

**【注意:】**

- 1、除明确要求外，已学过的知识中，**不允许**使用 goto、**不允许**使用全局变量，**不允许**使用 C++的 string 变量
- 2、多编译器下均要做到 “0 errors, 0 warnings”
- 3、部分题目要求 C 和 C++两种方式实现，具体见网页要求
- 4、如果使用 C 的动态内存申请函数，**必须**是类似于 malloc(10\*sizeof(char)) 的形式，而不能直接 malloc(10)，如果使用 C++的动态内存申请运算符，要求 new(nothrow) 形式
- 5、动态内存申请后，无论申请字节数的多少，**必须**判断申请是否成功
- 6、动态申请的内存，**必须对应释放!!!**（不允许程序退出时由操作系统自行回收）
- 7、违反 4-6 条者，该题得分直接为 0

**补充:**

- 2、假设一个学生的信息包括学号、姓名、成绩三项，从数据文件 student.txt 中读取若干学生的信息，读取完成后，按成绩排序并确定每个人的名次（相同成绩则名次相同，且下一名次要跳过并列名次），每个人的输出信息一行，4 项（自有 3 项+名次，每项间一个空格），输出方法有两种

方法一：按学号从小到大依次输出

**【提示:】**按成绩排完序，求得名次后，再重新以学号排序

```
1950001 张三 95 4
1950009 李四 98 1
1950783 王五 97 3
1951023 赵六 98 1
.....
```

**【注:】**排序方法只能用冒泡法或选择法（数据是结构体，按结构体的某个成员排序）

方法二：按名次从高到低依次输出，名次相同者按学号从小到大排列

```
1950009 李四 98 1
1951023 赵六 98 1
1950783 王五 97 3
1950001 张三 95 4
.....
```

**【提示:】**本题难点在于如何进行多关键字排序，即按成绩排序后，还要按学号排序

**【要求:】**1、数据文件 student.txt 的格式为第 1 行一个整数，代表学生的数量，后面每行 3 个数据，依次为学号(7 位数字)、姓名(2-4 个汉字，不含空格)、成绩(0-100 的整数)

**【假设数据文件正确，不必考虑数据文件出错的情况】**

- 2、给出示例的 13-b2-maketxt.exe，运行一次，可生成一个 student.txt 文件
- 3、要求用两种方法求解：

方法一：动态内存申请用 C 方式的 malloc/calloc/free 系列函数

方法二：动态内存申请用 C++方式的 new/delete 系列操作符

- 4、一个学生的多项信息放入一个结构体中，所有学生的信息**不准**用结构体数组表示，必须用动态内存申请方式(malloc/new)申请一个连续空间，之后对此连续空间可以用数组/指针方式进行处理，要求该空间必须按需分配，**不准**分配多余的空间
- 5、程序中不准使用 string 类型
- 6、程序排序过程中不准再额外定义数组(含动态申请的数组)用于存放排序结果等

7、有关的结构体定义如下，**不允许更改**

```
struct student {
    int no;          //学号，不考虑 0 开头
    char name[9];    //姓名，最长 4 个汉字（无生僻字，均为双字节汉字）
    int score;       //成绩，不考虑小数点
    int rank;        //名次
};
```

- 3、假设一个学生的信息包括学号、姓名、成绩三项，从数据文件 list.txt 中读取若干学生的信息，每读一个学生，就申请一个结点存放信息，所有学生的信息形成一个链表（新读取的学生放在链表的尾部），所有的信息读取完成后，遍历链表，打印所有学生的信息（每个学生一行，3 项，每项间一个空格），遍历完成后，释放链表。要求用两种方法求解：

**方法一：**动态内存申请用 C 方式的 malloc/calloc/free 系列函数

**方法二：**动态内存申请用 C++方式的 new/delete 系列操作符

**【要求：】**1、数据文件 list.txt 的格式为每行 3 个数据，依次为学号(7 位数字)、姓名(2-4 个汉字,不含空格)、成绩(0-100 的整数),读到学号 9999999 则表示输入结束(学号 9999999 **不允许加入链表中**)

**【假设数据文件正确，不必考虑数据文件出错的情况】**

- 2、给出示例的 13-b3-maketxt.exe，运行一次，可生成一个 list.txt 文件

- 3、有关链表的结构体定义如下，**不允许更改**

```
struct student {
    int no;
    char name[9];
    int score;
    struct student *next;
};
```

