



**实验舱**  
青少年编程  
走近科学 走进名校

# 蛟龙四班

## STL常用容器、模拟

# vector

vector 是 STL 提供的 内存连续的、可变长度 的数组数据结构

它能够提供线性复杂度的插入和删除,以及常数复杂度的随机访问

方法	功能
push_back	在末尾加入一个元素
pop_back	在末尾弹出一个元素
size	获取长度
clear	清空

更多用法可以翻阅文档: <http://cplusplus.com/reference/vector/vector/?kw=vector>



# #300、Mas的vector

## 题目描述

Mas 对 vector 非常感兴趣,他觉得 vector 比数组强大,比数组好用

现在有  $n$  个空的数组,你现在需要读入  $n$  行 2 个数  $p, x$  表示在第  $p$  个数组后面加入数字  $x$

最后需要你顺序输出所有数组里的数

## 输入格式

第一行一个数字  $n$

然后  $n$  行,每行 2 个数  $p, x$

## 输出格式

输出  $n$  行,共输出  $n$  个数,按数组的顺序输出内容

## 数据规模

对于 25% 的数据,  $n \leq 10$

对于 50% 的数据,  $n \leq 500$

对于 75% 的数据,  $n \leq 10000$

对于 100% 的数据,  $n \leq 100000$

对于全部数据  $1 \leq p, x \leq n$



# #399、木头移动

## 题面描述

$Mas$  有  $n$  块木头,编号从  $1 \sim n$ .

一开始,第  $i$  块木头放在  $i$  的位置

现在  $Mas$  进行  $m$  次操作,每次把  $b$  位置的所有木头移动到  $a$  位置上

比如 1 位置的木头是 1, 2 位置的木头是 2,那么把位置 2 的木头移动到位置 1 后,位置 1 上的积木从下到上依次为 1, 2

## 输入格式

第一行输入 2 个整数  $n, m$ 。

接下来  $m$  行,每行输入 2 个整数  $a, b$ ,如果  $a = b$  则本次不需要移动。

## 输出格式

输出  $n$  行

第  $i$  行输出位置  $i$  从下到上的木头编号,如果该行没有木头输出一行空行

## 数据规模

对于全部的数据  $1 \leq n \leq 10000, 0 \leq m \leq 10000, 1 \leq a, b \leq n$

# set

set 是关联容器,含有键值类型对象的已排序集,搜索、移除和插入拥有对数复杂度

和数学中的集合相似,set 中不会出现值相同的元素

如果有相同元素的集合,需要使用 multiset

multiset 的使用方法与 set 的使用方法基本相同

方法	功能
insert	插入一个元素
erase	删除一个元素
count	判断元素是否在set中
size	获取元素个数

更多用法可以翻阅文档: <http://cplusplus.com/reference/set/set/?kw=set>



# #398、集合合并

## 题目描述

给你两个集合,计算其并集,即  $A \cup B$

|  $A \cup B$  中不允许出现重复元素,但是  $A$  与  $B$  之间可能存在相同元素

## 输入格式

输入数据分为三行

第一行有两个数字  $n, m$ , 分别表示集合  $A$  和集合  $B$  的元素个数

后两行分别表示集合  $A$  和集合  $B$

每个元素为不超出 `int` 范围的整数,每个元素之间用一个空格隔开

## 输出格式

输出一行数据,表示合并后的集合

要求从小到大输出,每个元素之间用一个空格隔开

## 样例输入1

```
1 2
3
1 2
```

## 样例输出1

```
1 2 3
```

## 数据规模

对于全部的数据  $0 \leq n, m \leq 10000$



# #299、Mas的数字

## 题目描述

*Mas* 对数字非常感兴趣,他最近在研究一种数叫 *Mas* 数  
这种数不能被  $2, 3, 5$  之外的素数整除,只能被这  $3$  个素数整除

显然前几个 *Mas* 数为,  $2, 3, 4, 5, 6, 8, \dots$

现在需要你输出第  $n$  个 *Mas* 数

## 输入格式

第一行一个数字  $n$

## 输出格式

输出第  $n$  个 *Mas* 数

## 数据规模

对于  $25\%$  的数据,  $n \leq 10$

对于  $50\%$  的数据,  $n \leq 100$

对于  $75\%$  的数据,  $n \leq 1000$

对于  $100\%$  的数据,  $n \leq 10000$

将  $2, 3, 5$  加入 set

每次从set中取出最小值 $x$ ,并将  $2x, 3x, 5x$ 加入 set

第  $k$  次取出的元素即为 答案

时间复杂度  $O(k \log k)$

如何证明该做法的正确性?

