温州大学计算机与人工智能学院

程序设计课程设计　实验报告

实验名称	银行排队之三：出队				
班　　级	23大数据1	姓　　名	徐王晶	学　　号	23211870102
实验地点	南5B105	实验时间	2023-12-26,19:00:46	指导老师	虞铭财

一、问题编号：

1167

地址：http://10.132.254.54/problem/1167/

二、问题描述：

注意：用链表完成 现在银行都有排队叫号系统，如果你到银行去办理业务，首先取得一个顺序号，并告诉你前面有多少人在等待。 现在请你设计、并实现该系统。 客户信息包括：取号的号码、客户银行卡号码、客户类型（包括普通客户和vip客户，分别用Ordinary和VIP表示）。 本题要求实现以下功能：

（1）客户进入银行并取号，该功能对应的命令为IN number type，number表示客户银行卡号码，type表示客户类型。

（2）在上一题基础上，按取号顺序列出当前未办理业务的客户信息，该功能对应的命令为LIST。

（3）客户办理完业务并离开银行，该功能对应的命令为OUT。

（4）退出系统，该功能对应的命令为QUIT。

三、输入说明：

输入有多行，每行表示一种操作。 本题实现的操作有：

IN guestnumber guesttype 其中IN 表示入队，guestnumber表示客户银行卡号码，guesttype有两种取值，分别为Ordinary和VIP，表示普通客户与VIP客户。

LIST 按取号的顺序输出队列中所有排队的客户信息。

OUT 该命令后面没有参数，在队列中的队首的客户出队。

QUIT 退出银行排队系统

请注意：输入的数据中，只有最后一个命令是QUIT命令。

四、输出说明：

输出有多行，对不同的命令给出不同的输出。 以下是具体的输出说明：

IN 客户银行卡号码　客户类型 该命令首先输出"IN:"，然后再输出客户信息。 客户信息包括客户编号 客户银行卡号码　客户类型 该客户前面的客户数。客户编号由系统在客户取号的时候给定，从1开始顺序编号。 客户类型分为Ordinary和VIP。数据之间用一个空格分开。 请注意： 第一个客户入队后，则该用户的顺序号为1，前面有0个客户等待； 第二个客户入队后，则该用户的顺序号为2，前面有1个客户等待。 以后一直顺序延续。 以此类推。

LIST 该命令首先在单独的一行中输出"LIST:"，后面有若干行按客户取号的顺序输出，每一行输出一个客户的信息，每行的输出格式为 客户编号 客户银行卡号码 客户类型

OUT 该命令首先在单独的一行中输出"OUT:"，接下来一行输出办理业务的客户信息，包括顺序号，客户银行卡号码和客户类型。 如果没有办理业务的客户，则输出"FAILED:"。

QUIT 在单独的一行中显示"GOOD BYE!"后结束程序。

五、输入样列：

```
OUT
IN 1000001 Ordinary
IN 2000003 VIP
IN 2000009 VIP
OUT
OUT
IN 2000008 VIP
LIST
QUIT
```

六、输出样列：

```
FAILED:
IN:1 1000001 Ordinary 0
IN:2 2000003 VIP 1
IN:3 2000009 VIP 2
OUT:1 1000001 Ordinary
OUT:2 2000003 VIP
IN:4 2000008 VIP 1
LIST:
3 2000009 VIP
4 2000008 VIP
GOOD BYE!
```

七、解答内容：

所用语言：

源代码：

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

struct Data
{
    char id[20];
    char cla[10];
    int num;
    struct Data *prev;
    struct Data *next;
};

struct Queue
{
    struct Data *head;
    struct Data *tail;
};

void In(struct Queue *queue, int count, int num);
void List(struct Queue *queue);
void Out(struct Queue *queue, int *count);

int main(void)
{
    //freopen("E:\\Document\\Github\\WZU-OJ-Private\\chengxushejikechengsheji\\All doc\\in1.txt", "r", stdin);
    //freopen("E:\\Document\\Github\\WZU-OJ-Private\\chengxushejikechengsheji\\All doc\\out1.txt", "w", stdout);
    struct Queue queue;
    queue.head = NULL;
    queue.tail = NULL;

    int count = 0;
    int num = 0;
    char op[20];
    while (scanf("%s", op), strcmp(op, "QUIT") != 0)
    {
        if (strcmp(op, "IN") == 0)
        {
            num++;
            In(&queue, count, num);
        }
        else if (strcmp(op, "LIST") == 0)
```

```
        {
            List(&queue);
        }
        else if (strcmp(op, "OUT") == 0)
        {
            count++;
            Out(&queue, &count);
        }
    }
    printf("GOOD BYE!\n");
    return 0;
}

void In(struct Queue *queue, int count, int num)
{
    struct Data *data = (struct Data *)malloc(sizeof(struct Data));
    scanf(
        "%s %s",
        data -> id,
        data -> cla
    );

    if (queue -> head == NULL)
    {
        queue -> head = data;
        queue -> tail = data;
        data -> prev = NULL;
        data -> next = NULL;
        data -> num = num;
    }
    else
    {
        data -> prev = queue -> tail;
        queue -> tail -> next = data;
        data -> num = queue -> tail -> num + 1;
        queue -> tail = data;
        data -> next = NULL;
    }

    printf(
        "IN:%d %s %s %d\n",
        queue -> tail -> num,
        queue -> tail -> id,
        queue -> tail -> cla,
        queue -> tail -> num - 1 - count
    );

}

void List(struct Queue *queue)
{
    printf("LIST:\n");
    struct Data *data = queue -> head;

    while (data != NULL)
    {
        printf(
            "%d %s %s\n",
            data -> num,
            data -> id,
            data -> cla
        );
        data = data -> next;
    }
}

void Out(struct Queue *queue, int *count)
{
    if (queue -> head == NULL)
    {
        printf("FAILED:\n");
        (*count)--;
        return;
    }
    printf(
        "OUT:%d %s %s\n",
        queue -> head -> num,
        queue -> head -> id,
        queue -> head -> cla
    );

    struct Data *data = queue -> head;
    queue -> head = queue -> head -> next;
    free(data);
}
```

八、判题结果

AC - 答案正确

判题结果补充说明：

test id:2029,result:AC, usedtime:0MS, usedmem:864KB,score:100