



南昌大学实验报告

学生姓名： 丁俊 学 号： 8003119100 专业班级： 信安 193 班

实验类型： ☒ 验证 ☐ 综合 ☐ 设计 ☐ 创新 实验日期： 4.22 实验成绩：

一、实验项目名称

递归

二、实验目的

掌握对递归算法的使用和思想

三、实验任务

- 1、编写递归算法 `int max(int a[],int left,int right)`,求数组 `a[left..right]`中的最大数。
- 2、编写一个递归算法函数 `void partion(int a[],int left,int right)`,将数组 `a[left..right]`中的所有奇数调整到数组的左边，所有偶数调整到数组右边。
- 3、已知带头结点的单链表结构定义同实验 3，假设链表中所有结点值均不相同，请编写一个递归函数 `linklist max(linklist head)`，返回表中最大数所在的结点地址，若链表为空返回 NULL。

四、主要仪器设备及耗材

Dec++5.15 windows10

五、实验步骤

1、找到数组中最大的数并返回

```
1. #include "ArrayIo.h"
2. #include <iostream>
3. using namespace std;
4.
5. int max(int a[], int left, int right) {
6.     int mid, l, r;
7.     if (left == right)
8.         return a[left];
9.     else {
10.         mid = (left + right) / 2;
11.         l = max(a, left, mid); // 左区间的最大值
12.         r = max(a, mid + 1, right); // 右区间的最大值
13.
14.         return l > r ? l : r;
15.
16.     }
17.
18. }
19.
20. int main() {
21.     int a[10];
22.     input(a, 10);
23.     print(a, 10);
24.     printf("数组的最大数是:%d\n", max(a, 0, 9));
25.     return 0;
26. }
```

思路：

每次把数组分成两部分，分别找出两部分各自的最大值再进行比较，递归终点是左右边界重合相等，此时不能再把区间一分为二了，那么这个区间的最大值就是它本事。

2、编写递归函数把所有奇数调整到表的左边，偶数跳到右边。

```
1. #include "ArrayIo.h"
2. #define N 10
3.
4. // 左边找偶数,右边找奇数
5. void partion(int a[], int left, int right) {
6.     int temp;
7.     while (left < right) {
8.         while (left < right && a[left] % 2 == 1)
9.             left++;
10.        while (left < right && a[right] % 2 == 0)
11.            right--;
12.        if (left < right) { // 此时 left 和 right 下标对应的数分别是偶数和奇数
13.            temp = a[left];
14.            a[left] = a[right];
15.            a[right] = temp;
16.
17.            partion(a, left + 1, right - 1);
18.        }
19.    }
20.}
21.
22.int main() {
23.    int a[N];
24.    init(a, N);
25.    print(a, N);
26.    partion(a, 0, N - 1);
27.    print(a, N);
28.}
```

思路：

设置两个指针分别指向区间最左侧和最右侧，两个指针同时向中间遍历，当左边出现第一个偶数且右边出现第一个奇数，两个下标对应的数值进行交换。交换完成后再进行递归(left+1,right-1)子区间，直到 left==right。

