

ACM算法与程序设计 (一) 提升代码能力

杜育根

ygdu@sei.ecnu.edu.cn

如何提升代码能力

- 代码能力是对于编程语言的应用、读写、调试等一系列的能力，是一种将思维转换成计算机编程语言的能力。
- 提升代码能力没什么“好方法”，关键就是多做题、多动手写代码。
- 养成一个良好的代码风格习惯对于提升代码能力有很大的帮助。
- 程序设计竞赛中的调试技巧

练习题：a+b 问题

- 输入两个整数 a, b ，输出两个整数的和。
- 输入格式：第一行输入一个整数 T ，表示需要计算的次数。接下来 T 行，每行输入两个用空格分隔的整数 a, b 。
- 输出格式：对于每次输入的 a, b ，输出 $a + b$ 的值。结果保证在 32 位整型 (int) 范围内。
- 样例输入
- 5
- 1 2
- 3 4
- 5 6
- 7 8
- 9 10
- 样例输出
- 3
- 7
- 11
- 15
- 19

练习题：斐波那契数列

- 斐波那契数列，输入一个 n ，求出 $f(n)$ 对 $1000000007(10^9+7)$ 取模结果。
- 输入格式：输入一个整数 $n(1 \leq n \leq 100000)$ 。
- 输出格式
- 输入 $f(n) \bmod 1000000007$ 的值。
- 样例输入
- 3
- 样例输出
- 2

练习题：矩阵旋转

- 给出一个 $n \times m$ 的整数矩阵，请你把这个矩阵顺时针旋转 90度以后输出。
- 输入格式：第一行输入两个整数 $n, m (1 \leq n, m \leq 200)$ ，用空格隔开。接下来 n 行，每行输入 m 个整数，表示输入的矩阵。矩阵中元素都是 `int` 范围内的整数。
- 输出格式：输入 m 行，每行 n 个空格隔开的整数，表示旋转以后的矩阵。注意：每行末尾不能输出多余的空格。
- 样例输入
- 3 4
- -1 3 6 3
- 7 7 9 1
- 10 3 4 6
- 样例输出
- 10 7 -1
- 3 7 3
- 4 9 6
- 6 1 3

练习题：最大子阵

- 给定一个 $n \times m$ 的矩阵 A ，求 A 中的一个非空子矩阵，使这个子矩阵中的元素和最大。其中， A 的子矩阵指在 A 中行和列均连续的一部分。
- 输入格式：输入的第一行包含两个整数 $n, m (1 \leq n, m \leq 50)$ ，分别表示矩阵 A 的行数和列数。
- 接下来 n 行，每行 m 个整数，表示矩阵 $A_{i,j} (-1000 \leq A_{i,j} \leq 1000)$ 。
- 输出格式：输出一行，包含一个整数，表示 A 中最大子矩阵的元素和。
- 样例输入
- 3 3
- 2 -4 1
- -1 2 1
- 4 -2 2
- 样例输出
- 6

练习题：四平方和

- 四平方和定理，又称为拉格朗日定理：每个正整数都可以表示为至多四个正整数的平方和。如果把 0 包括进去，就正好可以表示为四个数的平方和。即对于一个给定的正整数 n ，可以表示为： $n = a^2 + b^2 + c^2 + d^2$ 。
- 你需要求出字典序最小的一组解 a, b, c, d 。字典序大小：从左到右依次比较，如果相同则比较下一项，直到有一项不同，较小的一方字典序更小，反之字典序更大，所有项均相同则二者字典序相同。
- 输入格式：程序输入为一个正整数 $N(1 \leq N \leq 5000000)$ 。
- 输出格式：输出四个非负整数 a, b, c, d ，中间用空格分开。
- 样例输入1
- 5
- 样例输出1
- 0 0 1 2
- 样例输入2
- 12
- 样例输出2
- 0 2 2 2

