2009853E-I011-0084

毛一凡

## 代码实现

**定义栈的结构体和实现相关方法**

typedef struct {

int\* stk;

int stkSize;

int stkCapacity;

} Stack;//定义栈

Stack\* stackCreate(int cpacity) {

Stack\* ret = (Stack\*)malloc(sizeof(Stack));

ret->stk = (int\*)malloc(sizeof(int) \* cpacity);

ret->stkSize = 0;

ret->stkCapacity = cpacity;

return ret;

}//生成栈的函数

void stackPush(Stack\* obj, int x) {

obj->stk[obj->stkSize++] = x;

}//在栈空间存数

void stackPop(Stack\* obj) {

obj->stkSize--;

}//栈空间数-1

int stackTop(Stack\* obj) {

return obj->stk[obj->stkSize - 1];

}//最上边的数输出

bool stackEmpty(Stack\* obj) {

return obj->stkSize == 0;

}//是否为空

**由栈组成的队列和相关方法**

typedef struct {

Stack\* inStack;

Stack\* outStack;

} Queue;//队列定义 两个栈组成

Queue\* QueueCreate() {

Queue\* ret = (Queue\*)malloc(sizeof(Queue));

ret->inStack = stackCreate(100);

ret->outStack = stackCreate(100);

return ret; //生成“队列 ”

}

void in2out(Queue\* obj) {

while (!stackEmpty(obj->inStack)) {

stackPush(obj->outStack, stackTop(obj->inStack));

stackPop(obj->inStack);

}//将一个栈的数反顺序存到另一个

}

void QueuePush(Queue\* obj, int x) {

stackPush(obj->inStack, x);

}// 队列存数

int QueuePop(Queue\* obj) {

if (stackEmpty(obj->outStack)) {

in2out(obj);

}

int x = stackTop(obj->outStack);

stackPop(obj->outStack);

return x;

}//队列出最上边的数

**测试**

int main() {

Queue\* test = QueueCreate();

int input = 0;

int num = 0;

printf("请输入的值（-1为退出）\n");

while (input != -1) {

scanf("%d", &input);

QueuePush(test, input);

num++;

}

printf("输出值为\n");

while(num-1) {

printf("%d\n", QueuePop(test));

num--;

}

return 0;

**}**

# 实现结果

# 