

# 2022 年程序设计基础（I）阶段测试

东北农业大学

2022 年 11 月 06 日



東北農業大學

Northeast Agricultural University

## 注意事项

1. 比赛时间：19:30 ~ 21:00。
2. 比赛网址：<http://contest.neauacm.cn/>
3. 比赛期间有任何问题，仅限在讨论版区域进行提问，讨论版中禁止进行代码提交等。
4. 比赛期间，允许使用纸质材料，禁止上网百度题目思路，禁止与其他人交流题目，禁止在任何平台泄漏比赛题目。
5. 比赛严厉打击代打等行为，有严格的代码查重机制，通过后台各种手段进行监控，请各位同学诚信参赛。作弊行为一经查实，影响考试成绩的，后果自负。
6. 请在比赛开始前 5 分钟做好准备，准时参加比赛。
7. 本次比赛共 8 题，题目难度与顺序无关，请合理安排解题顺序。
8. 所有题目评测忽略行末空白字符。

## Problem A. exam\_A

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

给你一个整数  $n$ , 请你计算  $n$  的数位和对  $k$  取余的结果, 输出结果对应的英文。

即 0 输出 zero, 1 输出 one, 2 输出 two, 以此类推。

对于数位和的定义是: 一个整数的各位累加起来的和, 例如正整数 123456 的数位和是  $1 + 2 + 3 + 4 + 5 + 6 = 21$ 。

### Input

数据仅包含一行, 输入两个整数,  $n, k$ 。

数据范围:  $1 \leq n \leq 10^9, 1 \leq k \leq 10$

### Output

对于每组测试数据, 输出一行表示答案。

### Example

standard input	standard output
12 5	three

## Problem B. exam\_B

Input file:           standard input  
Output file:          standard output  
Time limit:           1 second  
Memory limit:        128 megabytes

麻将是一种休闲的四人博弈游戏。你的任务是写一个模拟器来模拟一局游戏的进程。

接下来将详细介绍游戏规则和每个玩家的决策。注意：为了实现方便和使游戏更加有趣味，这里介绍的规则和主流的几种麻将规则均略有不同。

基础规则：

一副麻将由 148 张牌组成，其中包含 37 种不同的牌，每种各 4 张。

这 37 种牌分别是：一万到九万 (1M ~9M)、一筒到九筒 (1P ~9P)、一索到九索 (1S ~9S)、东 (E)、南 (S)、西 (W)、北 (N)、白 (B)、发 (F)、中 (Z)，以及 3 种特殊牌：跳过 (PASS)，反向 (REVERSE)，双重回合 (DOUBLE)。

游戏共有 4 名玩家，不妨称其为 A, B, C, D。

游戏开始前，将 148 张牌随机洗乱后摆成一排，称为牌堆。此后玩家摸牌一定是从牌堆中摸取最靠前的一张牌。

从 A 开始按照 ABCDABCD... 的顺序，每人依次从牌堆中摸一张牌，直到每人都有 13 张牌，这些牌组成每个玩家的手牌。

再从 A 开始按照 ABCDABCD... 的顺序，依次进入每人的回合：

在一个回合中，玩家先摸一张牌进入自己的手牌，再从自己的手牌中打出一张牌。

依次进行直到有人和牌或者无牌可摸时游戏结束。

特殊牌：

跳过 (PASS)：在出牌时打出这张牌，可以指定一名玩家，跳过他的下一个回合。

反向 (REVERSE)：在出牌时打出这张牌，反转进行回合的顺序，即由 ABCDABCD... 变为 ADCBADCB... 或由 ADCBADCB... 变为 ABCDABCD...。出牌后即按照反转后的顺序，从出牌者原先的上家开始进行回合。

双重回合 (DOUBLE)：在出牌时打出这张牌，该名玩家立即进入一个额外的回合。

牌型：有如下 3 种牌型：

顺子：3 张数字连续的万，或 3 张数字连续的筒，或 3 张数字连续的索，如 4P 5P 6P。

刻子：3 张完全一样的非特殊牌，如 B B B。

对子：2 张完全一样的非特殊牌，如 9M 9M。

吃、碰：

当一名玩家打出一张非特殊牌时，其他玩家可以进行吃或碰：

吃 (CHOW)：当打出的这张牌跟自己的手牌中的某两张牌能组成一个顺子时，可以将手牌中能与之组成顺子的其余两张牌取出，与这张牌一起摆在旁边。

注意只有上一名出牌玩家的下家（按当前顺序原本应在下一个进行回合的玩家）才能吃。

碰 (PONG): 当打出的这张牌跟自己的手牌中的某两张牌能组成一个刻子时, 可以将手牌中能与之组成刻子的其余两张牌取出, 与这张牌一起摆在旁边。

碰没有吃的上述限制, 任意其他玩家都能碰。

如果既有玩家能吃又有玩家能碰, 则碰优先于吃。

吃 (或碰) 不是强制性的, 也就是说玩家满足吃 (或碰) 的条件时, 可以选择不吃 (或碰)。

吃和碰统称为副露。为方便起见, 不将副露视为手牌的一部分。

在任意一名玩家吃 (或碰) 后, 跳过从上一名出牌的玩家到这名玩家之间的所有玩家的回合, 直接从当前玩家开始进行新的回合。但该玩家在这一回合中跳过摸牌直接出牌, 在下一回合 (如果没有吃碰的话) 恢复正常。

