

# 单链表（数组模拟链表）

实现一个单链表，链表初始为空，支持三种操作：

1. 向链表头插入一个数；
2. 删除第  $k$  个插入的数后面的数；
3. 在第  $k$  个插入的数后插入一个数。

现在要对该链表进行  $M$  次操作，进行完所有操作后，从头到尾输出整个链表。

**注意：**题目中第  $k$  个插入的数并不是指当前链表的第  $k$  个数。例如操作过程中一共插入了  $n$  个数，则按照插入的时间顺序，这  $n$  个数依次为：第 1 个插入的数，第 2 个插入的数，...第  $n$  个插入的数。

## 输入格式

第一行包含整数  $M$ ，表示操作次数。

接下来  $M$  行，每行包含一个操作命令，操作命令可能为以下几种：

1. **H x**，表示向链表头插入一个数  $x$ 。
2. **D k**，表示删除第  $k$  个插入的数后面的数（当  $k$  为 0 时，表示删除头结点）。
3. **I k x**，表示在第  $k$  个插入的数后面插入一个数  $x$ （此操作中  $k$  均大于 0）。

## 输出格式

共一行，将整个链表从头到尾输出。

## 数据范围

$1 \leq M \leq 100000$

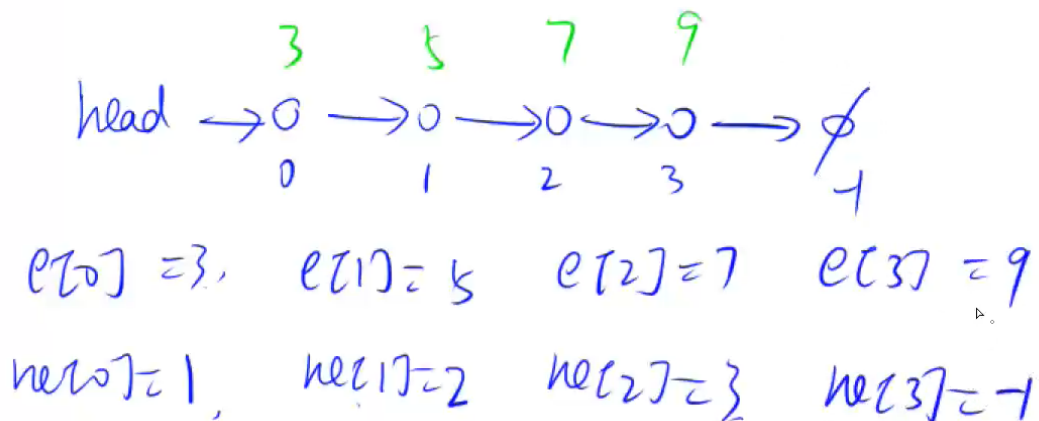
所有操作保证合法。

## 输入样例：

```
10
H 9
I 1 1
D 1
D 0
H 6
I 3 6
I 4 5
I 4 5
I 3 4
D 6
```

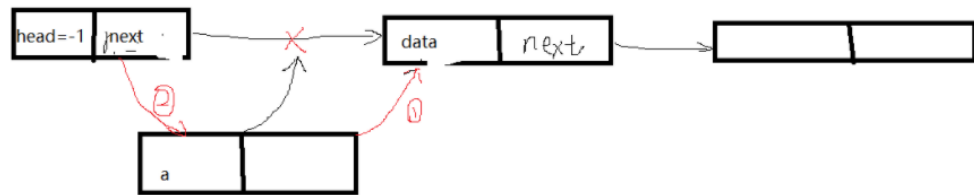
## 输出样例：

```
6 4 6 5
```



// 在链表头插入一个数a

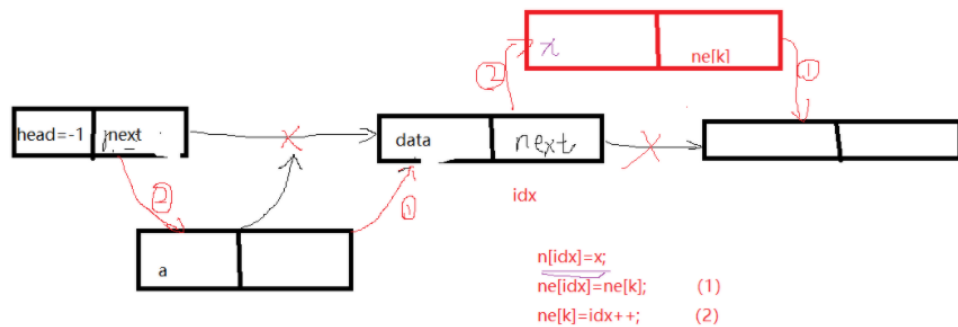
```
void insert(int a)
{
    e[idx] = a, ne[idx] = head, head = idx ++ ;
}
```



解释一下 就先把值赋到数据域，然后让head的地址值存入指针域，让idx向下移一位；

//向表中k位置插图x

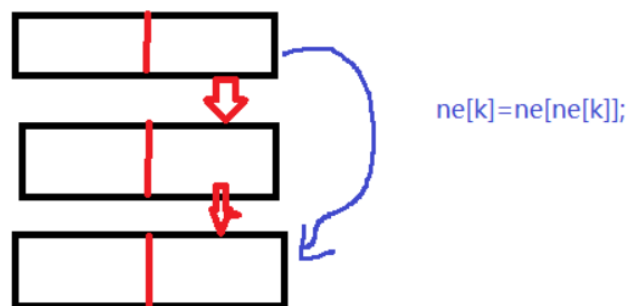
```
void add(int k,int x){
    n[idx]=x;
    ne[idx]=ne[k];
    ne[k]=idx++;
}
```



// 将k删除，需要保证头结点存在

```
void remove(int k)
{
    ne[k]=ne[ne[k]];
}
```

这个与插入思想类似：



y总代码：

```

#include <iostream>

using namespace std;

const int N = 100010;

// head 表示头结点的下标
// e[i] 表示节点i的值
// ne[i] 表示节点i的next指针是多少
// idx 存储当前已经用到了哪个点
int head, e[N], ne[N], idx;

// 初始化
void init()
{
    head = -1;
    idx = 0;
}

// 将x插到头结点
void add_to_head(int x)
{
    e[idx] = x, ne[idx] = head, head = idx ++ ;
}

// 将x插到下标是k的点后面
void add(int k, int x)
{
    e[idx] = x, ne[idx] = ne[k], ne[k] = idx ++ ;
}

// 将下标是k的点后面的点删掉
void remove(int k)
{
    ne[k] = ne[ne[k]];
}

int main()
{
    int m;
    cin >> m;

    init();

    while (m -- )
    {
        int k, x;
        char op;

        cin >> op;
        if (op == 'H')
        {
            cin >> x;
            add_to_head(x);
        }
        else if (op == 'D')
    }
}

```

```

    {
        cin >> k;
        if (!k) head = ne[head];
        else remove(k - 1);
    }
    else
    {
        cin >> k >> x;
        add(k - 1, x);
    }
}

for (int i = head; i != -1; i = ne[i]) cout << e[i] << ' ';
cout << endl;

return 0;
}

```

作者: yxc

链接: <https://www.acwing.com/activity/content/code/content/42977/>

来源: AcWing

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。