数据结构与算法第三次作业

> 陈梓乐 19336035

> 部分代码可在 gitee.com/Czile/homework 中找到

设计循环链表表示队列，并且只设置一个尾结点，设计出入队列的算法

前置条件：参照循环链表的代码：https://gitee.com/Czile/homework/blob/master/02/extra-cycle-list.h.

现在，我们来设计队列的代码：

```
template <class T>
class que {
    public:
        void push(const T& v) {
            p.insert(p.end(), v);
        }
        T pop() {
            auto t = p.end() -> next -> val;
            p.erase(p.end() -> next);
            return t;
        }
        void print() const {
            p.print();
        }
    private:
        cyclelist<T> p;
};
```

利用循环链表重排数目为偶数的栈中的元素，要求偶数项在栈顶，奇数项在栈底。

前置条件：包含循环队列的代码 https://gitee.com/Czile/homework/blob/master/02/extra-cycle-list.h 以及标准库 `<stack>` 中 `std::stack<T>` 的代码，现在对 `stack<T>` 中的内容进行重排：

```
template <class T>
void reVal(stack<T> & t) {
    cyclelist<T> q;
    auto p = q.begin();
    while (t.size()) {
        if (t.size() % 2)
            p = q.insert(p, t.top()) -> next;
        else
            q.insert(q.end(), t.top());
        t.pop();
    }
    t.reverse();
    t.push(p -> val);
    for (auto tmp = p -> next; tmp != p; tmp = tmp -> next == q.end() ? tmp -> next -> next : tmp -> next)
        t.push(tmp -> val);
}
```

循环数组中设计队列的出入队算法

前置条件：包含顺序表的代码 https://gitee.com/Czile/homework/blob/master/02/01.h. 下面，我们来实现队列：

```
template <class T, unsigned n>
class que {
    public:
        que(): flag(0u), first(0u), last(0u) {}
        bool push(T& v) {
            if ((first == last) && flag) {
                return 1;
            } else {
                p[last] = v;
                last = (last + 1) % n;
                flag = 1;
                return 0;
            }
        }
        T pop() {
            if (flag) {
                auto tmp = p[first];
                first = (first + 1) % n;
                if (first == last)
                    flag = 0;
                return tmp;
            }
            return T();
        }
        void print() {
            if (flag) {
                auto t = first + 1;
                cout<<p[first]<<" ";
                for (; t != last; t = (t+1) % n)
                    cout<<p[t]<<" ";
            }
            cout<<endl;
        }
    private:
        clist<T, n> p;
        unsigned first, last;
        bool flag;
};
```

顺序表、单链表、循环链表、双链表、静态链表的头文件

本题目已经在上次上机作业提交，而根据要求，在此补充静态链表的定义代码：

```
#ifndef _STATICLIST_H_
#define _STATICLIST_H_

#include <algorithm>
#include <iostream>

template <class T>
class staticlistnode {
    public:
        T val;
        unsigned next;
};


// define a static linkedlist of type T, with _nlist linkedlist, and totally _size elements.
template <class T, unsigned _nlist, unsigned _size>
class staticlist {
    public:
        // Constructor of the static chain.
        staticlist();
        // Destroys the static chain.
        ~staticlist();
        // Insert the element in the position.
        // Return the position of the element.
        // Para: first is the position, second is the value, third is the nlistth list=0.
        unsigned insert(unsigned, const T&, unsigned nlist = 0);
        // Delete the element in the position.
        // Return the position of the next element.
        // If return _size, it's to show that the length of the list is not enough.
        unsigned erase(unsigned);
        // Return the position of the first element of the kth linkedlist.
        unsigned begin(unsigned) const;
        // Return the position of the last + 1 element of all the linkedlist.
        unsigned end() const noexcept;
        // Print all the list in the screen.
        void print();
    private:
        unsigned *_first, _last, _free;
        staticlistnode<T> *p;
};
```

其余全部代码请于 gitee.com/Czile/homework 处查看。

画图演示增加、删除：

顺序栈、链表栈、共享栈、循环队列、链表队列的头文件和图示

关于链表队列以及循环队列，参照本作业的 `第一题` 与 `第三题` ，此处不再赘述，接下来实现链表栈：

链表栈

前置条件： 包含链表的代码 https://gitee.com/Czile/homework/blob/master/02/02.h

```
template <class T>
class stack {
    public:
        void push(T& v) {p.insert(p.begin(), v);}
        void pop() {p.pop(p.begin());}
        T & top() {return p.begin() -> val;}
        void print();
    private:
        linkedlist<T> p;
};
```

顺序栈

前置条件：包含顺序表的代码 https://gitee.com/Czile/homework/blob/master/02/01.h

```
template <class T, unsigned n>
class stack {
    public:
        stack(): flag(0u), first(0u), last(0u) {}
        bool push(T& v) {
            if (last == p.end())
                return 1;
            p[last++] = v;
            return 0;
        }
        T pop() {
            return p[--last];
        }
        void print() {
            for (auto t = first; t != last; t = (t+1) % n)
                cout<<p[t]<<" ";
            cout<<endl;
        }
    private:
        clist<T, n> p;
        unsigned first, last;
};
```

共享栈

```
template <class T, unsigned n>
class stack {
    public:
        stack(): flag(0u), first(0u), last(n) {}
        bool push(T& v, int t=0) {
            if (first == last)
                return 1;
            !t ? p[first++] = v : p[--last] = v;
            return 0;
        }
        T pop(int t = 0) {
            if (t && last == n || !t && !first)
                return T();
            return !t ? p[--first] : p[last++];
        }
        void print() {
            for (auto t = 0; t != first; t = (t+1) % n)
                cout<<p[t]<<" ";
            cout<<endl;
            for (auto t = n-1; t != last - 1; t = (t-1) % n)
                cout<<p[t]<<" ";
            cout<<endl;
        }
    private:
        clist<T, n> p;
        unsigned first, last;
};
```

图示如下：

栈与队列的完整代码

参照第四、第五题，已经给出了完整的代码。