}  
printf("\n");  
}  
}  
  
  
void ReadFront(linkqueue \*q) // 读队首元素函数  
{ if(q==NULL||q->front==NULL)  
printf("\t\t队列为空!没有队顶元素!\n");  
else  
printf("\n\t\t队首元素是：%4c\n",q->front->data);  
}  
  
void main()                               // 队列子系统主函数  
{   
int val,i=1;  
    char w,choice;  
    queuenode \*ptr,\*t;  
    linkqueue \*q;  
       q=(linkqueue \*)malloc(sizeof(linkqueue));  
q->front=q->rear=NULL;  
while (i)  
{ printf("\n\n\n\n");  
printf("\n\t\t\t\t  队 列 子 系 统\n");  
printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
printf("\n\t\t\*           1---------------进          队          \*");  
printf("\n\t\t\*           2---------------出          队          \*");  
printf("\n\t\t\*           3---------------队  顶  元  素          \*");  
printf("\n\t\t\*           4---------------显          示          \*");  
printf("\n\t\t\*           0---------------返          回          \*");  
printf("\n\t\t\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*");  
printf("\n\t\t请选择菜单号(0--4):");  
scanf("%c",&choice);  
getchar();  
switch(choice)  
{  
case '1':  
InQueue(q);break;  
case '2':  
if(OutQueue(q,&val)==0)  
printf("\n\t\t队列为空!\n");  
else  
printf("\n\t\t出队元素为：%c",val); break;  
case '3':  
ReadFront(q); break;  
case '4':  
ShowQueue(q); break;  
case '0':  
i=0; break;  
default:;  
}  
if(choice=='1'||choice=='2'||choice=='3'||choice=='4')  
{ printf("\n\t\t按回车键继续.\n");  
w=getchar();  
if(w!='\xA')  
{getchar();  
i=0;  
}  
}  
}  
ptr=q->front;  
     while(ptr!=NULL)  
 {   
t=ptr->next;  
free(ptr);  
ptr=t;  
 }  
  q->front=q->rear=NULL;  
}