# map

map 是有序键值对容器,它的元素的键是唯一的

搜索、移除和插入操作拥有对数复杂度 (map 通常实现为红黑树)

map 重载了 `operator[]` ,可以用任意定义了 `operator<` 的类型作为下标(在 map 中叫做 key,也就是索引)

方法	功能
insert	插入一对映射
count	查找关键字
erase	删除关键字
size	获取映射对个数

更多用法可以翻阅文档: <http://cplusplus.com/reference/map/map/?kw=map>





# #447 DD老师的通话记录

## 题面描述

DD 老师最近拨打了很多的电话,她想要统计一下每个电话拨打了多少次,请你帮帮她

## 输入格式

第一行一个整数  $n$

接下来  $n$  行,每行一个整数代表拨打的电话号码

DD 老师可能会向全宇宙的人拨打电话,所以电话号码的范围( $-2^{63} \leq p \leq 2^{63} - 1$ )

## 输出格式

按照号码从小到大的顺序(非字典序)输出号码和出现次数

每行一个号码及次数



# #1678、实验舱的报表

## 题目描述

*SYC* 的蛟龙计划有很多面向全国的课程

有一天 *Sifu Zhou* 想要知道每个省份的学员报名情况,想让 *Mas* 整理一份报表

## 输入格式

第一行是一个整数  $n$ ,表示共有  $n$  次成功的报名

其后有  $n$  行数据,每行表示一次报名情况,由课程代号(小写字母和数字组成,长度不超过 20),学员所在省份(小写字母组成,长度不超过 20)和报名人数(正整数,不超过 1000)组成.

## 输出格式

请你输出一份排版格式正确(请分析样本输出)的报名情况明细表

这份明细表包括所有已报名的省份、课程和数目的信息

先按省份分类,按字母顺序排列

同一省份的课程按照按字母顺序排序

## 样例输入

```
5
st01 jiangsu 3
st02 zhejiang 1
ct03 guangdong 1
st02 guangdong 3
st02 guangdong 1
```

## 样例输出

```
guangdong
|----ct03(1)
|----st02(4)
jiangsu
|----st01(3)
zhejiang
|----st02(1)
```

## 数据规模

对于全部的数据  $0 < N \leq 1000$

模拟就是用计算机来模拟题目中要求的操作

模拟题目通常具有码量大、操作多、思路繁复的特点

由于它码量大,经常会出现难以查错的情况,如果在考试中写错是相当浪费时间

写模拟题时,遵循以下的建议有可能会提升做题速度:

- 在动手写代码之前,在草纸上尽可能地写好要实现的流程
- 在代码中,尽量把每个部分模块化,写成函数、结构体或类
- 对于一些可能重复用到的概念,可以统一转化方便处理
- 调试时分块调试
- 写代码的时候一定要思路清晰,切勿想到什么写什么



# #1016 接水问题

## 题目描述

学校里有一个水房,水房里一共装有  $m$  个龙头可供同学们打开水,每个龙头每秒钟的供水量相等,均为 1

现在有  $n$  名同学准备接水,他们的初始接水顺序已经确定

将这些同学按接水顺序从 1 到  $n$  编号,  $i$  号同学的接水量为  $w_i$

接水开始时, 1 到  $m$  号同学各占一个水龙头,并同时打开水龙头接水

当其中某名同学  $j$  完成其接水量要求  $w_j$  后,下一名排队等候接水的同学  $k$  马上接替  $j$  同学的位置开始接水

这个换人的过程是瞬间完成的,且没有任何水的浪费

即  $j$  同学第  $x$  秒结束时完成接水,则  $k$  同学第  $x + 1$  秒立刻开始接水

若当前接水人数  $n'$  不足  $m$ ,则只有  $n'$  个龙头供水,其它  $m - n'$  个龙头关闭

现在给出  $n$  名同学的接水量,按照上述接水规则,问所有同学都接完水需要多少秒

## 输入格式

每组输入数据的第 1 行是 2 个整数  $n$  和  $m$ ,用一个空格隔开,分别表示接水人数和龙头个数

第 2 行  $n$  个整数  $w_1, w_2, \dots, w_n$ ,每两个整数之间用一个空格隔开,  $w_i$  表示  $i$  号同学的接水量

