

提交方式:

Canvas 上提交

提交内容:

一个压缩包，命名方式：姓拼音\_名拼音\_学号.zip/rar，如：zhou\_zhiming\_2020000112.zip（不要用中文）  
内含若干个文件夹，每个文件夹对应一道题的项目（仅保留.pro 和其他源文件；文件夹名也不要有中文）  
(.cpp / .h 等)

评分标准:

代码正确性和完整性 90%

代码风格 10% （代码看着很乱的酌情扣 0-10 分。唯一的要求：代码需按层次缩进、对齐）

逾期惩罚:

逾期提交的，成绩  $\times= 0.8$  （无论逾期多久）

以下题目均要求：基于 Stack 和 Queue 完成，不能使用 stack 和 queue（如有必要可以用 Vector）

1. 括号匹配（参看：<https://www.bilibili.com/video/BVluxCzYrEvK/>）

(1) console 输入：

① 一个字符串，代表一个 C++ 代码文件的文件名

(2) console 输出：

① 代码文件中任意一个不匹配的括号所在的行号和列号；如都匹配，则输出 -1

2. 中缀表达式转后缀表达式并求值（参看：<https://www.bilibili.com/video/BV1gL6UYbExu/>）

(1) console 输入：

① 一个字符串，代表一个中缀表达式。

② 可以假设表达式中仅包含：( )，二元运算符 + - \* /，以及个位数的整数

③ 感兴趣的同学可以尝试拓展，使支持多位整数/浮点数，以及支持更多的运算符，如指数<sup>^</sup>、负号

(2) console 输出：

① 第一行：输入的中缀表达式对应的后缀表达式

② 第二行：表达式的值

3. 实现一个基于两个栈的队列类（参看：<https://www.bilibili.com/video/BV1rG411L7XA/>）

(1) 要求支持：enqueue、dequeue、peek、isEmpty、size、clear

(2) 可以仅支持一个数据类型，如 int（感兴趣的同学也可以用 template 支持任意数据类型）

(3) 可以直接写在 main.cpp 中，不要求以 myQueue.h+myQueue.cpp 的形式

(4) 可参考下面给出的代码框架

```
#include "console.h"
#include "stack.h"
using namespace std;

class MyQueue
{
private:
    Stack<int> in, out;
public:

    void enqueue(int x){
        // TODO
    }

    int dequeue() {
        //TODO
    }

    // TODO: 补充其他需要的函数
};

int main()
{
```

```

MyQueue que;

que.enqueue(1);
cout << que.dequeue();

// TODO: 补充更多测试
}

```

#### 4. 栈和队列的逆序与旋转

(1) 实现函数 `void reverse(Stack<char>& x, int n);`

① 将栈 `x` 顶部的 `n` 个元素的顺序逆转，如：`S{1, 2, 3, 4, 5}`，`3`  $\rightarrow$  `S{1, 2, 5, 4, 3}`

(2) 实现函数 `void reverse(Queue<char>&x, int n);`

① 将队列 `x` 头部的 `n` 个元素的顺序逆转，如：`Q{1, 2, 3, 4, 5}`，`3`  $\rightarrow$  `Q{3, 2, 1, 4, 5}`

(3) 实现函数 `void rotate(Stack<char> &x, int n, int k);`

① 将栈 `x` 顶部的 `n` 个元素，旋转 `k` 次

1) 每次旋转的行为定义：将栈顶元素从栈顶移除，然后插入到自顶向下数的第 `n` 个元素的位置

a. 如：`S{1, 2, 3, 4, 5}`，`3`，`2`  $\rightarrow$  `S{1, 2, 5, 3, 4}`  $\rightarrow$  `S{1, 2, 4, 5, 3}`

2) 实现的时候，不一定要模拟旋转 `k` 次，最终结果等价即可

(4) 实现函数 `void rotate(Queue<char> &x, int n, int k);`

① 将队列 `x` 头部的 `n` 个元素，旋转 `k` 次

1) 每次旋转的行为定义：将头部元素移出，然后插入到自头向后数的第 `n` 个元素的位置

a. 如：`Q{1, 2, 3, 4, 5}`，`3`，`2`  $\rightarrow$  `Q{2, 3, 1, 4, 5}`  $\rightarrow$  `Q{3, 1, 2, 4, 5}`

2) 实现的时候，不一定要模拟旋转 `k` 次，最终结果等价即可