7-b6

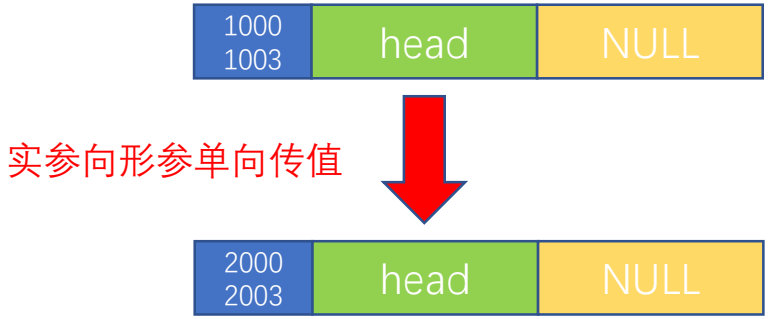
1652228 计算机一班 王哲源

- 1.链表的建立成功了
- 2.遍历不成功的原因在于int linklist_create(student *head)函数中的student *head是形参，对于int main()中的实参head并未进行修改，因此int linklist_traverse(student *head)中的head仍为NULL，无法遍历
- 3.链表并未成功销毁，其原因同2
- 4.程序有发生内存丢失的情况，发生在int linklist_create(student *head)函数的调用过程中，其原因同2
- 以下PPT将通过n内存示意图对以上问题进行解释（红字为当前程序执行到的位置）(此处结构体忽略填充字节)

```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```

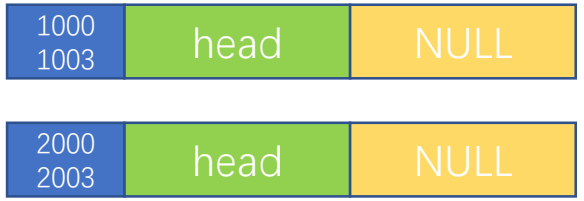
```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```

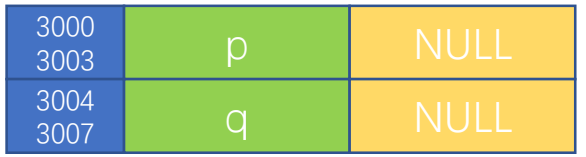
```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```



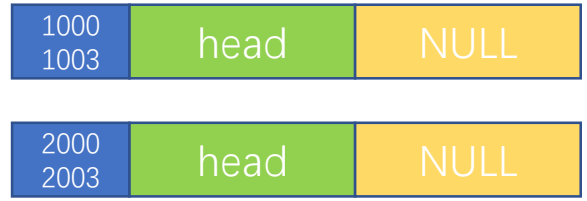
```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