3、找出带头结点的链表中最大值结点的地址并返回值。

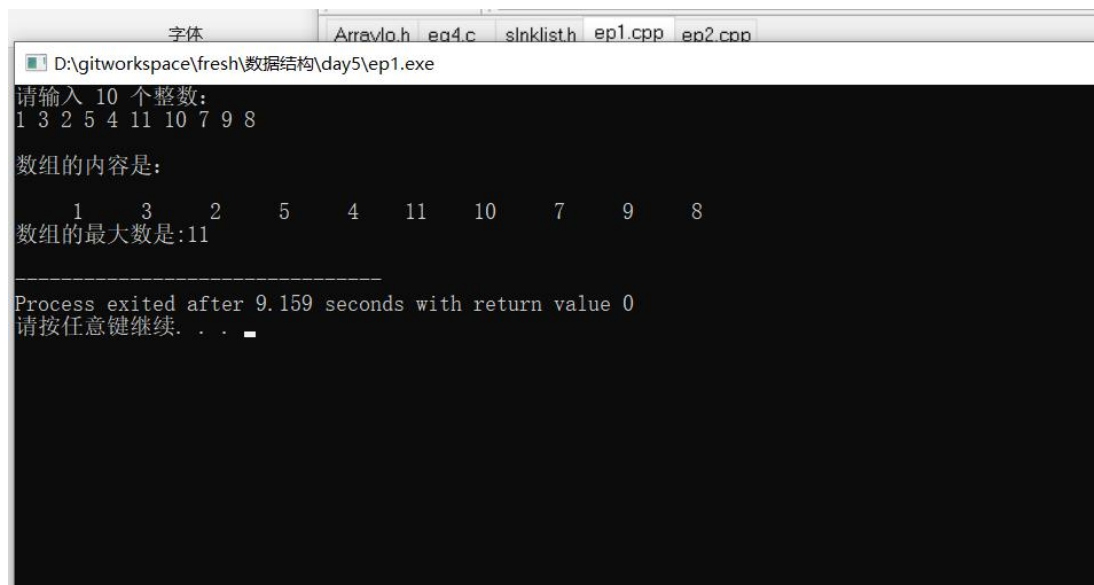
```
1. #include "slnklist.h"
2.
3. // 返回链表中最大数所在的结点地址
4. linklist max(linklist head) {
5.     linklist p;
6.     if (head->next == NULL) { // 表中没有元素
7.         return NULL;
8.     } else if (head->next->next == NULL) { // 表中有一个元素
9.         return head->next;
10.    } else {
11.        p = max(head->next); // 一直递归到最后一个元素，然后
12.        // 回溯，找到 head->next 后面的数中的最大值(不包括 head->next)
13.        if (head->next->info > p->info) {
14.            return head->next;
15.        }
16.        return p;
17.    }
18. }
19.
20. int main() {
21.     linklist head, p;
22.     head = creatbyqueue();
23.     print(head);
24.     p = max(head);
25.     if (p)
26.         printf("max=%d\n", p->info);
27.     else
28.         printf("链表为空\n");
29.     return 0;
30. }
```

思路：

因为是带头结点的单链表，递归寻找除第一个元素后面的所有数种的最大值，然后比较，最后回溯的时候，从最后一个数的最大值找到最后两个数的最大值再到最后三个数的最大值.....一直回溯到第一个数也就找到了整个链表最大数的地址。

六、实验结果

1、递归找最大数



The screenshot shows a C++ IDE with a terminal window. The terminal displays the following text:

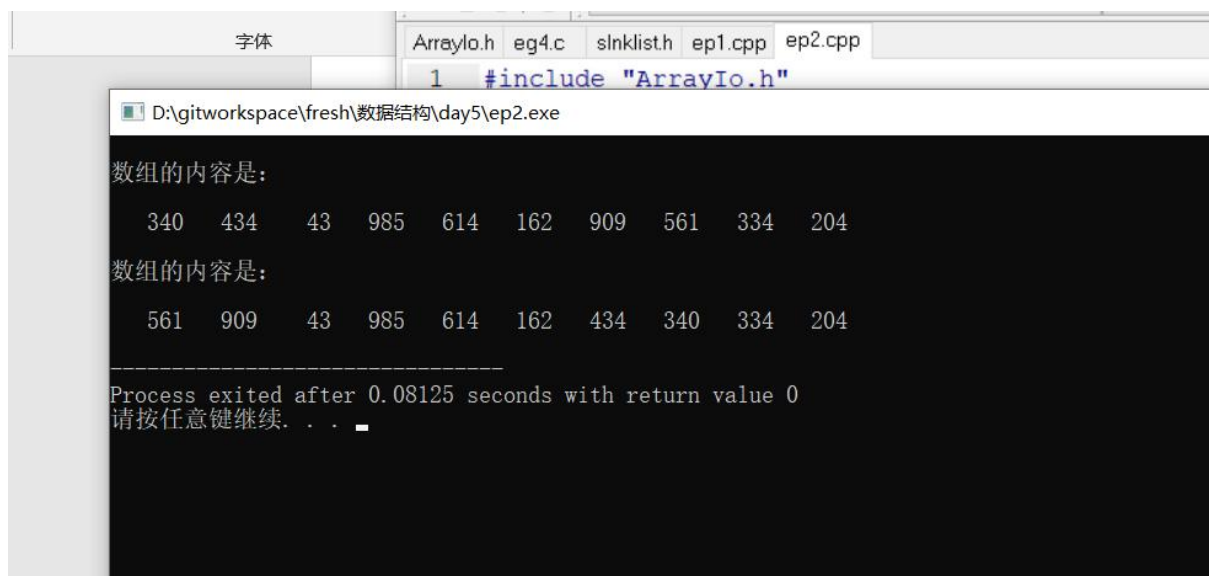
```
D:\gitworkspace\fresh\数据结构\day5\ep1.exe
请输入 10 个整数:
1 3 2 5 4 11 10 7 9 8

数组的内容是:

    1    3    2    5    4    11    10    7    9    8
数组的最大数是:11

-----
Process exited after 9.159 seconds with return value 0
请按任意键继续. . .
```

