

# 2022 年 CFS 入门组模拟训练

(请选手务必仔细阅读本页内容)

## 一. 题目概况

中文题目名称	四则混合运算	回扣	花不在多	密码锁
英文题目名称	operation	cicer	eliminate.cpp	lock
可执行文件名	operation	cicer	eliminate	lock
输入文件名	operation.in	cicer.in	eliminate.in	lock.in
输出文件名	operation.out	cicer.out	eliminate.out	lock.out
每个测试点时限	1 秒	1 秒	1 秒	1 秒
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10
附加样例文件	无	无	无	
题目类型	传统	传统	传统	传统

## 二. 提交源程序文件名

对于 C++语言	operation.cpp	cicer.cpp	eliminate.cpp	lock.cpp
----------	---------------	-----------	---------------	----------

## 三. 编译命令 (不包含任何优化开关)

对于 C++语言	<code>g++ -o operation.cpp -lm -std=c++11</code>	<code>g++ -o cicer.cpp -lm -std=c++11</code>	<code>g++ -o eliminate.cpp -lm -std=c++11</code>	<code>g++ -o lock.cpp -lm -std=c++11</code>
----------	--	--	--	---

## 四. 运行内存限制

内存上限	512M	512M	512M	512M
------	------	------	------	------

## 五. 注意事项

- 1、文件名 (程序名和输入输出文件名) 必须使用小写。
- 2、C/C++中函数 `main()` 的返回值类型必须是 `int`, 程序正常结束时的返回值必须是 0。
- 3、时间限制有可能会因评测机性能而改变。
- 4、注意打开 **c++11**。

# 四则混合运算（operation）

## 【题目描述】

小 W 最近刚刚学习了四则混合运算运算，四则运算有加法、减法、乘法和除法四种运算。四则混合运算是小学数学的重要内容，也是学习编程的一个基础。

对于一个四则运算组成的一个表达式，在没有括号的情况下，要遵循先乘除，后加减的法则进行运算。乘除法的优先级高于加减法，并且当运算级别相同时（乘和除相同，加和相同），则按照从左到右的顺序进行。

现在给定一个混合表达式，请你帮助小 w 给这个表达式的每一个运算符标出它在整个表达式求值计算时的时间戳，即表示该运算符在运算的过程中是第几个被计算的运算符。时间戳从 1 开始，并且用一个括号括起来跟在该运算符的后面。

例如给定表达式  $2+2+2*2/2-2*2$ ，该表达式中有 6 个运算符，将它们打上时间戳后表达式就是  $2+(4)2+(5)2*(1)2/(2)2-(6)2*(3)2$ ，第一个加号后面紧跟的 (4) 表示第一个加号在整个表达式中是第 4 个被计算的符号。

## 【输入格式】

第一行一个正整数 N，表示输入的四则混合运算表达式字符串的长度。

第二行一个长度为 N 的字符串，字符串仅包括正整数和四则运算符+\* /

## 【输出格式】

一行，对混合运算表达式的每个运算符加上时间戳后的输出式子。

## 【样例输入 1】

13

2+2+2\*2/2-2\*2

## 【样例输出 1】

2+(4)2+(5)2\*(1)2/(2)2-(6)2\*(3)2

## 【样例输入 2】

5

12345

## 【样例输出 2】

12345

## 【数据范围】

对于 30% 的数据，保证  $1 \leq N \leq 10^3$ 。

对于另 10% 的数据，保证运算符中只有+-两种符合。

对于 100% 的数据，保证  $1 \leq N \leq 10^6$ 。保证字符串是一个合法的混合运算表达式，即不会出现连续两个符号，且保证符号的左右侧都为正整数。

# 回扣 (cicer)

## 【问题描述】

小 C 在暑假参加了一个旅行团，他在旅游的过程中发现，黑导游不时的会将他们带到一些商店去购物。而游客每购买一件商品，导游都可以从中取得一定数额的回扣。因此导游会竭力推荐游客购买这些商品。小 C 发现，经过导游的“洗脑”，基本上所有的游客都会按照导游的意图购买这些商品，但他们购买的商品的总价格一定不会超过他们的预算。现在小 C 已经知道了所有游客的预算和每种商品的价格以及回扣金额，小 C 想知道，这个导游在这次旅行过程中最多能得到多少回扣。

## 【输入格式】

输入数据共若干行

第一行包括一个正整数  $n$  ( $n \leq 40$ )，表示有购买预算的游客的数量；

第二行包括  $n$  个用空格隔开的正整数，表示每个游客的的预算（不超过 2000）；

第三行是一个正整数  $m$  ( $m \leq 100$ )，表示商品的数量；

接下来  $m$  行，每行两个正整数  $p$  ( $p \leq 1000$ ) 和  $q$  ( $q \leq 0.2p$ )，分别表示每种商品的价格和回扣金额。

## 【输出格式】

输出数据仅一个正整数，表示导游能获得的最大回扣金额。

## 【样例输入】

```
3
200 300 300
5
12 2
20 4
30 5
50 5
100 18
```

## 【样例输出】

```
160
```

## 【数据范围】

见题面

# 花不在多 (eliminate)

## 【问题背景】

高家车匠店夹在两家之间。门面很小，只有一间。地势却颇高。跨进门槛，得上五层台阶。因为车匠店有点像个戏台（高车匠就好像在台上演戏）。店里正对面是一堵板壁。板壁上有一副一尺多长，四寸来宽的小小的朱红对子，写的是：