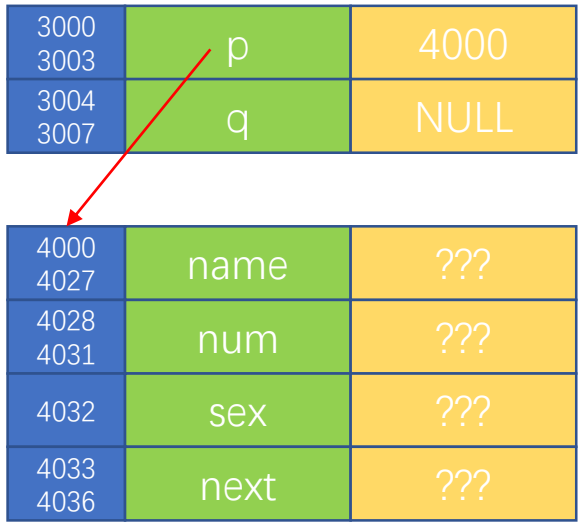
i=0



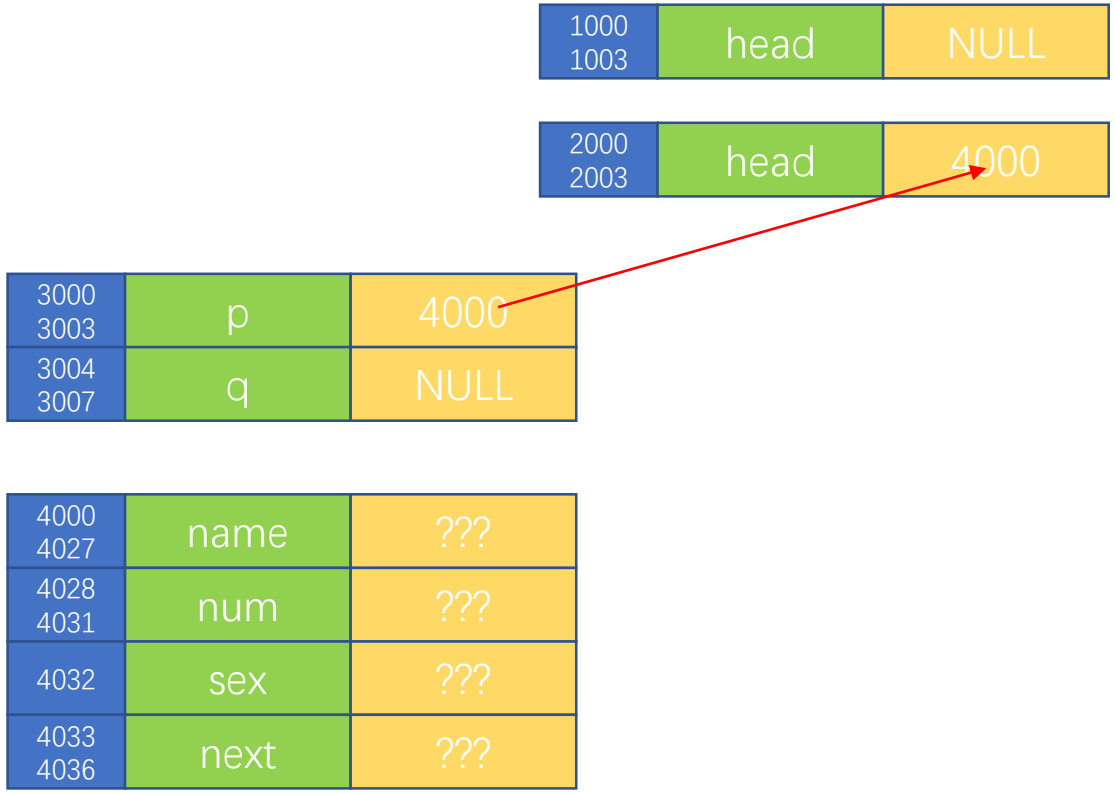


```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

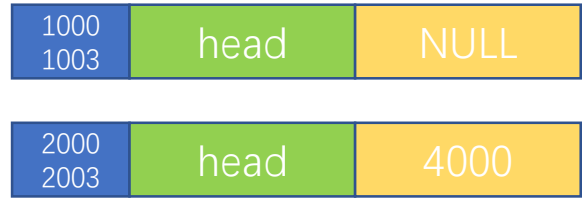


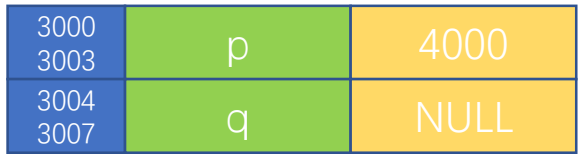


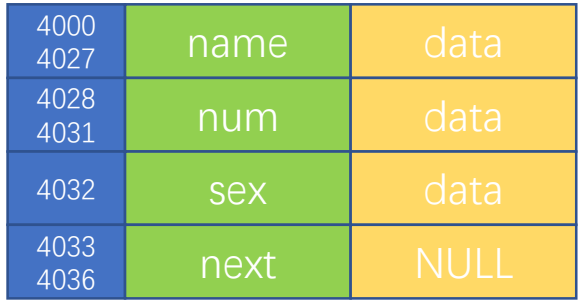
```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```



```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

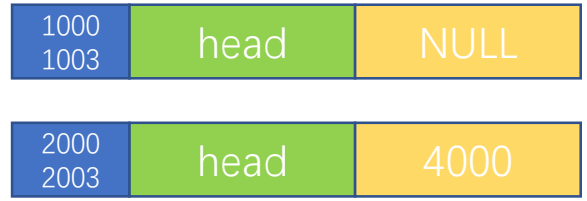


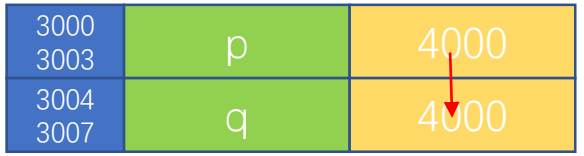


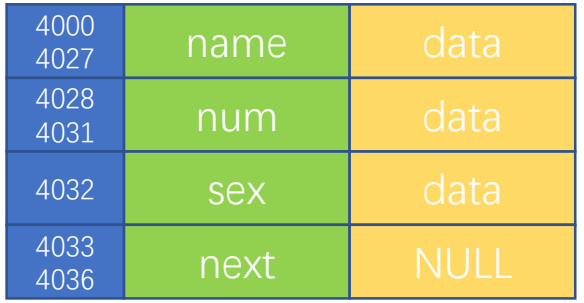