2、左边奇数右边偶数



The screenshot shows a C++ IDE with a terminal window. The terminal displays the following text:

```
D:\gitworkspace\fresh\数据结构\day5\ep2.exe

数组的内容是:

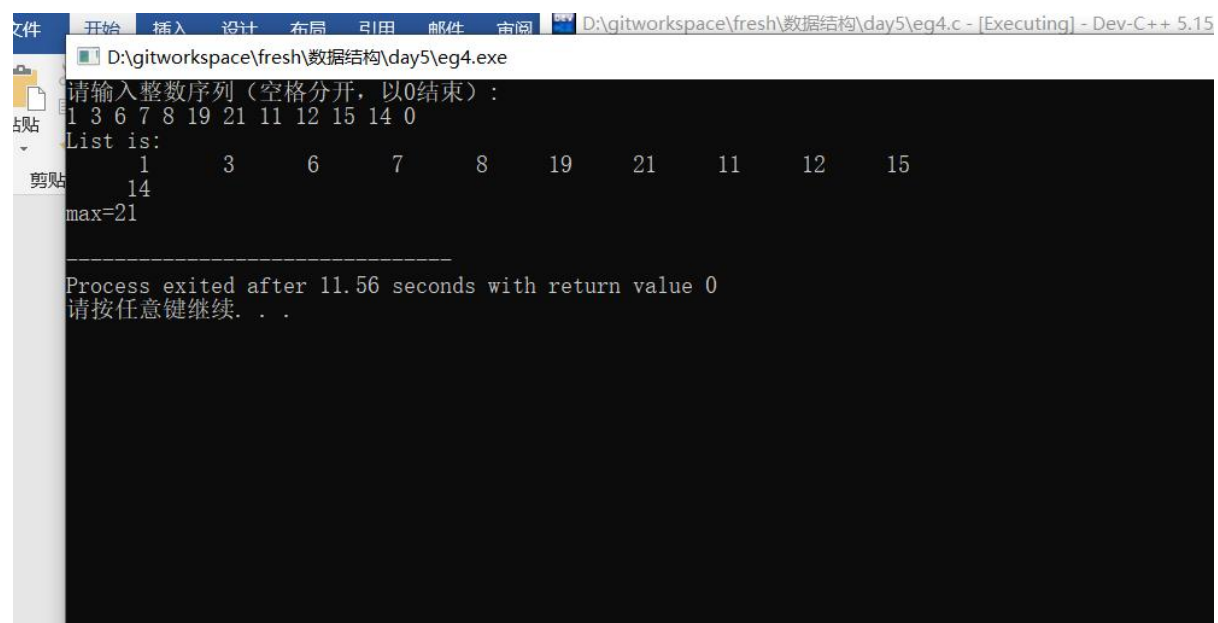
    340    434    43    985    614    162    909    561    334    204

数组的内容是:

    561    909    43    985    614    162    434    340    334    204

-----
Process exited after 0.08125 seconds with return value 0
请按任意键继续. . .
```

3、找到链表最大值地址且返回最大值



```
D:\gitworkspace\fresh\数据结构\day5\eg4.c - [Executing] - Dev-C++ 5.15
D:\gitworkspace\fresh\数据结构\day5\eg4.exe
请输入整数序列（空格分开，以0结束）：
1 3 6 7 8 19 21 11 12 15 14 0
List is:
1      3      6      7      8      19      21      11      12      15
14
max=21

-----
Process exited after 11.56 seconds with return value 0
请按任意键继续. . .
```

七、思考讨论题或体会或对改进实验的建议

八、参考资料