练习题：A+B 问题

- 计算两个非负整数 A,B 的和。不过这一次有些特殊，A,B可能会很大。
- 输入格式：第一行输入一个非负整数 A。 第二行输入一个非负整数 B。
A,B的长度不大于 500。
- 输出格式：输出 A+B的值。不要有多余的前导零。
- 样例输入
- 123
- 1234
- 样例输出
- 1357

练习题：A*B 问题

- 计算两个非负整数 A, B 的乘积， A, B 可能会很大。
- 输入格式
- 第一行输入一个非负整数 A 。
- 第二行输入一个非负整数 B 。
- A, B 的长度不大于 500。
- 输出格式
- 输出 $A \times B$ 的值。

练习题：小明的随机数

- 小明想在学校中请一些同学一起做一项问卷调查，为了确保实验的客观性，他先用计算机生成了 $n(1 \leq n \leq 100)$ 个 1 到 1000 之间的随机整数，对于其中重复的数字，只保留一个，把其余相同的数去掉，不同的数对应着不同的学生的学号。然后再把这些数从小到大排序，按照排好的顺序去找同学做调查。请你协助小明完成“去重”与“排序”的工作。
- 输入格式：共两行，第一行为一个正整数 n 。第二行有 n 个用空格隔开的正整数，为所产生的随机数。
- 输出格式：第一行输出一个正整数 m ，表示不相同的随机数的个数。第二行输出 m 个用空格隔开的正整数，为从小到大排好序的不相同的随机数。
- 样例输入
- 10
- 20 40 32 67 40 20 89 300 400 15
- 样例输出
- 8
- 15 20 32 40 67 89 300 400

练习题：交叉排序

- 有 N 个数 $A_i (1 \leq i \leq N)$ 。现在先将 第 l_1 到第 r_1 个数从小到大排序。再将第 l_2 到第 r_2 从大到小排序。请输出排序后的最终结果。
- 输入格式：第一行五个整数 $N, l_1, r_1, l_2, r_2 (l_1 < r_1, l_2 < r_2)$ ，均不超过 100000。第二行 N 个不超过 `int` 范围的整数 $A_i (1 \leq i \leq N)$ 。
- 输出格式：一行 N 个用空格分隔的整数，表示排序以后的结果。
- 样例输入
- 6 1 3 2 4
- 8 3 1 6 9 2
- 样例输出
- 1 8 6 3 9 2

练习题：进制转换

- 输入一个十进制数 N ，将它转换成 R 进制数输出。在 $10 \leq R \leq 16$ 的情况下，用'A'表示 10，用'B'表示 11，用'B'表示 11，用'C'表示 12，用'D'表示 13，用'E'表示 14，用'F'表示 15。
- 输入格式：输入包含两个整数 $N(N \leq 10000)$ 和 $R(2 \leq R \leq 16)$ 。注意， N 有可能是负整数。
- 输出格式：输出一行，表示转换后的数。
- 样例输入
- 23 12
- 样例输出
- 1B

练习题：回文数

- 一个正整数，如果交换高低位以后和原数相等，那么称这个数为回文数。比如 121，2332 都是回文数，13,4567 不是回文数。任意一个正整数，如果其不是回文数，将该数交换高低位以后和原数相加得到一个新的数，如果新数不是回文数，重复这个变换，直到得到一个回文数为止。例如，57 变换后得到 132(57 + 75)，132 得到 363(132 + 231)，363 是一个回文数。
- 曾经有数学家猜想：对于任意正整数，经过有限次上述变换以后，一定能得出一个回文数。至今这个猜想还没有被证明是对的。现在请你通过程序来验证。
- 输入格式：输入一行一个正整数 n。
- 输出格式：输出第一行一个正整数，表示得到一个回文数的最少变换次数。接下来一行，输出变换过程，相邻的数之间用"--->"连接。输出格式可以参见样例。保证最后生成的数在 int 范围内。
- 样例输入
- 349
- 样例输出
- 3
- 349--->1292--->4213--->7337