```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

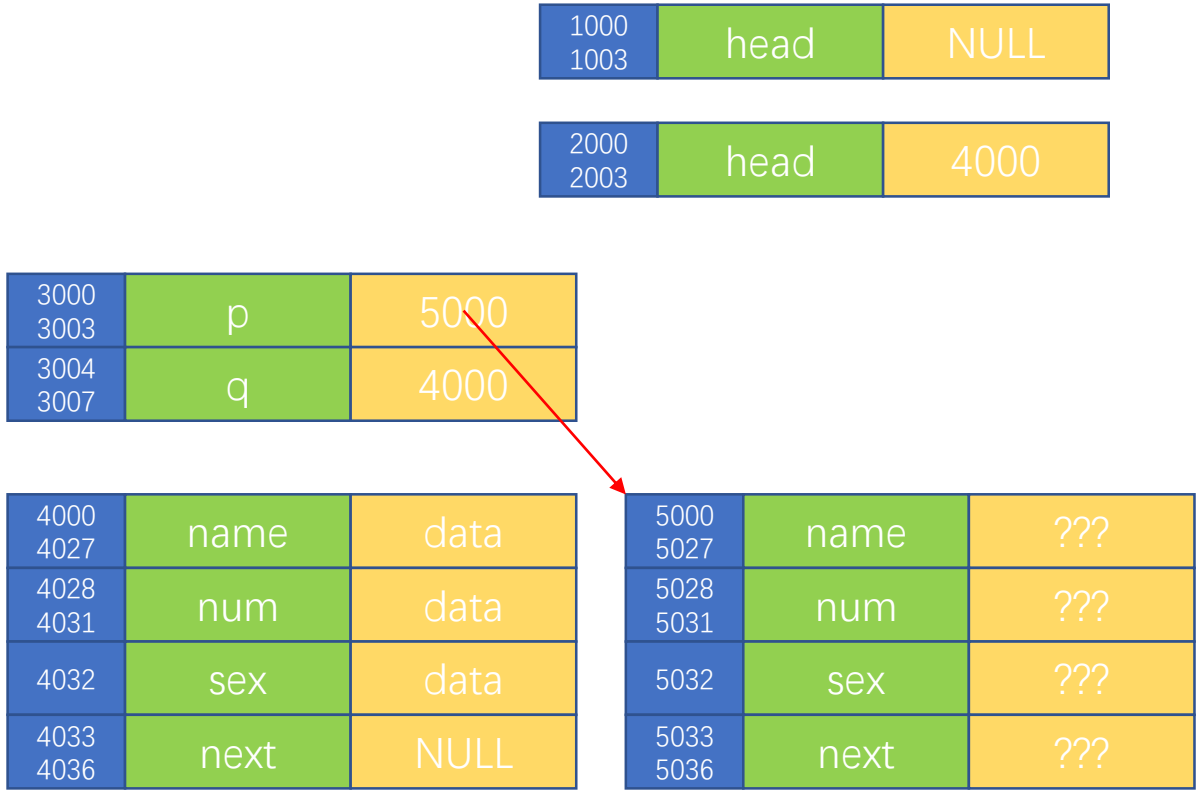
i=1



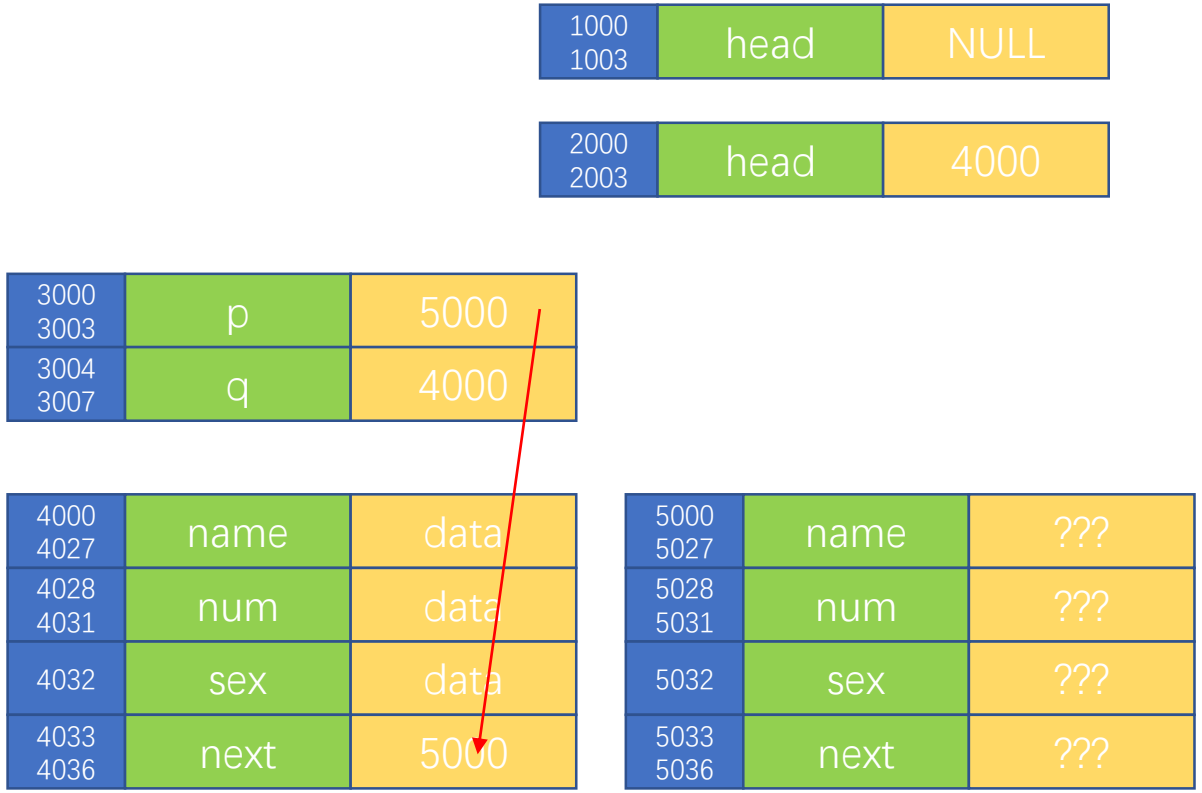




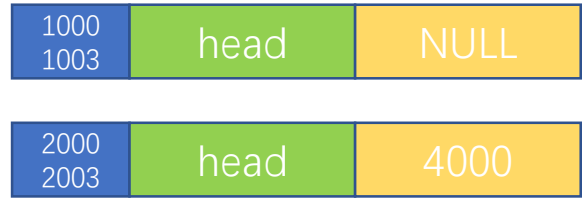
```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

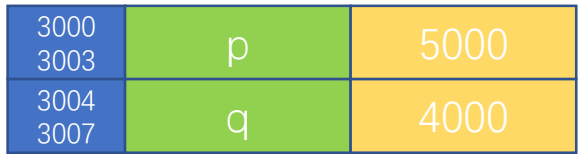


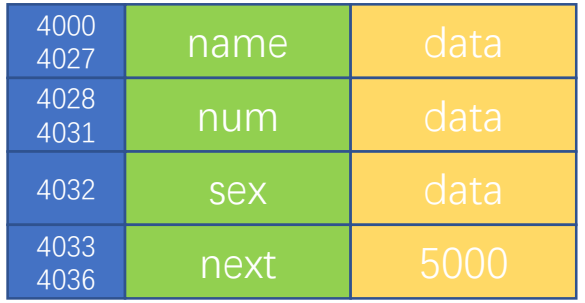
```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

