

数据结构与算法 课程实验报告

学号：201700130033	姓名：武学伟	班级：2017 级 2 班
实验题目：队列		
实验学时：4	实验日期：2018.11.11	
实验目的： 1. 掌握队列结构的定义与实现。 2. 掌握队列结构的使用。		
软件环境： Win10home, codeblocks, sublime		
1. 实验内容（题目内容，输入要求，输出要求） 1. 创建队列类，采用链式描述； 2. 实现卡牌游戏 假设桌上有一叠扑克牌，依次编号为 1-n（从最上面开始）。当至少还有两张的时候，可以进行操作：把第一张牌扔掉，然后把新的第一张放在整叠牌的最后，输入 n，输出每次扔掉的牌，以及最后剩下的牌。 2. 数据结构与算法描述（整体思路描述，所需要的数据结构与算法） 数据结构： 队列 算法： 对纸牌游戏的操作 首先输出队首的牌，然后队首出队，这样就完成了第一个操作； 然后将队首的牌入队，放置队尾，再次令队首出队，这样纸牌游戏就完成了； 游戏结束判定条件，队列成员 queuesize == 0 时，结束游戏 3. 测试结果（测试输入，测试输出，结果分析） 测试一： 输入： 10 输出： <pre> Input the number of playing cards 10 Number: 10 1 2 3 4 5 6 7 8 9 10 No.1 Operation: Remove '1' Put '2' on the following No.2 Operation: Remove '3' Put '4' on the following No.3 Operation: Remove '5' Put '6' on the following No.4 Operation: Remove '7' Put '8' on the following No.5 Operation: Remove '9' Put '10' on the following No.6 Operation: Remove '2' Put '4' on the following No.7 Operation: Remove '6' Put '8' on the following No.8 Operation: Remove '10' Put '4' on the following No.9 Operation: Remove '8' Put '4' on the following The Last is '4' </pre> Input the number of playing cards 10 Number: 10		

1 2 3 4 5 6 7 8 9 10

No.1 Operation: Remove '1' Put '2' on the following
No.2 Operation: Remove '3' Put '4' on the following
No.3 Operation: Remove '5' Put '6' on the following
No.4 Operation: Remove '7' Put '8' on the following
No.5 Operation: Remove '9' Put '10' on the following
No.6 Operation: Remove '2' Put '4' on the following
No.7 Operation: Remove '6' Put '8' on the following
No.8 Operation: Remove '10' Put '4' on the following
No.9 Operation: Remove '8' Put '4' on the following
The Last is '4'

测试二:

输入:

100

输出:

No.90 Operation: Remove '32' Put '40' on the following
No.91 Operation: Remove '48' Put '56' on the following
No.92 Operation: Remove '64' Put '72' on the following
No.93 Operation: Remove '80' Put '88' on the following
No.94 Operation: Remove '96' Put '8' on the following
No.95 Operation: Remove '24' Put '40' on the following
No.96 Operation: Remove '56' Put '72' on the following
No.97 Operation: Remove '88' Put '8' on the following
No.98 Operation: Remove '40' Put '72' on the following
No.99 Operation: Remove '8' Put '72' on the following
The Last is '72'

Input the number of playing cards

100

Number: 100

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49
50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72
73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95
96 97 98 99 100

No.1 Operation: Remove '1' Put '2' on the following
No.2 Operation: Remove '3' Put '4' on the following
No.3 Operation: Remove '5' Put '6' on the following
No.4 Operation: Remove '7' Put '8' on the following
No.5 Operation: Remove '9' Put '10' on the following
No.6 Operation: Remove '11' Put '12' on the following
No.7 Operation: Remove '13' Put '14' on the following
No.8 Operation: Remove '15' Put '16' on the following
No.9 Operation: Remove '17' Put '18' on the following
No.10 Operation: Remove '19' Put '20' on the following
No.11 Operation: Remove '21' Put '22' on the following
No.12 Operation: Remove '23' Put '24' on the following
No.13 Operation: Remove '25' Put '26' on the following
No.14 Operation: Remove '27' Put '28' on the following

