**任务4 栈和队列**

# 任务04\_1 删除栈中重复元素

请根据要求，按照**C语言语法，**编写栈的四个函数：InitStack、JudgRepetition、Push、PrintStack，四个函数可以互相调用，完成规定的运行效果。

**预定义常量和类型**

#define Status int

#define OK 1

#define ERROR 0

#define STACK\_INIT\_SIZE 20

#define STACKINCREMENT 5

#define ElemType int

**顺序栈结构**

typedef struct {

ElemType \*base;

ElemType \*top;

int stacksize;

}SqStack;

**四个函数的函数原型及功能如下：**

Status InitStack(SqStack \*S) ：栈的初始化;

Status JudgRepetition(SqStack \*S,ElemType e)：将e与栈中元素比较判断是否有重复元素，有的话返回ERROR，没有的话返回OK;

Status Push(SqStack \*S,ElemType e):完成不重复的元素入栈操作，若e与栈中元素重复，不予入栈；

void PrintStack(SqStack S)：打印栈中元素

**编写要求：**

**1. 请用C语言完成函数编写，且符合C89语法规则（影响成绩评定）**

**2. 基本操作可以相互调用**

**3. 运行效果必须严格符合要求（影响成绩评定）**

**main函数如下：**

int main(void)

{

SqStack s;

ElemType \*p = NULL;

InitStack(&s);

Push(&s,11);

Push(&s,22);

Push(&s,3);

Push(&s,22);

Push(&s,5);

Push(&s,5);

PrintStack(s);

return 0;

}

**运行效果：**

22为重复元素不可插入!

5为重复元素不可插入!

11 22 3 5

请按任意键继续. . .

**提交格式：**

InitStack函数如下：

JudgRepetition函数如下：

Push函数如下：

PrintStack函数如下：

# 任务04\_2栈的应用——括号匹配

请按照要求，根据**C语言语法**编写函数：matchBracket，该通过调用顺序栈的基本操作，完成括号匹配。

**预定义常量和类型**

#define OK 1

#define ERROR 0

#define Status int

#define STACK\_INIT\_SIZE 20

#define STACKINCREMENT 5

#define ElemType char

**顺序栈结构**

typedef struct {

ElemType \*base;

int top;

int stacksize;

}SqStack;

**顺序栈基本操作函数原型：**

Status InitStack(SqStack \*S)：初始化栈

Status push(SqStack \*S,ElemType e)：入栈

Status pop(SqStack \*S,ElemType \*e)：出栈

**matchBracket函数原型及功能如下：**

Status matchBracket(SqStack \*S,ElemType \*str)：这个函数实现对str指向的字符串判断括号是否能匹配成功，可以调用栈的基本操作函数。

**编写要求：**

**1. 请用C语言完成函数编写，且符合C89语法规则（影响成绩评定）**

**2. matchBracket函数中可以调用栈的基本操作**

**3. 仅提交matchBracket函数**

**4. 运行效果必须严格符合要求（影响成绩评定）**

**main函数如下：**

int main(void)

{

SqStack s;

ElemType e;

char str[20];

printf("请输入需要判断的字符串：");

scanf("%s",str);

matchBracket(&s,str);

return 0;

}

**运行效果1：**

请输入需要判断的字符串：{[(q)(a)]}<(s)>

括号匹配成功！

请按任意键继续. . .

**运行效果2：**

请输入需要判断的字符串：{[(q)(a)]}<(s)

左括号太多，匹配失败！

请按任意键继续. . ..

**运行效果3：**

请输入需要判断的字符串：{[(q)(a]]}<(s)>

匹配错误，有左括号没遇到与之对应的右括号！

请按任意键继续. . .

**运行效果4：**

请输入需要判断的字符串：{[(q)(a)]}<(s)>w)

右括号太多，匹配失败！

请按任意键继续. . .

**提交格式：**

matchBracket函数如下：

# 任务04\_3 队列的实际应用

请按照要求，用**C语言语法，**编程建立循环队列存储结构，对汽车进停车场过程进行模拟，要求程序在控制台屏幕上显示菜单：

============================================

| 1. 排 队 |

| 2. 入 库 |

| 3. 打 印 |

| 4. 退 出 |

============================================

**菜单含义**

1.排队：输入车牌号，加入进停车队列中。

2.进入：排队队列中最前面的车先进入，并将其从队列中删除。

3.打印：将队列中所有车辆的车牌号打印。

4.退出：结束系统。

**题目要求**：

① 由于排队场地有限，队伍长度不超过5；

② “排队”时，若队满，提示等待；

③ 进入时，若队空，提示无人排队；

④ 所有车辆必须经过排队才能进入。

**预定义常量和类型**

#define Status int

#define OK 1

#define ERROR 0

#define m 5

#define CHARLEN 10

typedef char\* QElemType;