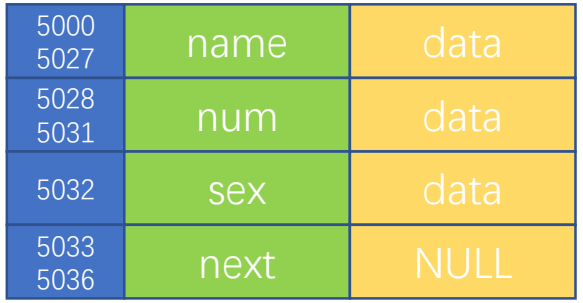


```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```









```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

接下来i=2~4同i=1操作

3000 3003	p	8000
3004 3007	q	7000

1000 1003	head	NULL
2000 2003	head	4000

此处head为形参，并未将值传回main函数

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

链表已经成功建立

```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```

3000 3003	p	???
3004 3007	q	???
2000 2003	head	???

1000 1003	head	NULL

函数执行完毕，内存释放

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_traverse(student *head)
{
    student *p;

    p = head; //p复位，指向第1个结点
    while (p != NULL) { //循环进行输出
        cout << p->name << " " << p->num << " " << p- >sex << endl;
        p = p->next;
    }
    return OK;
}
```

实参对形参赋值

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

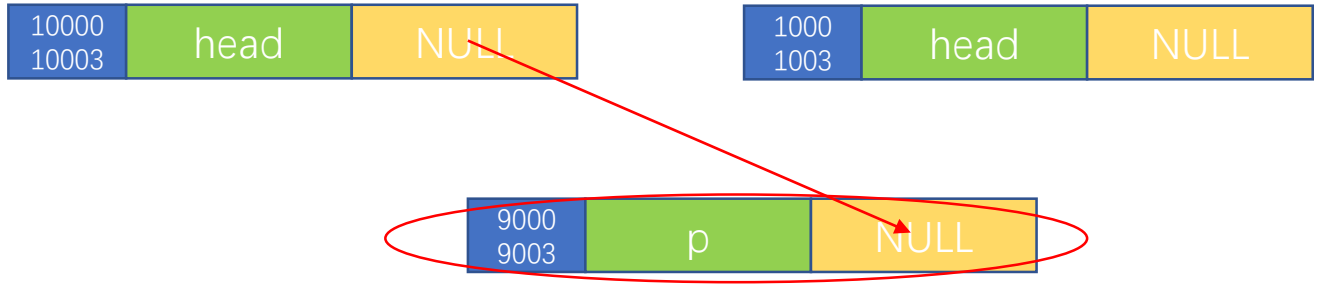
5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_traverse(student *head)
{
    student *p;

    p = head; //p复位，指向第1个结点
    while (p != NULL) { //循环进行输出
        cout << p->name << " " << p->num << " " << p- >sex << endl;
        p = p->next;
    }
    return OK;
}
```



4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_traverse(student *head)
{
    student *p;

    p = head; //p复位，指向第1个结点
    while (p != NULL) { //循环进行输出
        cout << p->name << " " << p->num << " " << p- >sex << endl;
        p = p->next;
    }
    return OK;
}
```

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

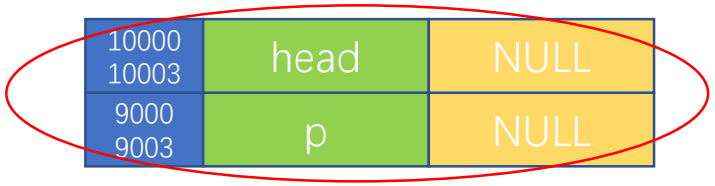
6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