No. 15	Operation: Remove '29'	Put '30' on the following
No. 16	Operation: Remove '31'	Put '32' on the following
No. 17	Operation: Remove '33'	Put '34' on the following
No. 18	Operation: Remove '35'	Put '36' on the following
No. 19	Operation: Remove '37'	Put '38' on the following
No. 20	Operation: Remove '39'	Put '40' on the following
No. 21	Operation: Remove '41'	Put '42' on the following
No. 22	Operation: Remove '43'	Put '44' on the following
No. 23	Operation: Remove '45'	Put '46' on the following
No. 24	Operation: Remove '47'	Put '48' on the following
No. 25	Operation: Remove '49'	Put '50' on the following
No. 26	Operation: Remove '51'	Put '52' on the following
No. 27	Operation: Remove '53'	Put '54' on the following
No. 28	Operation: Remove '55'	Put '56' on the following
No. 29	Operation: Remove '57'	Put '58' on the following
No. 30	Operation: Remove '59'	Put '60' on the following
No. 31	Operation: Remove '61'	Put '62' on the following
No. 32	Operation: Remove '63'	Put '64' on the following
No. 33	Operation: Remove '65'	Put '66' on the following
No. 34	Operation: Remove '67'	Put '68' on the following
No. 35	Operation: Remove '69'	Put '70' on the following
No. 36	Operation: Remove '71'	Put '72' on the following
No. 37	Operation: Remove '73'	Put '74' on the following
No. 38	Operation: Remove '75'	Put '76' on the following
No. 39	Operation: Remove '77'	Put '78' on the following
No. 40	Operation: Remove '79'	Put '80' on the following
No. 41	Operation: Remove '81'	Put '82' on the following
No. 42	Operation: Remove '83'	Put '84' on the following
No. 43	Operation: Remove '85'	Put '86' on the following
No. 44	Operation: Remove '87'	Put '88' on the following
No. 45	Operation: Remove '89'	Put '90' on the following
No. 46	Operation: Remove '91'	Put '92' on the following
No. 47	Operation: Remove '93'	Put '94' on the following
No. 48	Operation: Remove '95'	Put '96' on the following
No. 49	Operation: Remove '97'	Put '98' on the following
No. 50	Operation: Remove '99'	Put '100' on the following
No. 51	Operation: Remove '2'	Put '4' on the following
No. 52	Operation: Remove '6'	Put '8' on the following
No. 53	Operation: Remove '10'	Put '12' on the following
No. 54	Operation: Remove '14'	Put '16' on the following
No. 55	Operation: Remove '18'	Put '20' on the following
No. 56	Operation: Remove '22'	Put '24' on the following
No. 57	Operation: Remove '26'	Put '28' on the following
No. 58	Operation: Remove '30'	Put '32' on the following

No.59 Operation: Remove '34' Put '36' on the following
 No.60 Operation: Remove '38' Put '40' on the following
 No.61 Operation: Remove '42' Put '44' on the following
 No.62 Operation: Remove '46' Put '48' on the following
 No.63 Operation: Remove '50' Put '52' on the following
 No.64 Operation: Remove '54' Put '56' on the following
 No.65 Operation: Remove '58' Put '60' on the following
 No.66 Operation: Remove '62' Put '64' on the following
 No.67 Operation: Remove '66' Put '68' on the following
 No.68 Operation: Remove '70' Put '72' on the following
 No.69 Operation: Remove '74' Put '76' on the following
 No.70 Operation: Remove '78' Put '80' on the following
 No.71 Operation: Remove '82' Put '84' on the following
 No.72 Operation: Remove '86' Put '88' on the following
 No.73 Operation: Remove '90' Put '92' on the following
 No.74 Operation: Remove '94' Put '96' on the following
 No.75 Operation: Remove '98' Put '100' on the following
 No.76 Operation: Remove '4' Put '8' on the following
 No.77 Operation: Remove '12' Put '16' on the following
 No.78 Operation: Remove '20' Put '24' on the following
 No.79 Operation: Remove '28' Put '32' on the following
 No.80 Operation: Remove '36' Put '40' on the following
 No.81 Operation: Remove '44' Put '48' on the following
 No.82 Operation: Remove '52' Put '56' on the following
 No.83 Operation: Remove '60' Put '64' on the following
 No.84 Operation: Remove '68' Put '72' on the following
 No.85 Operation: Remove '76' Put '80' on the following
 No.86 Operation: Remove '84' Put '88' on the following
 No.87 Operation: Remove '92' Put '96' on the following
 No.88 Operation: Remove '100' Put '8' on the following
 No.89 Operation: Remove '16' Put '24' on the following
 No.90 Operation: Remove '32' Put '40' on the following
 No.91 Operation: Remove '48' Put '56' on the following
 No.92 Operation: Remove '64' Put '72' on the following
 No.93 Operation: Remove '80' Put '88' on the following
 No.94 Operation: Remove '96' Put '8' on the following
 No.95 Operation: Remove '24' Put '40' on the following
 No.96 Operation: Remove '56' Put '72' on the following
 No.97 Operation: Remove '88' Put '8' on the following
 No.98 Operation: Remove '40' Put '72' on the following
 No.99 Operation: Remove '8' Put '72' on the following
 The Last is '72'