注意在本规则中不能杠。

胡牌规则:

称一名玩家的牌能和, 当且仅当满足如下条件:

牌数为  $14-3n$ , 其中  $n$  为该玩家副露 (即吃碰) 的个数;

这些牌中无特殊牌;

这些牌能够被分成  $(5-n)$  组, 其中  $(4-n)$  组均为 3 张且均为顺子或刻子, 其余一组为 2 张且为对子。

注意本规则中不支持七对子、十三幺、全不靠等特殊的和牌规则。

另外, 定义一组包含  $13-3n$  张牌的手牌的和牌距离为最小的  $x$ , 使得向这些牌中加入特定的  $x$  张牌, 再去掉  $x-1$  张手牌后, 每种牌仍不超过 4 张且能和。

定义一组包含  $14-3n$  张牌的手牌的和牌距离为最小的  $x$ , 使得向这些牌中加入特定的  $x$  张牌, 再去掉  $x$  张手牌后, 每种牌仍不超过 4 张且能和。

特别地, 一手能和的牌的和牌距离为 0; 和牌距离为 1 的牌称为听牌。

注意这里的“每种牌仍不超过 4 张”的限制: 如果一手牌是 1M 1M 1M 1M 且副露数为 3, 再向其中加入一张 1M 就能和, 但是由于有 5 张 1M 所以是不被允许的, 故不认为其和牌距离为 1。

但如果一手牌是 1M 且副露数为 4, 但是曾进行过一次 1M 1M 1M 的碰, 仍然认为其和牌距离为 1 (虽然缺的这张 1M 永远也等不到)。

终局:

荣和 (RON): 当一名玩家出牌后, 某名其他玩家的手牌加上这张牌能和, 则称这名玩家荣和。荣和优先于吃碰。

如果有多名玩家同时达到荣和的标准, 规定只有从上一名出牌玩家开始, 沿回合进行顺序的第一名能荣和的玩家才能荣和, 其余玩家荣和不了, 称这种情况为截和。

自摸 (SELFDRAWN): 一名玩家摸牌后其手牌能和, 称这名玩家自摸。

一旦有一名玩家荣和或自摸, 游戏立即结束, 该名玩家胜利。

如果某名玩家摸牌时发现牌堆中已经无牌可摸, 游戏立即结束, 称此种情况为流局。

出牌策略: 每名玩家的出牌策略相同且固定:

出牌时, 若手里有特殊牌一定会优先出, 且如果有多种特殊牌, 按照 PASS、REVERSE、DOUBLE 的优先顺序; 出的 PASS 一定指定下家。

出牌时若手里没有特殊牌, 则会对于每一种可能的出牌方法计算出完牌后的和牌距离, 选择和牌距离最小的一种方案。如果有并列最小, 按照 Z, F, B, N, W, S, E, 9S, 8S, ..., 1S, 9P, ..., 1P, 9M, ..., 1M 的优先顺序出牌。

同一个人能吃且能碰时，优先考虑碰；因为每种牌只有 4 张所以不会有两名玩家同时可以碰的情况；当且仅当吃（或碰）后能使得和牌距离严格减小才会去吃（或碰）；如果有多种吃的方案使得和牌距离严格减小，优先选择数字较大的方案。

能荣和一定荣和（除非被截和），能自摸一定自摸，不会拒和（能和时故意选择不和）。

## Input

输入共 148 行，按照牌堆从前到后的顺序输入每一张牌。

每行输入一个字符串表示这一张牌。

用 1M, 2M, …… , 9M 代表万, 1P, 2P, …… , 9P 代表筒, 1S, 2S, …… , 9S 代表索, E, S, W, N, B, F, Z 分别代表东、南、西、北、白、发、中, PASS 代表跳过, REVERSE 表示反向, DOUBLE 表示双重回合。