不满足表达式，无法遍历进行输出，结束

```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```



函数执行完毕，内存释放

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_destroy(student *head)
{
    student *p, *q;

    p = head; //p复位，指向第1个结点
    while (p) {  //循环进行各结点释放
        q = p->next;
        delete p;
        p = q;
    }
    return OK;
}
```

实参对形参赋值

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

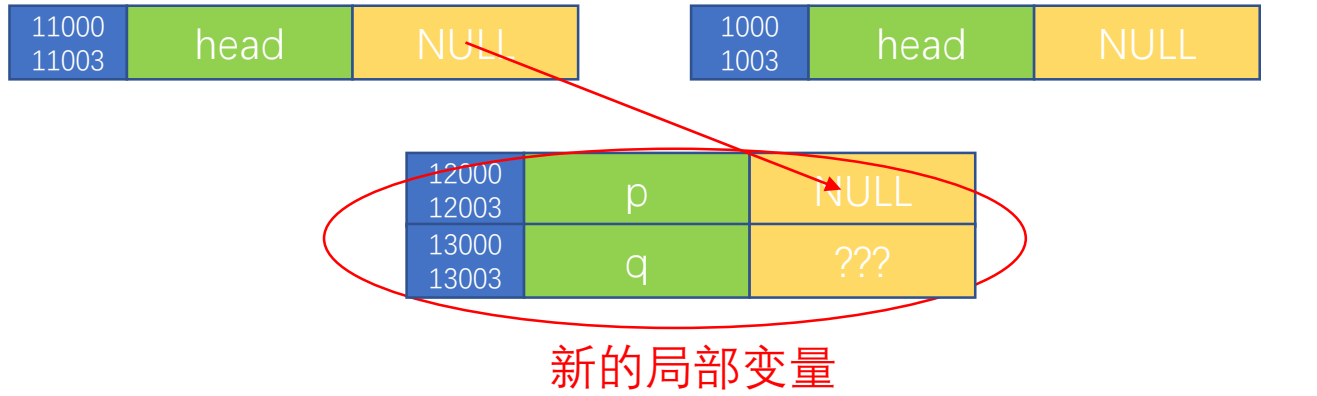
5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_destroy(student *head)
{
    student *p, *q;