**循环队列结构**

typedef struct{

QElemType Q[m];

int length;

int rear;

}SqQueue;

**循环队列基本操作函数原型：**

Status InitQueue(SqQueue \*Sq)：初始化队列

Status EnQueue(SqQueue \*Sq,QElemType e)：入队

Status DeQueue(SqQueue \*Sq,QElemType \*e)：出队

Status TraQueue(SqQueue Sq)**：遍历打印队列中元素**

**编写要求：**

**1. 请用C语言完成函数编写，且符合C89语法规则（影响成绩评定）**

**2. 运行效果必须严格符合要求（影响成绩评定）**

**main函数如下**

int main(void)

{

SqQueue Squeue;

int i;

char e[CHARLEN] = {0}, choice;

char \*pe = e;

InitQueue(&Squeue);

while(1)

{

system("cls");

printf(" ============================================\n");

printf("| 1. 排 队 |\n");

printf("| 2. 入 库 |\n");

printf("| 3. 打 印 |\n");

printf("| 4. 退 出 |\n");

printf(" ============================================\n");

printf("请选择:");

fflush(stdin);

scanf("%c",&choice);

switch (choice)

{

case '1':

printf("请输入车牌号:");

scanf("%s",e);

EnQueue(&Squeue,e);

break;

case '2':

DeQueue(&Squeue,&pe);

break;

case '3':

TraQueue(Squeue);

break;

case '4':

return 0;

default:

printf("选择项输入错误，重新选择！\n");

break;

}

system("pause");

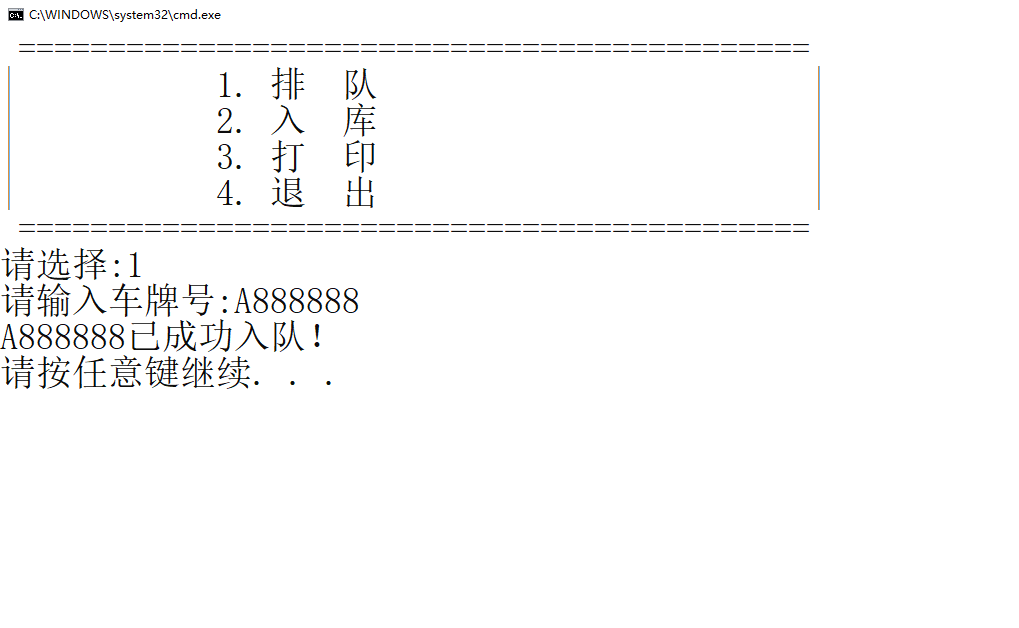
}

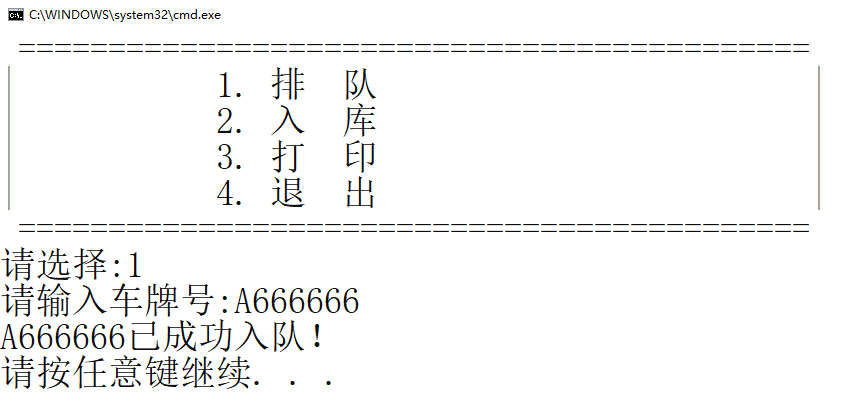
return 0;