4. 分析与探讨（结果分析，若存在问题，探讨解决问题的途径）
 结果分析：

数据正确。

5. 附录：实现源代码（本实验的全部源程序代码，程序风格清晰易理解，有充分的注释）

```
#include <iostream>
#include <string>
using namespace std;

/*节点的结构体定义*/
template <class T>
struct chainNode
{
    T element;
    chainNode * next;

    chainNode() {}
    chainNode(const T& theElement)
    {
        this->element = theElement;
    }
    chainNode(const T& theElement, chainNode<T>* theNext)
    {
        this->element = theElement;
        this->next = theNext;
    }
};

template <class T>
class chainQueue
{
private:
    int queuesize; //队列长度
    chainNode<T> *queueFront; //队列的头指针
    chainNode<T> *queueBack; //队列的尾指针
public:
    chainQueue() //构造函数
    {
        queuesize = 0;
        queueFront = NULL;
        queueBack = NULL;
    }
    ~chainQueue() //析构函数
    {
        while(queueFront != NULL)
        {
```

```

        chainNode<T> *NextNode = queueFront->next;
        delete queueFront;
        queueFront = NextNode;
    }
}

bool Empty() {return queuesize == 0;}
int Size() {return queuesize;}
T& Front()    //返回队列的首位元素
{
    if (Empty())
    {
        cout << "The queue is empty" << endl;
        string error = "The queue is empty";
        throw error;
    } //为空抛出异常
    return queueFront->element;
}

T& Back()    //返回队列的末尾元素
{
    if (Empty())
    {
        cout << "The queue is empty" << endl;
        string error = "The queue is empty";
        throw error;
    } //为空抛出异常
    return queueBack->element;
}

void Pop();    //删除
void Push(const T&);    //插入
void Create(int Number); //创建 Number 个节点的链表
void Output();    //输出链表
void Play();    //进行纸牌游戏
};

```

/*删除*/

```

template <class T>
void chainQueue<T>::Pop()
{
    if (Empty())
    {
        cout << "The queue is empty" << endl;
        string error = "The queue is empty";
        throw error;
    } //为空抛出异常
}

```

```

        chainNode<T> *NextNode = queueFront->next;
        delete queueFront;
        queueFront = NextNode;
        queuesize--;
    }

```

/*插入*/

```

template <class T>
void chainQueue<T>::Push(const T& theElement)
{
    chainNode<T> *New = new chainNode<T>(theElement, NULL);
    if (queuesize == 0)
        queueFront = New;
    else
        queueBack->next = New;
    queueBack = New;
    queuesize++;
}

```

/*创建 Number 个节点的链表(表示扑克牌)*/

```

template <class T>
void chainQueue<T>::Create(int Number)
{
    int n=Number;
    for (int i=0; i<n; i++)    //从头至尾分别是 1, 2, 3……n
        Push(i+1);
}

```

/*输出链表*/

```

template <class T>
void chainQueue<T>::Output()
{
    chainNode<T> *temp;
    temp = queueFront;
    for (int i=0; i<queuesize; i++)
    {
        cout << temp->element << " ";
        temp = temp->next;
    }
    cout << endl;
    delete temp;
}

```

/*进行纸牌游戏*/

```

template <class T>
void chainQueue<T>::Play()
{
    int i = 0;
    T lastElement;
    while (queuesize>1)
    {
        cout << "No." << i+1 << " Operation: ";
        cout << "Remove ' " << queueFront->element << " " << endl;
        Pop(); //移除顶部的一张
        cout << "Put ' " << queueFront->element << " " << " on the
following" << endl;
        lastElement = queueFront->element;
        Push(queueFront->element); //将移除之后的顶部的拍放到底部
        Pop();
        i++;
    }
    cout << "The Last is ' " << lastElement << " " << endl;
}

int main()
{
    try
    {
        while(1)
        {
            int number; //纸牌数量
            cout << "Input the number of playing cards" << endl;
            cin >> number;
            chainQueue<int> Playing_card;
            Playing_card.Create(number);
            cout << "Number: " << Playing_card.Size() << endl;
            Playing_card.Output();
            Playing_card.Play();
        }
    }
    catch (string error)
    {
        cout << error << endl;
        return 0;
    }
    return 0;
}

```