```
#include "stdafx.h"
#include "string.h"
#include "malloc.h"
#define MAXSIZE 100

/*回文*/
typedef struct node
{
   char data;
   struct node *next;
}QNode,*QNodePtr,SNode,*SLink;
typedef struct{
   QNodePtr front;
   QNodePtr rear;
}QLink;

void InitStack(SLink *sl)
{
   *sl = NULL;
}
void Push(SLink *sl,char e)
{
   SLink p = (SLink)malloc(sizeof(SNode));
   p->data = e;
   p->next = *sl;
   *sl = p;
}
```

```
int Pop(SLink *sl,char *e)
{
  if(*sl == NULL)
    return 0;
  *e = (*sl)->data;
  SLink p = *sl;
  *sl=p->next;
  free(p);
}
int isEmptyStack(SLink s)
{
  return s==NULL;
}

int InitQueue(QLink *ql)
{
  QNodePtr p = (QNodePtr)malloc(sizeof(QNode));
  p->next = NULL;
  ql->front = ql->rear = p;
  return 1;
}

int InQueue(QLink *ql, char e)
{
  QNodePtr p = (QNodePtr)malloc(sizeof(QNode));
  p->data = e;
  p->next = NULL;
```

```
    ql->rear->next = p;
    ql->rear = p;
    return 1;
}
int OutQueue(QLink *ql,char *e)
{
    if(ql->front == ql->rear)
        return 0;
    QNodePtr q = ql->front->next;
    *e = q->data;
    ql->front->next = q->next;
    if(q==ql->rear)
        ql->rear = ql->front;
    free(q);
}

int Judge(char str[])
{
    QLink q;
    SLink s=NULL;
    InitQueue(&q);
    InitStack(&s);
    char *p = str;
    char ch= *p;
    while(ch!='\0')
    {
        Push(&s,ch);
```

```
        InQueue(&q,ch);
        p++;
        ch = *p;
    }
    while(!isEmptyStack(s))
    {
        char ch1,ch2;
        Pop(&s,&ch1);
        OutQueue(&q,&ch2);
        if(ch1!=ch2)
            break;
    }
    if(isEmptyStack(s))
        return 1;
    else
        return 0;
}
int _tmain(int argc, _TCHAR* argv[])
{
    char str[20];
    printf("输入字符串:\n");
    gets(str);
    int i = Judge(str);
    if(i==1)
        printf("是回文！\n");
    else
        printf("不是回文！\n");
```

```
    scanf("%d",&i);
    return 0;
}
```