## Output

按照如下几条规则进行输出：

当任意一名玩家摸牌时（包括游戏最开始的摸牌），输出一行：

x IN y

其中 x 为玩家名称, y 为摸到的牌。

当任意一名玩家出牌时，如果出的牌不是 PASS，输出一行：

x OUT y

其中 x 为玩家名称, y 为出的牌。

如果出的牌是 PASS，应当输出一行：

x OUT PASS z

其中 z 为 PASS 指定的对象。

当任意一名玩家吃时，输出一行：

x CHOW y1 y2 y3

其中 x 为玩家名称, y1, y2, y3 为吃涉及到的 3 张牌，按数字递增的顺序输出。

当任意一名玩家碰时，输出一行：

x PONG y1 y2 y3

其中 x 为玩家名称, y1, y2, y3 为碰涉及到的 3 张牌，根据碰的规则, y1, y2, y3 应相同。

当任意一名玩家荣和时，输出一行：

x RON

其中 x 为玩家名称。

当任意一名玩家自摸时，输出一行：

x SELFDRAWN

其中 x 为玩家名称。

游戏的最后，如果某名玩家获得胜利，输出一行：

x WIN

其中 x 为玩家名称。

如果出现流局，输出一行：

DRAW

需要特别注意的是，输入输出中出现的英文字母均为大写。

## Example

standard input	standard output
8M	A IN 8M
Z	B IN Z
E	C IN E
9P	D IN 9P
3P	A IN 3P
9S	B IN 9S
5P	C IN 5P
W	D IN W
3M	A IN 3M
8P	B IN 8P
DOUBLE	C IN DOUBLE
5P	D IN 5P
Z	A IN Z
2P	B IN 2P
3M	C IN 3M
8S	D IN 8S
2S	A IN 2S
5P	B IN 5P
5M	C IN 5M
E	D IN E
6M	A IN 6M
9S	B IN 9S
6P	C IN 6P
5S	D IN 5S
7M	A IN 7M
4S	B IN 4S
3S	C IN 3S
6M	D IN 6M
3S	A IN 3S
2M	B IN 2M
9M	C IN 9M
5S	D IN 5S
Z	A IN Z
7P	B IN 7P
5P	C IN 5P
8M	D IN 8M
3M	A IN 3M
F	B IN F
7M	C IN 7M
2S	D IN 2S
N	A IN N
4P	B IN 4P
3S	C IN 3S
S	D IN S
PASS	A IN PASS
1P	B IN 1P
6S	C IN 6S
3P	D IN 3P
9P	A IN 9P
9S	B IN 9S
4M	C IN 4M

standard input	standard output
8P	D IN 8P
N	A IN N
Z	A OUT PASS B
N	C IN Z
5M	C OUT DOUBLE
DOUBLE	C IN N
REVERSE	C OUT Z
S	A PONG Z Z Z
3P	A OUT 9P
4M	B CHOW 7P 8P 9P
4S	B OUT Z
1S	C IN 5M
PASS	C OUT N
4P	A PONG N N N
6S	A OUT 3P
7S	B CHOW 3P 4P 5P
7P	B OUT F
6S	C IN DOUBLE
9M	C OUT DOUBLE
REVERSE	C IN REVERSE
3P	C OUT REVERSE
7P	B IN S
DOUBLE	B OUT S
B	A IN 3P
9P	A OUT 3P
4S	D IN 4M
5S	D OUT W
7S	C IN 4S
7S	C OUT E
7P	B IN 1S
6S	B OUT 4S
9S	A RON
B	A WIN
9M	
S	
F	
2P	
1P	
PASS	
9P	
DOUBLE	
4P	
PASS	
5S	
2M	
2P	
6P	
W	
1M	
8S	
REVERSE	
8M	

standard input	standard output
6M	
5M	
F	
4M	
F	
8P	
2S	
1M	
2M	
3M	
7M	
3S	
B	
7S	
1S	
REVERSE	
8P	
6M	
4S	
2M	
B	
1M	
S	
6P	
5M	
W	
7M	
2S	
8S	
8M	
1S	
4P	
E	
4M	
9M	
1M	
8S	
1S	
1P	
2P	
1P	
W	
6P	
N	
E	