- 4、链表建立过程中判断申请不成功的错误处理，直接用 return -1;即可，不考虑释放

- 4、题目同 13-b3，所有学生的信息形成一个链表，但需要二次申请，其它要求不变。要求用两种方法求解：

**方法一：**动态内存申请用 C 方式的 malloc/calloc/free 系列函数

**方法二：**动态内存申请用 C++方式的 new/delete 系列操作符

**【要求：】**1、仍然从 list.txt 中读取数据，规则不变

- 2、有关链表的结构体定义如下，**不允许更改**

```
struct student {
    int *no;          //学号
    char *name;       //姓名
    int *score;       //成绩
    struct student *next;
};
```

- 3、结构体的 name 成员必须**按需分配**（例：名字两个汉字，只允许申请 5 字节）

- 4、链表建立过程中判断申请不成功的错误处理，直接用 return -1;即可，不考虑释放

- 5、模拟随机抽取程序，假设某重点初中今年招生名额为 N 人，而报名人数高达 M 人，为显示公平，要求做一个随机抽取程序，从 M 人随机抽取 N 人（保证  $N \leq M$ ）

**【要求：】**1、报名者的基本资料放在 stulist.txt 中，该文件的第 1 行为招生人数 N，第 2 行为报名总人数 M，后面每行是每个报名者的基本信息（报名号、姓名、毕业学校，其中“姓名+数字”当做一个完整的姓名处理）

- 2、将随机抽中的学生信息放入 result.txt 中（不需要显示在屏幕上），每个学生一行，基本信息之间用空格分隔即可
- 3、屏幕上有一些简单的提示性输出即可，具体内容自定
- 4、按需申请，不准浪费空间（思考：应该是申请连续空间当数组用，还是申请链式空间？那种方法好？在源程序最后用注释语句简单说明你的选择理由即可）
- 5、给出 13-b5-maketxt.exe，运行一次，可生成一个 stulist.txt 文件
- 6、必须保证抽取结果中不能有重复的学生
- 7、如果选用链表方式，则链表建立过程中判断申请不成功的错误处理，直接用 return -1;即可，不考虑释放

6、课件的 P. 54-P. 80 给出了一个链表的示例程序及内存分析，完整地展现了一个链表的建立、遍历及销毁的完整过程，是一个正确的程序。现在要求：改写这个程序，将链表的建立、遍历及销毁分为三个函数分别实现，然后 main 中依次调用这三个函数，但是附件中目前给出的这个程序是错误的。

现要求回答下列问题，其中关键部分画出内存示意图：

- 1、链表的建立是否正确？
- 2、为什么遍历不成功？
- 3、链表的销毁是否成功了？
- 4、程序是否有内存丢失情况发生？如果有，发生在哪个函数被调用的阶段？
- 5、只允许修改某个函数的参数类型/该函数的声明，并在该函数内部改动一个地方，main 函数调用处改动一个地方，使程序正确，应该如何改动？（一共修改四处，且 **不允许用引用**，**不允许** 改动函数返回类型）

### 【编译器要求：】

		编译器VS	编译器Dev	编译器Linux
13-b1. pdf	realloc专项讨论	/	/	/
13-b2-1. c	成绩排序(学号输出, C方式)	Y	Y	Y
13-b2-2. cpp	成绩排序(学号输出, C++方式)	Y	Y	Y
13-b2-3. c	成绩排序(名次输出, C方式)	Y	Y	Y
13-b2-4. cpp	成绩排序(名次输出, C++方式)	Y	Y	Y
13-b3-1. c	链表(C方式)	Y	Y	Y
13-b3-2. cpp	链表(C++方式)	Y	Y	Y
13-b4-1. c	链表二次申请(C方式)	Y	Y	Y
13-b4-2. cpp	链表二次申请(C++方式)	Y	Y	Y
13-b5. cpp	随机抽取	Y	Y	Y
13-b6. pdf	链表参数传递思考题	/	/	/

### 【作业要求：】

- 1、**10月17日前**网上提交本次作业
- 2、每题所占平时成绩的具体分值见网页
- 3、超过截止时间提交作业则不得分