

1. 链表未正确建立，linklist_creative 函数中修改 head 指针的指向无法传回主函数。
2. 遍历不成功，因为链表没有成功建立，head 并不是链表的头指针。
3. 链表销毁未成功，因为 head 并不是链表的头指针，无法通过 head 遍历链表进行每个节点的释放。
4. 有丢失内存的现象发生，在 linklist_creative 中 p 申请完空间后一直未释放，会发生丢失内存的情况（但程序退出后系统回收）。
5. 四处修改如下：

① 将链表建立函数 linklist_creative 的定义中参数 student * head 改为 student **head

② 将链表建立函数 linklist_creative 的声明中参数 student * head 改为 student **head

③ 将 main 函数中传入的参数 head 指针改为&head (head 的地址)

④ 将函数 linklist_creative 中的 head = p 改为*head=p

这样传入函数的参数就不是指针，而可以看为指向 head 二级指针，就可以在函数内对 head 的指向进行改变，从而使链表建立，遍历，销毁成功

代码修改和关键部分内存图见下页

```

1  #include <iostream>
2  #include <string> //C++ string类 需要用到
3  using namespace std;
4
5  struct student {
6      string name;
7      int num;
8      char sex;
9      struct student *next; //指向结构体自身的指针(下个结点)
10 };
11
12 #define OK      1
13 #define ERROR   0
14
15 int linklist_create(student **head);
16 int linklist_traverse(student *head);
17 int linklist_destroy(student *head);
18
19 int linklist_create(student **head)
20 {
21     student *p = NULL, *q = NULL;
22     int i;
23
24     for (i = 0; i < 5; i++) {
25         if (i > 0)
26             q = p;
27         p = new(nothrow) student; //思考：为什么不能用malloc
28         if (p == NULL)
29             return ERROR;
30         if (i == 0)
31             *head = p; //head指向第1个结点
32         else
33             q->next = p;
34         cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
35         cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
36         p->next = NULL;
37     }

```

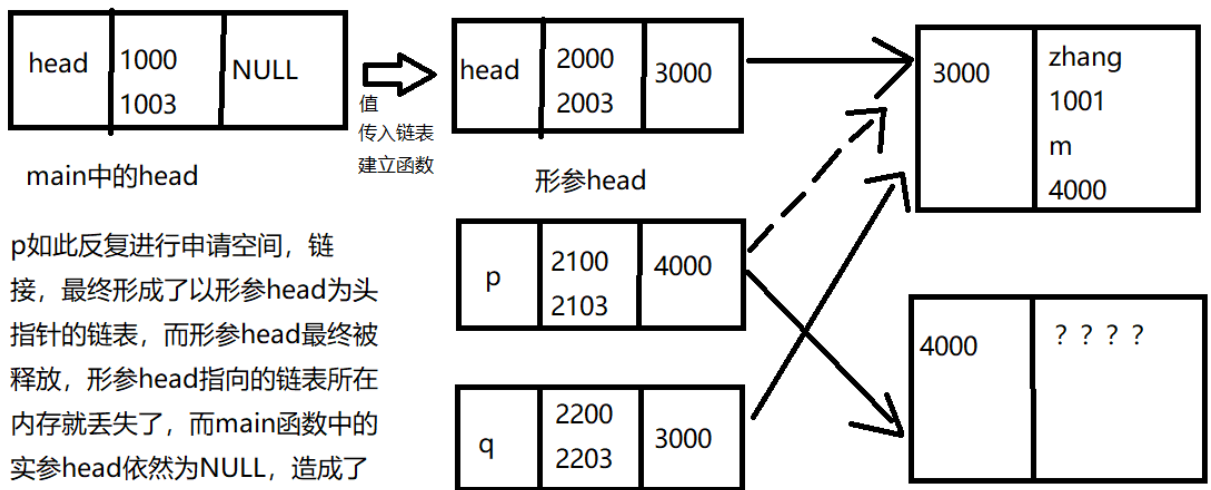
```

int main()
{
    student *head = NULL;
    student *p;
    if (linklist_create(&head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}

```

关键内存图：



错误程序建立链表时的内存图

p如此反复进行申请空间，链接，最终形成了以形参head为头指针的链表，而形参head最终被释放，形参head指向的链表所在内存就丢失了，而main函数中的实参head依然为NULL，造成了错误。所以应该传入实参head的地址来实现在链表建立函数中改变指向