实验二 单链表

一、 实验目的

复习 C/C++语法；
学习链表的基本操作。

二、 实验内容

节点定义

```
typedef struct node {
int idx;
int age;
struct node* next;
}Node, *List;
```

1. 编写函数 CreateList()和 PrintList()，从给定数组创建链表，打印链表。

```
int idx[8] = {1,2,3,4,5,6,7,8};
int age[8] = {15,18,13,22,50,18,30,20};
List CreatList(int idx[], int age[],int len){}
int PrintList(List L){}
```

2. 编写函数 DeleteNode(List &L, int delete_age)，完成以下操作。

```
int DeleteNodeAge(List &L, int delete_age){}
```

该函数传入 List L，可以直接修改链表的节点，建议返回值为 int 或 void 类型，无需为 List 类型，3，4 题同上

2.1 删除年龄为 18 的成员，打印链表。
2.2 删除年龄为 20 的成员，打印链表。
2.3 删除年龄为 15 的成员，打印链表。
2.4 删除年龄为 21 的成员（因无此成员，报错），打印链表。

3.编写函数 InsertNodeByIdx(List &L, Node nd)，完成以下操作。（或编写函数 InsertNodeByIdx(List &L, Node *pnd)，完成以下操作。）

（建议用 Node *pnd，因 Node nd 作为参数传给函数 InsertNodeByIdx，nd 本身不能被修改，而插入链表需修改 nd.next，故需创建新的节点把 nd 的 idx 和 age 赋值给新节点。）

3.1 将(idx,age)=(6,23)插入链表，保证链表的 idx 仍为升序，打印链表。
3.2 将(idx,age)=(1,25)插入链表，保证链表的 idx 仍为升序，打印链表。

4.（选做）编写函数 InsertNodeByAge(List &L, Node nd)，完成以下操作。（或编写函数 InsertNodeByAge (List &L, Node *pnd)，完成以下操作。）

4.1 将(idx,age)=(9,31)插入链表，不用保证链表的 idx 仍为升序，新节点插在节点 nd0 后面，要求 nd0.age 是整个链表节点的 age 小于且最接近 nd.age，打印链表。
提示：本例要求插在(7,30)后面。

4.2 插入节点(8,1)
提示：本例要求插在(1,15)之前，即最前面，因 age=1 最小。