## 输出格式

每组输出只有一行,为 1 个整数,表示接水所需的总时间

## 样例输入

```
5 3
4 4 1 2 1
8 4
23 71 87 32 70 93 80 76
```

## 样例输出

```
4
163
```

## 数据规模

对于全部的数据  $1 \leq n \leq 10000, 1 \leq m \leq 100, m \leq n, 1 \leq w_i \leq 100$



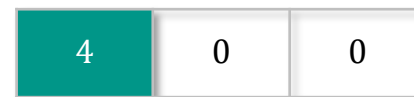
# #1016 接水问题

## 思路1

按秒模拟

每一秒将  $w$  中的前  $m$  个非零的元素减一,直到  $w$  内没有非 0 元素

可将  $w$  中为 0 的元素移除以减少内层循环遍历次数



## 思路2

记  $T_i$  为第  $i$  个水龙头的累计接水时间

队伍中第  $i$  个人需要接水时

必然选择  $T_i$  最小的水龙头

$\max\{T_i\}$ 即为答案



# #1608、新建Microsoft Word文档

## 问题描述

当你新建一个 *word* 文档时,会得到一个名为 新建 Microsoft Word 文档.doc 的文件

再新建一个则名为 新建 Microsoft Word 文档(2).doc

再新建便是 新建 Microsoft Word 文档(3).doc

不断新建,编号不断递增

倘若你现在新建了三个文档,然后删除了 新建 Microsoft Word 文档(2).doc ,再新建就又会得到一个 新建 Microsoft Word 文档(2).doc

严格说, *Windows* 在每次新建文档时,都会选取一个与已有文件编号不重复的最小正整数作为新文档的编号

请编程模拟以上过程,支持以下两种操作:

- New : 新建一个 *word* 文档,反馈新建的文档的编号
- Delete id : 删除一个编号为 *id* 的 *word* 文档,反馈删除是否成功

初始时一个文件都没有, 新建 Microsoft Word 文档.doc 的编号算作 1

## 数据规模和约定

对于 50% 的数据,操作数  $\leq 1000$  ,最大 *id*  $\leq 1000$

对于 100% 的数据,操作数  $\leq 100000$  ,最大 *id*  $\leq 100000$

## 输入格式

第一行一个正整数 *n* 表示操作次数,接下来 *n* 行,每行表示一个操作

若该行为 New ,则表示新建

若为 Delete id 则表示要删除编号为 *id* 的文档,其中 *id* 为一个正整数

操作按输入顺序依次进行

## 输出格式

对于输入的每一行,输出其反馈结果

对于新建操作,输出新建的文档的编号

对于删除操作,反馈删除是否成功:

如果删除的文件存在,则删除成功,输出 Successful

否则输出 Failed

# #1608、新建Microsoft Word文档

对于每次新建操作

若从前往后查找第一个未被占用的编号,时间复杂度 $O(nm)$

考虑 *set* 维护所有未被占用的编号

每次新建从 *set* 中取出最小编号

每次删除将编号加入 *set*

时间复杂度  $O(m\log n)$



# #2645、网络链接

## 题目描述

*TCP/IP* 协议是网络通信领域的一项重要协议。今天你的任务,就是尝试利用这个协议,还原一个简化后的网络连接场景。在本问题中,计算机分为两大类:服务机 ( `Server` ) 和客户机 ( `Client` )。服务机负责建立连接,客户机负责加入连接

需要进行网络连接的计算机共有  $n$  台,编号为  $1 \sim n$ ,这些机器将按编号递增的顺序,依次发起一条建立连接或加入连接的操作。每台机器在尝试建立或加入连接时需要提供一个地址串。服务机提供的地址串表示它尝试建立连接的地址,客户机提供的地址串表示它尝试加入连接的地址。

一个符合规范的地址串应当具有以下特征:

- 必须形如 `a.b.c.d:e` 的格式,其中  $a, b, c, d, e$  均为非负整数。
- $0 \leq a, b, c, d \leq 255, 0 \leq e \leq 65535$ 。
- $a, b, c, d, e$  均不能含有多余的前导  $0$ 。

相应地,不符合规范的地址串可能具有以下特征:

- 不是形如 `a.b.c.d:e` 格式的字符串,例如含有多于  $3$  个字符 `.` 或多于  $1$  个字符 `:` 等情况
- 整数  $a, b, c, d, e$  中某一个或多个超出上述范围
- 整数  $a, b, c, d, e$  中某一个或多个含有多余的前导  $0$

例如,地址串 `192.168.0.255:80` 是符合规范的,但

```
192.168.0.999:80
192.168.00.1:10
192.168.0.1:088
192:168:0:1.233
```

均是不符合规范的

如果服务机或客户机在发起操作时提供的地址串不符合规范,这条操作将被直接忽略

在本问题中,我们假定凡是符合上述规范的地址串均可参与正常的连接,你无需考虑每个地址串的实际意义

由于网络阻塞等原因,不允许两台服务机使用相同的地址串,如果此类现象发生,后一台尝试建立连接的服务机将会无法成功建立连接;除此之外,凡是提供符合规范的地址串的服务机均可成功建立连接