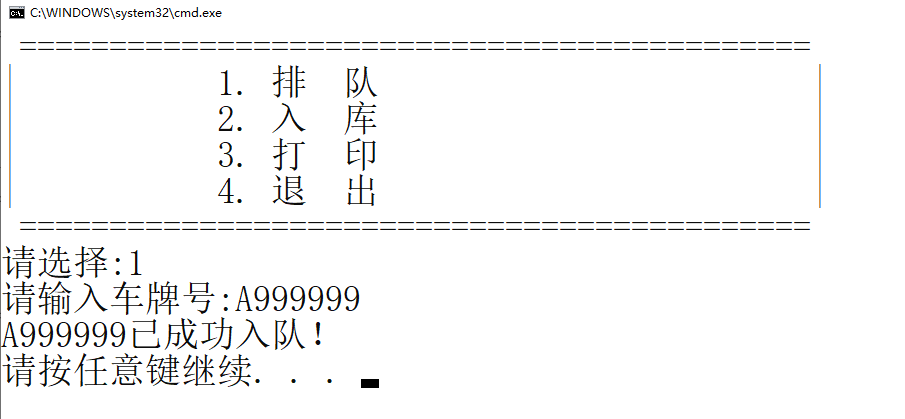
}

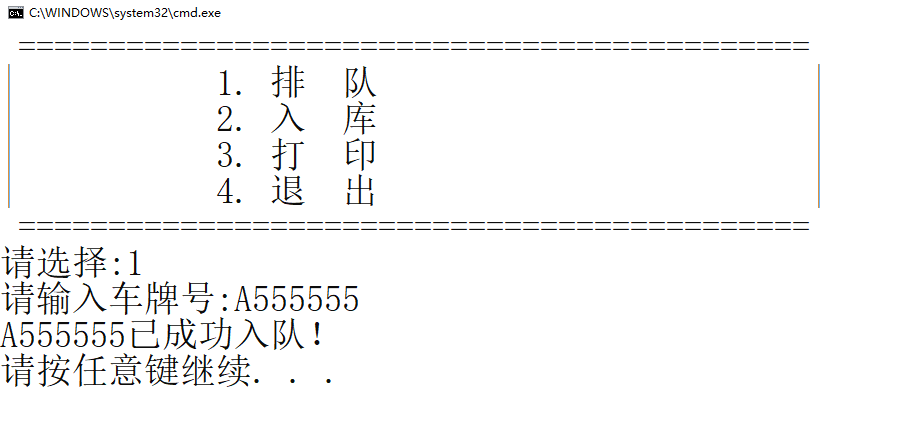
**运行效果：**

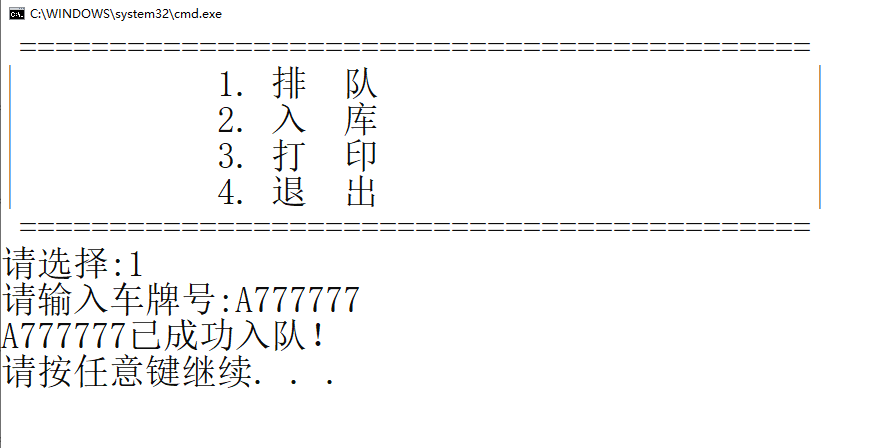
**排队效果：**

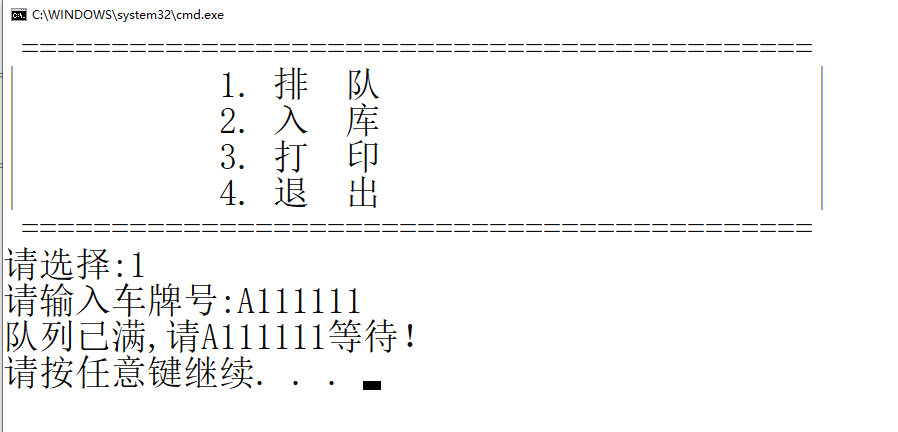




