

约瑟夫环

一.实验要求

- 1.完成约瑟夫环的基本功能
- 2.假设命令行参数是齐全的且是正确的，运行所编写的程序能正确地输出结果
3. 能将输出结果导到文件中

二.设计思路

采用单循环链表构成约瑟夫环，链表上的每个节点均为一个结构体，该结构体中包括一个整型变量 num，记录该位置的序号，一个整型变量 secret，记录该位置的关键值，一个 next 指针域，指向下一个节点。在处理约瑟夫环的过程中，使用了 for 循环，每次出圈一个人，并输出他的序号，然后用该节点的关键值代替原关键值，知道圈中所有人均出列。

三.关键代码

```
for(i=1;i<argc;i++) sscanf(argv[i], "%d", &arg[i]);
```

此处将命令行参数取数字存入 arg 数组中，方便直接用 int 进行操作。

```
tail=(single*)malloc(sizeof(single));
head=(single*)malloc(sizeof(single));
head=tail;
for(i=3,j=1;i<argc;i++,j++){
    node=(single*)malloc(sizeof(single));
    node->num=j;
    node->secret=arg[i];
    tail->next=node;
    tail=tail->next;
}
tail->next=head->next; //tail的next域为num=1的结点
```

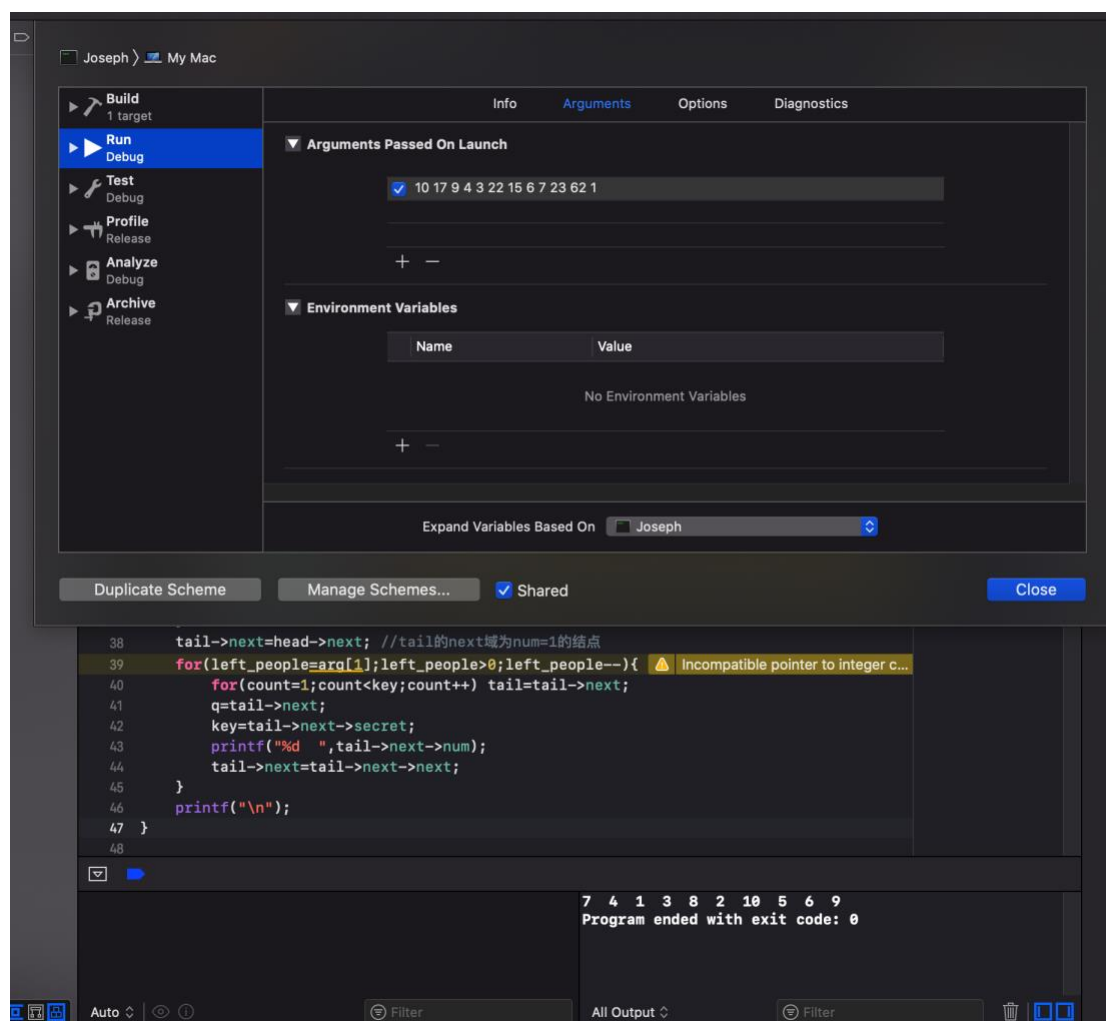
记录 head 节点，且 head 为空节点，head 的 next 为第一个节点，用 tail 尾插法建立链表，然后和 head 相连。

四.调试分析

程序的第一个 for 循环时间复杂度应为 $n-3$ ，第二个 for 循环时间复杂度应为 $(n-3) * key$ ，key 为每个关键值，则整体算法的时间复杂度为 $O(n^2)$ ，如果 key 的关键值也为 n 的数量级。

五.代码测试

测试值和运行结果如下：



六.实验总结：

本次实验使用了但循环链表的结构，锻炼了链表和建立和使用，同时还熟悉了结构体的使用。在链表的建立过程中，采用了尾插法，相比数组的使用，链表需要更清晰的空间结构认识，需要每个节点逐个连接到链表之上，同时删除节点的时候，也需要考虑到上一个节点的操作。

七.附录

Joseph 文件夹中包含两个文件，一个是源代码文件，为 main.c，一个为输出文件，为 out.txt。