练习题：机器人

- 小明收到了一份礼物，是一个最新版的机器人。这个机器人有 4 种指令：
- 1) forward x，前进x米。 2) back x，先向后转，然后前进x米。 3) left x，先向左转，然后前进x米。 4) right x，先向右转，然后前进x米。
- 现在把机器人放在坐标轴原点，起始朝向为x轴正方向。经过一系列指令以后，你能告诉小明机器人的坐标位置吗。坐标轴上一个单位长度表示1米。
- 输入格式：第一行输入一个整数n($1 \leq n \leq 100$) 表示指令的个数。 接下来n行，每行输入形如上面的指令，其中 $-1000 \leq x \leq 1000$ 。
- 输出格式：输出两个整数x,y表示机器人最后坐标。用空格隔开。

样例输入

10

back -9

left 3

left 8

back 15

right 10

right -7

right -3

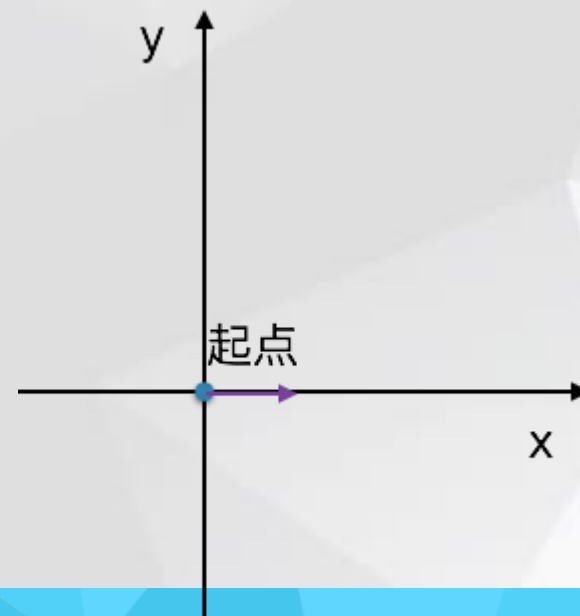
left 11

right 17

left 3

样例输出

9 -7



练习题：表达式求值

- 给定一个只包含加法和乘法的算术表达式，请你编程计算表达式的值的最后四位。
- 输入格式：输入仅有一行，为需要你计算的表达式，表达式中只包含数字、加法运算符 '+' 和乘法运算符 '*'，且没有括号，所有参与运算的数字均为 0-9 之间的整数。输入数据保证这一行只有 0~9、'+'、'*' 这 12 种字符。（输入字符串长度不超过 600000）
- 输出格式：输出只有一行，包含一个整数，表示这个表达式的值。注意：当答案长度多于 4 位时，请只输出最后 4 位。不要输出多余的前导零
- 样例输入1
- $1+1*3+4$
- 样例输出1
- 8
- 样例输入2
- $1+1234567890*1$
- 样例输出2
- 7891

练习题：HZF 爱斗牛

- 斗牛的规则如下，游戏开始的时候，每位玩家发5张扑克牌，每张牌都是A-K中的一张牌（1 表示 A, 2-10 表示对应的牌，11-J, 12-Q, 13-K）。每张牌都有一个点数，1-10 分别对应其点数，11-13 对应点数10。牌型分为以下5种情况，牌型的优先级按照给定从上到下。
- 1. 四炸——5张牌中有4张牌相同，优先级最高。
- 2. 五小牛——即5张牌点数都小于5，且5张牌的点总数和小于或等于10，优先级次之。
- 3. 牛牛——其中3张牌的点数和是10的整数倍，另外2张牌的点数和是10的整数倍。
- 4. 牛 x ($x=1, 2, 3 \dots 9$)——其中3张牌的点数和是10的整数倍，另外2张牌的点数和对10取模为 x 。
- 5. 以上情况都不是，就是无牛。
- hzf 要根据自己的牌型来决定自己的下注的倍数，但是 hzf 总是不能看出他的牌型，你能写一个程序帮助他吗。
- 输入格式：一行输入5个空格隔开的整数。
- 输出格式：若牌型为四炸，输出一行quadra bomb orz。 若为五小牛，输出一行penta calf。若为无牛，输出一行gg。
- 样例输入1
10 10 10 1 1
- 样例输出1
too young too simple:calf 2
- 样例输入2
10 10 10 10 9
- 样例输出2
quadra bomb orz