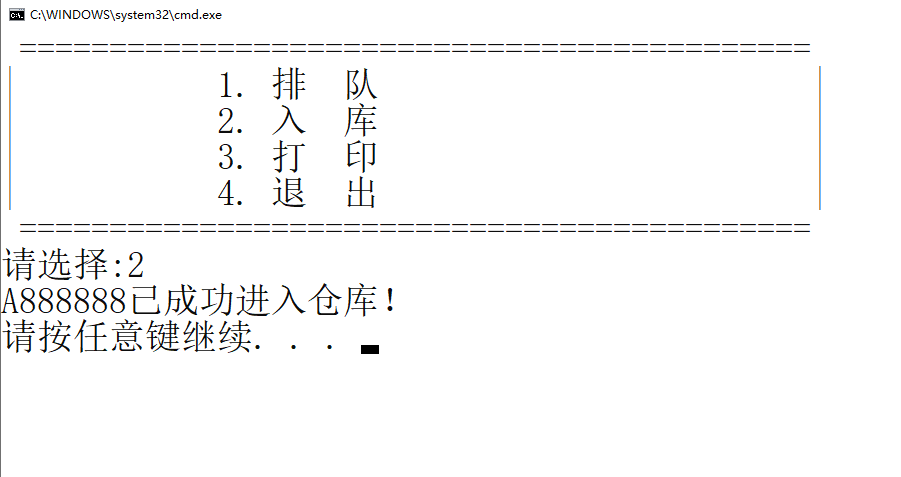




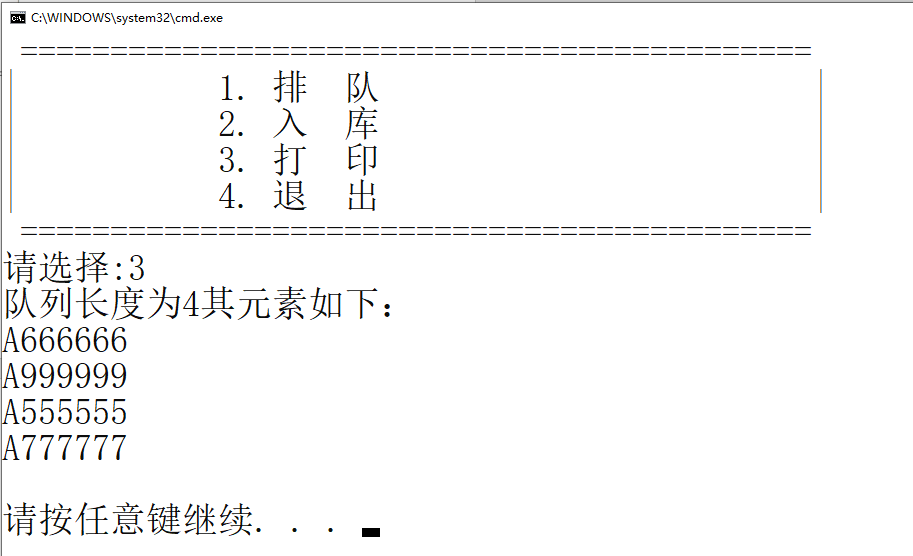




**入库效果**



**打印效果**



**提交格式：**

InitQueue函数如下：

EnQueue函数如下

DeQueue函数如下

TraQueue函数如下