实验报告#4 - 顺序表与链式表

开发环境

- 编译器：gcc v6.3.0
- 编辑器：VS Code
- 平台：X86-64
- OS：win10 home

程序说明

基本运行逻辑

1. 运行程序
2. 选择表的类型
    - a 顺序表
    - l 线性表
3. 选择表的操作
    - i 在指定位置插入指定数据
    - e 删除指定位置的数据
    - q 退出这个舞台

线性表 array_list

- 定义在 `myList.hpp` 中
- 使用定长（100）一维数组储存数据，插入删除时间复杂度均为O(n)

```
class array_list
{
    char data[100];
    int size;
  public:
    array_list();
    bool insert(char ch, int pos);
    bool erase(int pos) ;
    void print_all();
};
```

链式表 link-list

- 定义在 `myList.hpp` 中
- 使用(头结点储存数据的)链表储存数据，插入删除时间复杂度均为O(1)

```
struct Node
{
    char data;
    Node *next;
    Node(char d);
};

class link_list
{
    int size;
    Node *head;
  public:
    link_list();
    bool insert(char data, int pos);
    bool erase(int pos);
    void print_all();
};
```

程序运行结果

线性表的基本操作

```
//插入
List based on [a]rray or based on [l]inklist?
a
- [i]nsert
- [e]rase
- [q]uit
i
- data: c
- position: 0
- [current data field]c
i
- data: o
- position: 1
- [current data field]c o

//太长省略一部分

i
- data: r
- position: 7
- [current data field]c o m p u t e r

//删除
e
- position: 0
- [current data field]o m p u t e r
e
- position: 3
- [current data field]o m p t e r
e
- position: 5
- [current data field]o m p t e
```

链式表的基本操作

```
//插入
List based on [a]rray or based on [l]inklist?
l
- [i]nsert
- [e]rase
- [q]uit
i
data: c
position: 0
- [current data field]c
i
data: o
position: 1
- [current data field]co
i
data: m
position: 2
- [current data field]com
i
data: p
position: 3
- [current data field]comp

//删除
e
position: 2
- [current data field]cop
```

异常处理

```
//两部分表的异常处理类似，不单独列出

//在非法位置插入/删除 -> 提示非法并退回操作选项
e
position: 3
invaild position
- [current data field]cop

//插入超长的数据 -> 仅插入首字母
i
data: adsiujfhjadsfajdsnlkjhcvjnrdg
position: 0
- [current data field]acop

//输入非法操作选项 -> 显示当前数据，并退回操作选项
asdfuih
- [current data field]acop
```

info

16340247 席睿 软件工程教务三班