室雅何须大  
花香不在多

## 【题目描述】

因为花不要太多，所以我们要把花给消掉。

现在小 W 发现，他的面前有  $n$  朵排成一列的各种颜色的花，共有  $C$  种颜色，分别是  $1, 2, 3, \dots, C$ 。他开始困惑自己应该做什么才能把这些花变少一点。

正当他疯狂挠头，束手无策之际，他突然发现有些花神秘地消失了。经过长时间的观察，他终于发现了其中的奥秘：一旦有大于等于  $m$  朵同样颜色的花靠在了一起并且其中有至少一份火元素，这  $m$  朵花就会一起被引燃并消失，同时他们两边的花会自动撞到一起，那两朵直接相撞的花会各获得半点火元素。如果被消掉的花在序列的最左边或者最右边，那么就不会产生撞击，也就不会有新的火元素产生。（可以参考样例理解）

为了把这些花消掉，小 W 会在这些花中间任意位置插入一朵带有恰好一份火元素的花。

现在小 W 开始好奇自己面前最终会剩下哪些花，因为他要去卡常，所以这个问题就交给你了。

## 【输入格式】

第一行给出两个正整数  $n, m$  和  $x, y$ 。  $n, m$  的含义如题面描述中所讲。整数  $x, y$ ，表示小 W 计划在第  $x$  朵花的后面加入一朵带有一个单位火元素的颜色为  $y$  的花。特别地，如果  $x=0$ ，则表示小 W 计划在序列的开头放置这朵花。

第二行  $n$  个相互之间用一个空格隔开的正整数，表示这  $n$  朵花分别是什么颜色（他们都不带有火元素）。

## 【输出格式】

一行  $n$  个用空格隔开的整数，表示最终花朵序列长什么样子。

如果花朵全都被消完了，请输出 "pcftxdy"（不包含引号）。。

## 【样例输入 1】

```
7 3 3 2
1 1 1 2 2 1 1
```

## 【样例输出 1】

```
pcftxdy
```

## 【样例输入 2】

```
8 3 2 2
1 1 2 2 1 3 3 3
```

## 【样例输出 2】

```
3 3 3
```

## 【样例输入 3】

8 3 3 2  
1 1 1 2 2 3 3 3

**【样例输出 3】**

1 1 1 3 3 3

**【样例输入 1 说明】**

我们用 red 表示带有一个火元素的花朵，orange 表示带有半个火元素的花朵。  
在刚刚插入的那一刹那，花朵序列是 1 1 1 2 2 2 1 1 。此时 2 2 2 满足条件，  
会被消掉。花朵序列经过碰撞，就会变成 1 1 1 1 1 。此时 1 1 1 1 1 满足条件，  
会被消掉。

**【样例输入 2 说明】**

我们用 red 表示带有一个火元素的花朵，orange 表示带有半个火元素的花朵。  
序列会这样改变：1 1 2 2 2 1 3 3 3 -> 1 1 1 3 3 3 -> 3 3 3  
因为 3 3 3 中间没有火元素，所以他就不会被消掉。

**【样例输入 3 说明】**

我们用 red 表示带有一个火元素的花朵，orange 表示带有半个火元  
素的花朵。序列会这样改变：1 1 1 2 2 2 3 3 3 -> 1 1 1 3 3 3

**【数据范围】**

对数据有  $1 < m < n < 10^6$   
设 C 是可能用到的颜色个数。

测试点编号	n	C	特殊性质
1	=1	= 1	
2	=2	= 2	
3	=10	= 3	不会有花会被消掉
4	=10	= 3	不会有花会被消掉
5	=10	= 3	m = 2
6	=1000	< 1000	
7	=1000	< 1000	
8	=99999	= 233	初始序列随机生成
9	=200000	< 100000	
10	=1000000	< 100000	

# 密码锁 (lock)

## 【问题描述】

小 Z 发明了一种全新的密码锁，该密码锁上有 9 个数字转轮（用字母 A~I 表示），每个数字转轮上只有 0~3 四个数字。与普通密码锁不同的是，该密码锁不允许用户直接转动数字转轮，用户必须通过密码锁上的 9 个特定按钮来完成开锁的操作。每按动一下按钮，都会使得这些数字轮中的某几个数字轮转动一下（数字轮每转动一下，其显示的数字会按照 0→1→2→3→0 的顺序变化），下表列出了每个按钮所影响到的数字轮序号。

按钮	影响的数字轮	按钮	影响的数字轮	按钮	影响的数字轮
1	ABDE	4	ADG	7	DEGH
2	ABC	5	BDEFH	8	GHI
3	BCEF	6	CFI	9	EFHI

现在已知该密码锁的当前状态，请你计算要复位该密码锁（使 9 个数字轮都显示“0”）至少需要按动多少次按钮。

## 【输入格式】

输入数据仅一行，包含 9 个用空格隔开的 0~3 之间的数字，表示密码锁当前的状态。

## 【输出格式】

输出数据仅一个整数，表示要将密码锁复位至少需要按动按钮的次数。

## 【样例输入】

3 3 0 2 2 2 2 1 2

## 【样例输出】

4

## 【样例说明】

第一次：按钮 4，状态变为 0 3 0 3 2 2 3 1 2

第二次：按钮 5，状态变为 0 0 0 0 3 3 3 2 2

第三次：按钮 8，状态变为 0 0 0 0 3 3 0 3 3

第四次：按钮 9，状态变为 0 0 0 0 0 0 0 0 0

**【提示】**

- 1、每个按钮最多只需按多少次？
- 2、按钮的按动次序是否会影响结果？
- 3、每种状态在无多余按键和不考虑按键顺序的情况下有多少种开锁方案？