二进制枚举子集

- 给定一个集合，枚举所有可能的子集。枚举子集的方法有很多，本节我们介绍一种代码编写非常方便的枚举子集方法——二进制法。我们可以用二进制的一位表示集合对应某一元素的选取状态，1表示选取，0表示未选取。举个例子，我们拥有一个集合 $\{0,1,2,3,4,5,6\}$ 。那么二进制0101101就代表子集合 $\{0,2,3,5\}$ ，如下图所示：

2进制数位	6	5	4	3	2	1	0
二进制数值	0	1	0	1	1	0	1
选取的元素	-	5	-	3	2	-	0

位运算

- **位运算** 代码中对于二进制的处理可以用位运算来实现。位运算是对二进制的每一位进行计算，所以每一位只有 00 或 11 两种可能。先介绍三种常用的位运算符：与&、或|、异或^，运算规则如下表所示。
- **与运算**：两者都为 1 时，结果即为 1，否则为 0。 **或运算**：两者都为 0 时，结果即为 0，否则为 1。 **异或运算**：是两者同为 0 或 1 时，结果即为 0，否则为 1。 两个十进制整数进行位运算结果是多少呢？举个例子 $A = 25$ 与 $B = 14$ 做位运算， A 转化为二进制是 11001， B 转化为二进制是 01110，那么 $A \& B$ 、 $A | B$ 、 $A ^ B$ 分别是多少？
- **位运算符**中有两种操作，左移<<和右移>>。右移具体还分为带符号右移与无符号右移，本节我们提到的右移指的是带符号右移，无符号右移使用较少，在这里不做解释。 对于 $A \ll B$ ，表示把 A 转化为二进制后向左移动 B 位（在末尾添加 B 个 0）。 对于 $A \gg B$ ，表示把 A 转化为二进制后向右移动 B 位（删除末尾的 B 位）。 比如 $2 \ll 2$ ，2 转化为二进制则是 10，那么就是 10 左移动 2 位，就变成了二进制 1000，转化为十进制是 8，所以 $2 \ll 2 = 8$ 。

例子

- 我们来看一个可以用二进制枚举的方法解决的题目。
- 话说大诗人李白，一生好饮。幸好他从不开车。 一天，他提着酒壶，从家里出来，酒壶中有酒两斗。他边走边唱： 无事街上走，提壶去打酒。 逢店加一倍，遇花喝一斗。 这一路上，他一共遇到店 5 次，遇到花 10 次，已知最后一次遇到的是花，他正好把酒喝光了。请你计算李白遇到店和花的次序，有多少种可能的方案。
- 这个题目解法很多，二进制枚举是一种写起来非常简洁的解法。我们已知遇店 5 次，遇花 10 次，并且最后一次遇到花，正好把酒喝光。那么我们可以把店作为二进制中的 1，把花作为二进制中的 0，因为已经确定最后一次遇到的是花，所以我们需要判断枚举的结果是否刚好有 5 个 1 和 9 个 0。那么我们就枚举出 14 位二进制的所有可能并加以判断即可，判断思路为判断二进制是否有 9 个 0，5 个 1，并且最终酒刚好剩 1 斗

