一．队列（#include<queue> 定义：queqe<int>q）

栈和队列要求手打！！！！！！！！！

1.队列：

<1>队列与栈一样，是一种线性存储结构；遵循先进先出的原理（FIFO），队尾添加新元素，队头删除元素。

2.相关概念：

<1>队头与队尾：允许插入的一段为对尾，允许删除的一端为对头。

<2>入队：向队列中插入元素；出队：从队列中删除元素。

3.队列的操作：

<1>入队：push();

<2>出队：pop();

<3>求队列中元素个数；

<4>判断队列是否为空；

<5>获取队首元素。

4.队列的分类：

<1>循环队列；

<2>链队列；

5.eg.

C++队列模板定义在queue中，需要两个参数，一个元素类型，一个容器类型，

元素类型是必要的，容器类型是可选的。默认容器类型为deque。

如何判断队列是空队还是已经排满的呢？

1. 栈空：队手标志=队尾标志 表示栈空。
2. 队满：队尾+1=对手，表示队满

6.队列操作函数：

q.empty() 如果队列为空返回true，否则返回false

q.size() 返回队列中元素的个数

q.pop() 删除队列首元素但不返回其值

q.front() 返回队首元素的值，但不删除该元素

q.push(i) 在队尾压入新元素i

q.back() 返回队列尾元素的值，但不删除该元素

7.循环队列：以数组为底层数据结构，每次从队列头部删除元素（出队）后，需要将头部以后的所有元素向前移动一个单位 时间复杂度为O（n）可将数组看作是一个首尾相连的环，删除元素是队列标志向后移动，添加元素时若数组尾部没有空间，考虑数组头部空间是否为空闲，若是，则在数组头部进行插入。

定义front为队列元素头元素的位置，rear为队列尾元素的位置，MAXSIZE为循环队列的最大长度。

循环队列注意问题：a.求元素的个数：(rear-front+MAXSIZE)%MAXSIZE；

b. front/real指向下一个逻辑空间front=(front+1)%MAXSIZE,rear=(rear+1)%MAXSIZE;

c.判空：front=rear;

d.判满：（rear+1+MAXSIZE）==front;

8.循环队列中判断队列为空的方法时判断front==rear，队满的方法是判断front=(rear+1)%MAXSIZE,至于为什么不用一个变量接收队长，因为在频繁的操作中多出一个变量会大大增大执行时间，不如浪费一个数组空间。链式队列适合元素变动较大的情况。

二．栈（Stack） 头文件 #include<stack> 定义：stack<int>s

栈和队列要求手打！！！！！

1.栈是一种线性存储结构. <1>栈中元素遵循“先进后出”原则（看作一个只有大门的客栈） 简称FILO结构。<2>限定只能在栈顶进行插入删除操作。

2.栈的相关概念：

《1》栈顶与栈底：允许元素插入与删除的一端称为栈顶，另一端成为称为栈底。

《2》压栈：栈的插入操作，叫做进栈、压栈、入栈。

《3》弹栈：栈的删除操作，也叫做出栈。

3.栈的常见操作：

《1》弹栈 pop; s.pop(); //弹出栈顶元素, 但不返回其值

《2》压栈 push; s.push(i); //将元素i压入栈顶

《3》求栈大小； s.size(); //返回栈中元素的个数

《4》判断栈是否为空； s.empty(); //如果栈为空则返回true, 否则返回false;

《5》获取栈顶元素的值； s.top(); //返回栈顶元素, 但不删除该元素

4.栈的常见分类：

《1》基于数组的栈：以数组为底层数据结构，通常以数组头为栈底，数组头到数组尾为栈的生长方向。手打栈。《2》基于单链表的栈：以链表为底层数据结构，以链表头为栈顶，便于节点的插入与删除，压栈产生的新节点将一直出现在链表的头部。



三、vector动态数组：头文件#inclde<vector> 定义：vector<int||float||double||char>a;

（2）

初始化：vector<int>a(n,m) 在a里装入N个M；

vector<int >a(n)在a里装入N个0；

或者：为了节省空间，防止mlt

Vector<int>A;

Vector<int >B(A)；创建一个和A一样的动态数组B；

Vector<int >C(B.begin+l,B.end()-r);//继承B动态数组下标[I,B.end()-r]的值；

（3）析构：B.~vector<int>();

动态数组名.~vector<数组的数据结构>();

注意：vector数组的下标也是从0开始的！

（4）Vector基本操作：以Vector<int> v为例：



（5）vector函数插入：

<i>, v.push\_back(a);//在V的尾部插入数a；

<ii>. V.insert(v.begin()+k,a); 在下标k前面插入数a，k之后的数依次后退一位。（下标从0开始）；

<iii>v.insert(v.begin()+k,p,a);//在下标k之前插入p个a;

(6)vector 的删除：

<i>v.pop\_back();//删除最后一个元素；

<ii>v.erase(v.begin()+k);//删除下标为k的数，返回下一个位置的下标；

<iii>v.erase(v.begin()+I,v.end()-r);//删除下标在[I,end()-r]的元素；

（7）vector数组长度测量：v.capacity();//返回v数组的长度，相当于int a[1000];中的1000；

v.resize(k);//将数组长度设为k;

vector内存机制：当动态数组内的元素比动态数组长度多一时，动态数组长度翻倍，因此要把动态数组长度拉足够，避免长度翻倍花时间。



四、链表（结构体实现）包含数字域与指针域；

1.直接代码：

2.双向循环链表：



五、补充栈：

1．数组实现栈：



六、队列

1.队列数组实现：

2.队列的链表实现：

七、树、二叉树与森林