如果某台提供符合规范的地址的客户机在尝试加入连接时,与先前某台已经成功建立连接的服务机提供的地址串相同,这台客户机就可以成功加入连接,并称其连接到这台服务机;如果找不到这样的服务机,则认为这台客户机无法成功加入连接

请注意,尽管不允许两台不同的服务机使用相同的地址串,但多台客户机使用同样的地址串,以及同一台服务机同时被多台客户机连接的情况是被允许的

你的任务很简单:在给出每台计算机的类型以及地址串之后,判断这台计算机的连接情况。

## 输入样例 #1

```
5
Server 192.168.1.1:8080
Server 192.168.1.1:8080
Client 192.168.1.1:8080
Client 192.168.1.1:80
Client 192.168.1.1:99999
```

## 输出样例 #1

```
OK
FAIL
1
FAIL
ERR
```

## 数据范围

对于  $100\%$  的数据,保证  $1 \leq n \leq 1000$



# #2645、网络链接

对于每个地址需要检验其是否合法

可使用map标记所有的合法服务端地址

可借助sscanf,sprintf简化逻辑

方法	功能
int sscanf(char *str,char *format,...)	以str的数据作为输入以format中的格式分别写入变量列表中 若写入失败返回EOF
int sprintf(char *str,char *format,...)	将变量列表中的值按照format中的格式写入str中 若写入失败返回EOF

```
bool validate()
{
    if (sscanf(ip, "%d.%d.%d.%d:%d", a, a + 1, a + 2, a + 3, a + 4) != 5)
        return false; //判断格式是否非法
    sprintf(buf, "%d.%d.%d.%d:%d", a[0], a[1], a[2], a[3], a[4]);
    if (strcmp(ip, buf))
        return false; //比较是否存在前导零
    for (int i = 0; i < 4; i++)
        if (a[i] < 0 || a[i] > 255) //ip各个段是否合法
            return false;
    if (a[4] < 0 || a[4] > 65536) //端口是否非法
        return false;
    return true;
}
```



# #1628、公交换乘

## 题目描述

著名旅游城市  $B$  市为了鼓励大家采用公共交通方式出行,推出了一种地铁换乘公交车的优惠方案:

1. 在搭乘一次地铁后可以获得一张优惠票,有效期为 45 分钟,在有效期内可以消耗这张优惠票,免费搭乘一次票价不超过地铁票价的公交车。在有效期内指开始乘公交车的时间与开始乘地铁的时间之差小于等于 45 分钟,即:

$$T_{bus} - T_{subway} \leq 45$$

2. 搭乘地铁获得的优惠票可以累积,即可以连续搭乘若干次地铁后再连续使用优惠票搭乘公交车,但每次搭乘公交车只能使用一张优惠券
3. 搭乘公交车时,如果可以使用优惠票一定会使用优惠票;如果有多张优惠票满足条件,则优先消耗获得最早的优惠票

现在你得到了小轩最近的公共交通出行记录,你能帮他算算他的花费吗?

## 输入格式

第一行包含一个正整数  $n$ ,代表乘车记录的数量

接下来的  $n$  行,每行包含 3 个整数,相邻两数之间以一个空格分隔

第  $i$  行的第 1 个整数代表第  $i$  条记录乘坐的交通工具,0 代表地铁,1 代表公交车

第 2 个整数代表第  $i$  条记录乘车的票价  $price_i$

第三个整数代表第  $i$  条记录开始乘车的时间  $t_i$  (距 0 时刻的分钟数)

我们保证出行记录是按照开始乘车的时间顺序给出的,且 不会有两次乘车记录出现在同一分钟

## 输出格式

有一行,包含一个正整数,代表小轩出行的总花费

## 数据范围

对于 30% 的数据,  $n \leq 1000, t_i \leq 10^6$

另有 15% 的数据,  $t_i \leq 10^7, price_i$  都相等

另有 15% 的数据,  $t_i \leq 10^9, price_i$  都相等

对于 100% 的数据,  $n \leq 2 \times 10^5, t_i \leq 10^9, 1 \leq price_i \leq 1000$



## #1628、公交换乘

使用 vector 维护所有历史上的 地铁乘车信息

对于 每一次 公交乘车信息,在 vector 查找 可用的地铁信息抵扣

时间复杂度  $O(n^2)$

容易想到

已使用过的地铁记录不再能够抵扣,对于时间相差超过 45min 的地铁记录在之后也不再能抵扣

令 *pre* 指向 最早的地铁乘车记录

每次查找前移动 *pre* 至合法的位置

从 *pre* 往后查找时间合法且未使用的地铁记录

时间复杂度  $O(n)$



实验舱  
青少年编程  
走近科学 走进名校

谢谢观看