代码

```
○ int ans = 0;
○ for (int i = 0; i < (1<<14); ++i) {
○     int tot_1 = 0;
○     int tot_0 = 0;
○     int num = 2;
○     for (int j = 0; j < 14; ++j) {
○         if (i&(1 << j)) // 这里判断二进制 i 从右数第 j + 1 位是否为 1
○             {tot_1++;
○              num = num * 2;
○             } else {
○                 tot_0++;
○                 num = num - 1;
○             }
○     }
○     if (tot_1 == 5 && tot_0 == 9 && num == 1) {
○         ++ans; // 记录合法方案数
○     }
○ }
```

练习题：得到整数 X

- 小明有 n 个互不相同的正整数，现在 he 要从这 n 个正整数之中无重复地选取任意个数，并仅通过加法凑出整数 X 。求小明有多少种不同的方案来凑出整数 X 。
- 输入格式
- 第一行，输入两个整数 $n, X (1 \leq n \leq 20, 1 \leq X \leq 2000)$ 。
- 接下来输入 n 个整数，每个整数不超过 100。
- 输出格式
- 输出一个整数，表示能凑出 X 的方案数。
- 样例输入
- 6 6
- 1 2 3 4 5 6
- 样例输出
- 4

练习题：幼儿园买玩具

- 幼儿园有 n 个小朋友，每个小朋友都有自己想玩的玩具。身为幼儿园园长的你决定给幼儿园买一批玩具，由于经费有限，你只能买 m 个玩具。已知玩具商店一共卖 k 种玩具，编号为 $1, 2, 3, \dots, k$ ，你让每个小朋友把想玩的玩具编号都写在了纸上。你希望满足尽可能多的小朋友的需求，请计算出最多能满足多少小朋友的玩具需求。
- 输入格式：第一行，输入三个整数 n, m, k ($1 \leq n \leq 100, 1 \leq m \leq k \leq 15$)，中间用空格分开。接下来 n 行，第 $i+1$ ($0 \leq i < n$) 行的第一个数字 a_i 代表第 i 个小朋友想玩的玩具数量，接下来有 a_i 个数字，代表这 a_i 个玩具的编号。
- 输出格式：输出一个整数，表示最多能满足多少小朋友的玩具需求。
- 样例输入
- 5 3 5
- 2 1 4
- 0
- 2 3 1
- 3 2 3 4
- 2 4 5
- 样例输出
- 3

练习题：islands 打炉石传说

- islands最近在完一款游戏“炉石传说”，又名“魔兽英雄传”。炉石传说是一款卡牌类对战的游戏。游戏是2人对战，总的来说，里面的卡牌分成2类，一类是法术牌，另一类是随从牌（所谓随从就是怪物）。为了简化问题，现在假设随从牌的作用是召唤一个具有一定攻击力的怪物，法术牌的作用是给某个随从增加一定攻击力。随从牌和法术牌的使用都需要消耗一定的法力值。现在islands有10点法力值，手上有n张牌（islands最多有10张牌，否则他将会被爆牌T_T），有些是法术牌，有些是随从牌。islands现在是大劣势，他想要是利用这10点法力值使得召唤出来的所有随从的攻击力总和最高（法力值可以不用完）。注意，任何法术牌都必须使用在某个召唤出来的随从上，也就是如果islands没有召唤过随从，他将不能使用任何法术牌。告诉islands他能召唤的随从的总攻击力最大是多少。
- 输入格式：每组数据首先输入一个 $n(0 \leq n \leq 10)$ ，表示 islands 有 n 张牌。接下来n 行，每行输入 3 个整数cost($0 \leq \text{cost} \leq 10$)，d (0 或者 1)，w($|w| \leq 1000$)。其中 cost表示该牌的法力值消耗，如果d=0，表示该牌是攻击力为 w 的随从牌；如果 d=1，表示是能给一个随从增加w 攻击的法术牌。
- 输出格式：输出一行表示答案。
- 样例输入
- 1
- 1 0 100
- 样例输出
- 100