# 实验四 栈和队列

实验预备知识：

1．熟练运用指针进行程序设计，掌握结构体指针。

2．掌握使用结构体指针访问结构体变量。

3．掌握指针作为函数的参数使用。

4．理解栈和队列的含义、目的和处理方法。

## 一、实验目的

1. 理解和掌握栈和队列的类型定义方法。
2. 掌握栈和队列中的基本运算，包括创建、判空及判满、出队（栈）/入队（栈）等基本操作。
3. 学习利用栈或队列解决实际问题

## 二、实验要求（本次实验要求上交）

**【题目1】----验证型**

实现链栈各种基本运算的算法。参考代码见（文件夹4-1）

该程序主函数完成如下功能：

1. 初始化链栈
2. 判断链栈是否为空
3. 元素依次入栈
4. 输出该链栈的长度
5. 输出从栈顶到栈底的元素
6. 销毁该链栈

该工程的程序结构图如下图所示：

algo3-2.cpp

exp3-2.cpp

InitStack

Pop

DestroyStack

StackEmpty

Push

main

GetPop

实现循环队各种基本运算的算法。参考代码见（文件夹4-2）

该程序主函数完成如下功能：

1. 初始化队列
2. 判断循环队是否为空
3. 元素依次入队
4. 队列元素出队
5. 输出出队序列
6. 销毁该循环队

该工程的程序结构图如下图所示：

algo3-3.cpp

exp3-3.cpp

InitQuequ

enQuequ

DestroyQuequ

QuequEmpty

deQuequ

main

**【题目2】**

**编写一个程序，利用栈的特性实现二进制转换为八进制**。**(给出了二进制转换为十进制的参考算法供大家参考，见bin\_dec.c)**

|  |
| --- |
| 算法分析： |
| 转换时的算法是:每三个从栈中取出字符，然后将其转换成10进制数，放入另一个栈中，当还没到栈底的时候继续取出，在取出前判断栈中剩余的字符是不是小于三个(用栈的长度减去已取出的数目)来判断，如果小于就不再取三个字符，把剩余的字符取出来转换为十进制数即可，再将其放入另一个栈中，最后再将另一个栈中的十进制数依次从栈顶取出来，再依次显示，显示出来的数即为相应二进制数转换为的十进制数。 |

|  |
| --- |
| 请将源程序附录于此： |
| #include <stdio.h>  #include <stdlib.h>  #include <math.h>  #define STACK\_INIT\_SIZE 20  #define STACKINCREMENT 10  typedef char ElemType;  typedef struct  {  ElemType \*base; //栈底  ElemType \*top; //栈顶  int stackSize;  }sqStack;  void InitStack(sqStack \*s)  {  s->base = (ElemType \*)malloc(STACK\_INIT\_SIZE \* sizeof(ElemType));  if( !s->base )  {  exit(0);  }  s->top = s->base;  s->stackSize = STACK\_INIT\_SIZE;  }  void Push(sqStack \*s, ElemType e)  {  if( s->top - s->base >= s->stackSize )  {  s->base = (ElemType \*)realloc(s->base, (s->stackSize + STACKINCREMENT) \* sizeof(ElemType));  if( !s->base )  {  exit(0);  }  }  \*(s->top) = e;  s->top++;  }  void Pop(sqStack \*s, ElemType \*e)  {  if( s->top == s->base )  {  return;  }  \*e = \*--(s->top);  }  int StackLen(sqStack s)  {  return (s.top - s.base);  }  int main()  {  ElemType c;  sqStack s,q;  int len, i,j,n=3;  int sum=0;  InitStack(&s);  InitStack(&q);  printf("请输入二进制数，输入#符号表示结束！\n");  scanf("%c", &c);  while( c != '#' )  {  Push(&s, c);  scanf("%c", &c);//如果前一个scanf输入的是字符串,那么这里的scanf就不用再输入了，从前面的缓存里面取还没有读入的字符  }  getchar(); // 把'\n'从缓冲区去掉  len = StackLen(s);  printf("栈的当前容量是: %d\n", len);  for( i=0; i < len; i+=n )  {  if(len-i==1){  j=0;  for(j=0;j<1;j++){  Pop(&s, &c);  sum = sum + (c-48) \* pow(2, j);  };  Push(&q, sum);  break;  }else if(len-i==2){  j=0;  for(j=0;j<2;j++){  Pop(&s, &c);  sum = sum + (c-48) \* pow(2, j);  };  Push(&q, sum);  break;  }  j=0;  for(j=0;j<3;j++){  Pop(&s, &c);  sum = sum + (c-48) \* pow(2, j);  };  Push(&q, sum);  sum=0;  }  len = StackLen(q);  printf("二进制转化为八进制数是:");  for( i=0; i < len; i++ ){  Pop(&q, &c);  printf("%d", c);  }  printf("\n", c);  return 0;  } |
| 请将运行结果图附录于此 |
| C:\Users\Administrator\AppData\Roaming\Tencent\Users\1061892125\QQ\WinTemp\RichOle\[DJQGUP]Y2Z$SIZ5_0VADO2.png  C:\Users\Administrator\AppData\Roaming\Tencent\Users\1061892125\QQ\WinTemp\RichOle\`AZL6Y835~CXSPCKQ6_D`WP.png |

## 实验上传方法：

将源程序文件和本word文档（添加了相应的源程序和截图）上传。多个文件请将其放入一个文件夹压缩后上传。请于指定时间前上交，过时不补！