

第1章《正整数个数统计》实验报告

21051117 卢俊成

主要数据结构和变量

- 链表 (节点 `Node`)
- 加入数字 `void Add(Node*, int)`
- 链上插入排序 `void Sort(Node*)`

程序主要流程

1. 初始化链表
2. 读入数据直到输入为0
3. 链上插入排序
4. 打印链表内容
5. 释放链表

程序主要函数功能

- `void Add(Node*, int)`
顺序查找链表中的数字，若不存在则插入，若存在则次数加1
- `void Sort(Node*)`
链上插入排序
- `void Print(Node*)`
打印链表内容
- `void DelTable(Node*)`
释放链表空间

已实现功能

全部题目要求

编译与运行信息

编译信息

```
/usr/bin/gcc -std=c17 -fdiagnostics-color=always -Wfatal-errors -Wall -Wextra -g  
1.c -o 1 -lm
```

测试数据1

输入

```
1 1 2 2 2 3 4 5 6 3 5 4 0
```

输出

```
2:3
1:2
3:2
4:2
5:2
6:1
```

源代码

```
// Work on stdc17
#include <stdio.h>
#include <stdlib.h>

typedef struct node {
    int number;
    int count;
    struct node* next;
} Node;

void Add(Node* head, int number) {
    Node* current = head;
    while (current->next != NULL) {
        if (current->next->number == number) {
            current->next->count++;
            return;
        }
        current = current->next;
    }
    Node* newNode = (Node*)malloc(sizeof(Node));
    if (newNode == NULL) {
        fprintf(stderr, "Error for malloc.\n");
        exit(1);
    }

    newNode->number = number;
    newNode->count = 1;
    newNode->next = NULL;
    current->next = newNode;
}

void Sort(Node* head) {
    Node* current = head->next;
    if (current == NULL) return;
    Node* tmp;
    while (current->next != NULL) {
        tmp = head;
        while (tmp->next->count >= current->next->count && tmp->next != current->
```

```

>next)
    tmp = tmp->next;
    if (tmp->next != current->next) {
        Node* tmp2 = current->next;
        current->next = tmp2->next;
        tmp2->next = tmp->next;
        tmp->next = tmp2;
    } else
        current = current->next;
}
}

void Print(Node* node) {
    while (node != NULL) {
        printf("%d:%d\n", node->number, node->count);
        node = node->next;
    }
}

void DelTable(Node* node) {
    while (node != NULL) {
        Node* temp = node;
        node = node->next;
        free(temp);
    }
}

int main() {
#ifdef ONLINE_JUDGE
    freopen("1.in", "r", stdin);
    freopen("1.out", "w", stdout);
    freopen("1.err", "w", stderr);
#endif
    int n;
    Node* head = (Node*)malloc(sizeof(Node));
    if (head == NULL) {
        fprintf(stderr, "Error for malloc.\n");
        exit(1);
    }
    head->number = -1;
    head->count = 0;
    head->next = NULL;
    while (scanf("%d", &n), n)
        Add(head, n);
    Sort(head);
    Print(head->next);
    DelTable(head);
#ifdef ONLINE_JUDGE
    fclose(stdin);
    fclose(stdout);
    fclose(stderr);
#endif
    return 0;
}

```