    p = head; //p复位，指向第1个结点
    while (p) {  //循环进行各结点释放
        q = p->next;
        delete p;
        p = q;
    }
    return OK;
}
```



4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

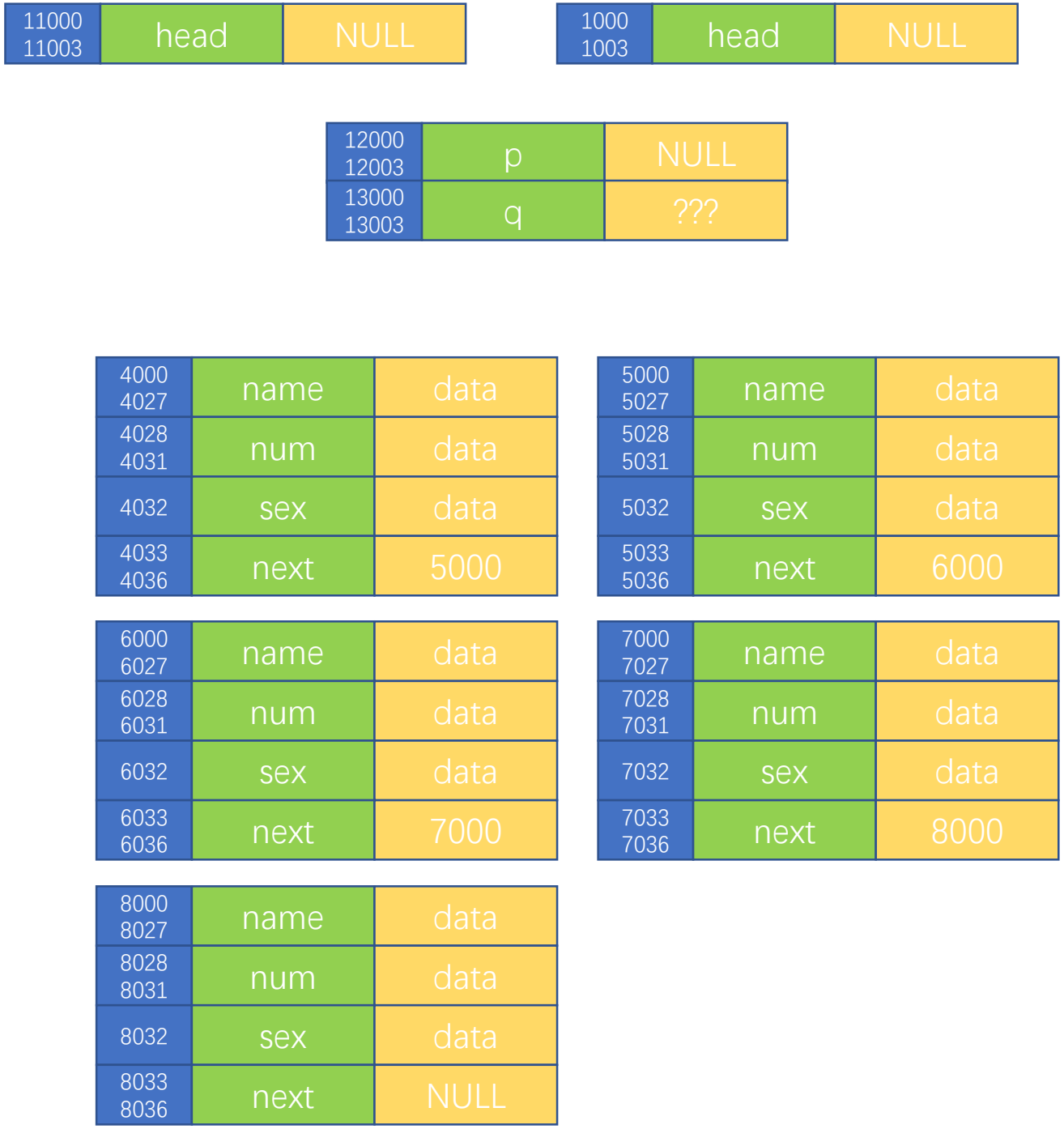
6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

```
int linklist_destroy(student *head)
{
    student *p, *q;

    p = head; //p复位，指向第1个结点
    while (p) {  //循环进行各结点释放
        q = p->next;
        delete p;
        p = q;
    }
    return OK;
}
```

不满足表达式，无法遍历进行释放，结束



```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```

函数执行结束，可以注意到申请的内存空间并未被释放，程序发生内存丢失

11000 11003	head	NULL
12000 12003	p	NULL
13000 13003	q	???

1000 1003	head	NULL

函数执行完毕，内存释放

4000 4027	name	data
4028 4031	num	data
4032	sex	data
4033 4036	next	5000

5000 5027	name	data
5028 5031	num	data
5032	sex	data
5033 5036	next	6000

6000 6027	name	data
6028 6031	num	data
6032	sex	data
6033 6036	next	7000

7000 7027	name	data
7028 7031	num	data
7032	sex	data
7033 7036	next	8000

8000 8027	name	data
8028 8031	num	data
8032	sex	data
8033 8036	next	NULL

- 程序只需要对以下三个地方进行修改即可：

1. main函数中

```
int main()
{
    student *head = NULL;

    if (linklist_create(head) == OK) {
        linklist_traverse(head);
        linklist_destroy(head);
    }
    else
        cout << "LinkList Create failed." << endl;

    return 0;
}
```

linklist_create(&head)

2&3. linklist_create函数中

```
int linklist_create(student *head)
{
    student *p = NULL, *q = NULL;
    int i;

    for (i = 0; i < 5; i++) {
        if (i > 0)
            q = p;
        p = new(nothrow) student; //思考：为什么不能用malloc
        if (p == NULL)
            return ERROR;
        if (i == 0)
            head = p; //head指向第1个结点
        else
            q->next = p;
        cout << "请输入第" << i + 1 << "个人的基本信息" << endl;
        cin >> p->name >> p->num >> p->sex; //键盘输入基本信息
        p->next = NULL;
    }
    return OK;
}
```

student **head

*head = p;