## Problem C. exam\_C

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

给你一个仅由小写字母组成的字符串，我们想用某种策略将字符串压缩。

这种策略是：请你计算每一个字符的出现次数，并用一个该字符和字符出现的次数来代替原本的重复字符来压缩字符串。

即如果该字符串为 *aaa*，那么你需要输出 *a3*，如果该字符串为 *abbccc*，那么你需要输出 *a1b2c3*。

给出原本的字符串，请你输出压缩后的结果。

### Input

输入仅包含一行，一个仅由小写字母组成的字符串 *s*。

数据范围：  $1 \leq \text{len}(s) \leq 10^3$

### Output

对于每组数据输出占一行，为压缩后的字符串。

### Example

standard input	standard output
ddghhjllsooolohbbbbbb	d2g1h2j4l2s1o3l1o1h1b6

## Problem D. exam\_D

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

给你一个数字  $n$ , 求  $1 \sim n$  中所有  $n$  的正整数的因子的个数的和。

### Input

输入仅包含一个数字  $n$ 。

数据范围:  $1 \leq n \leq 10^4$

### Output

对于每一组测试数据, 输出一个数表示答案。

### Example

standard input	standard output
10	4

## Problem E. exam\_E

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 128 megabytes

给你  $n$  个数，然后现在有  $m$  次操作，每一次操作有如下两种类型。

操作 1：将区间内的一段数字从小到大排序。

操作 2：将区间内的一段数字从大到小排序。

现在想让你输出经过  $m$  次操作后的这个序列。

### Input

输入第一行两个整数  $n$  和  $m$ ，代表  $n$  个数， $m$  次操作。

第二行输入  $n$  个数，代表数组中的每一个元素的值。

接下来  $m$  行，每一行 3 个数  $op, l, r$ ，其中  $op$  为 1 或 2，代表上述的操作 1 和操作 2， $l$  和  $r$  代表要排序的区间。

数据范围：  $1 \leq n, m \leq 3 \times 10^4, 1 \leq op \leq 2, 1 \leq l \leq r \leq n, 1 \leq a_i \leq 10^9$

### Output

对于每组数据，输出  $m$  次操作后的序列情况

### Examples

standard input	standard output
5 5 1 4 2 3 5 1 1 3 1 1 5 2 2 3 1 1 4 2 1 3	3 2 1 4 5

## Problem F. exam\_F

Input file: standard input  
Output file: standard output  
Time limit: 1 second  
Memory limit: 128 megabytes

现在给你一个数组，这个数组里面初始状态是空的，现在有  $q$  次操作，每一次操作的指令如下：

- 1.push  $x$  将  $x$  插入数组的末端，注意数组里第一个元素的位置默认为 1。
- 2.sum 求该数组的所有元素之和。
- 3.memset 将该数组所有元素变为 0。
- 4.sort 1 将该数组的所有元素按照从小到大进行排序。
- 5.sort 2 将该数组的所有元素按照从大到小进行排序。
- 6.max 求该数组里面的最大值。
- 7.min 求该数组里面的最小值。

现在想让你帮忙处理这些操作。

### Input

输入第一行一个整数  $q$ ，代表有  $q$  次操作。

接下来  $q$  行，每一行都有一个操作，保证所有操作均为以上 8 种的任何一种。

数据范围：  $1 \leq q \leq 10^6$

数据保证操作的所有数都在 int 内，并且数据保证数组在进行操作的时候不会改变数组的最小值。

### Output

对于操作 2，输出一行一个整数代表数组元素之和。

对于操作 4 和 5，输出一行，包括该数组里的所有元素，并按照要求进行排序输出，注意，在此操作之后所有的操作均按照排序完成的数组顺序进行操作。

对于操作 6 和 7，输出一行一个整数代表最大或最小值。

对于其他操作，不需要输出任何字符。

### Examples

standard input	standard output
12	15
push 2	5
push 1	1
push 3	1 2 3 4 5
push 4	5 4 3 2 1
push 5	0
sum	
max	
min	
sort 1	
sort 2	
memset	
sum	

## Problem G. exam\_G

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

有一个  $n \times m$  的矩阵，我们定义左上角为 (1,1)，右下角为 (n,m)。

其中的每一个元素都为 0 或 1。

现在我们知道初始的矩阵，有  $q$  次操作，每次操作都能使  $(x_1, y_1)$  到  $(x_2, y_2)$  的矩阵内所有路灯状态反转。

请你输出  $q$  次操作之后的状态。

### Input

每组数据第一行输入三个数字  $n, m, q$ 。

接下来的  $n$  行  $m$  列代表这个矩阵的初始状态。

接下来的  $q$  行，输入  $x_1, y_1, x_2, y_2$ 。

数据范围：  $1 \leq n, m \leq 200, 1 \leq q \leq 10^3, 1 \leq x_1, x_2 \leq n, 1 \leq y_1, y_2 \leq m$

### Output

输出操作后的  $n \times m$  的矩阵，中间没有空格。

### Examples

standard input	standard output
3 4 2	1110
0 0 0 0	1001
0 0 0 0	0111
0 0 0 0	
1 1 2 3	
2 2 3 4	

## Problem H. exam\_H

Input file:           standard input  
Output file:         standard output  
Time limit:          1 second  
Memory limit:       128 megabytes

现在有  $n$  个任务，每一个任务需要  $k$  天，每一天只能做一个任务，求所有任务都做完需要多少天。

### Input

输入一行包含两个整数  $n, k$ 。

数据范围:  $1 \leq n, k \leq 10^4$

### Output

对于每组数据输出一个整数，表示答案。

### Example

standard input	standard output
5 3	15