

练习题 04 数组

说明：第一部分为离线题，无测试数据，通过样例为正确，以练习 c++ 代码为目的。数据范围无特别说明均为 int 范围。

4.1 最小数

输入 n 个数，找出最小数所在的位置，并与第一个数对调。

输入文件：min.in 输出文件：min.out

样例输入：

5 //n 的个数

12 27 3 15 8

样例输出：

3 //最小数出现的位置

3 27 12 15 8

4.2 排序

输入 n 个数，请分别使用冒泡法和选择法排序。

提示：这是很重要的基本功，要掌握纯熟，要求达到可以默写核心语句的程度。请分别编写两个函数实现。

输入文件：rank.in 输出文件：rank.out

样例输入：

略

样例输出：

略

4.3 折半查找

查找是程序中最常见的数据处理问题。对一个无序的数列来说，最原始的方法就是从头到尾逐个查找。但是一个已经有序地数组，折半查找的执行效率要高得多。

给定有序数列，从中找出指定的数字。

输入文件：half.in 输出文件：half.out

样例输入 1：

7 15 //有序数列的长度；待查找的数字

2 5 12 15 21 22 36

样例输出 1：

4

样例输入 2：

3 18

11 16 18

样例输出 2：

3

样例输入 3：

5 9

2 5 12 15 21

样例输出 3 :

No answer!

4.4 插入数字

在一个已经排好序的数组中，插入一个数字，要求使得该数组仍然有序。

输入文件：insert.in 输出文件：insert.out

样例输入：

20 //待插入数字

7 11 12 15 28

样例输出：

7 11 12 15 20 28

4.5 数组的平移

有一个已经排好序的数组，输入序号 m，将 m 以前（含 m）的数和 m 以后的数整体对调。要求不能开辟新的数组，但允许数组的长度+1。

如：原数组：18 27 29 45 62 83 85，输入 m=3，则数组变为：45 62 83 85 18 27 29。

输入文件：move.in 输出文件：move.out

样例输入：

3

18 27 29 45 62 83 85

样例输出：

45 62 83 85 18 27 29

4.6 数组的合并

用边排序边合并的方法把两个有序数列合并为一个新的有序数列，不得先合并再重新排序。

输入文件：merge.in 输出文件：merge.out

样例输入：

5 6 //第一个数列的长度、第二个数列的长度

5 10 14 17 23

3 7 15 17 24 28

样例输出：

3 5 7 10 14 15 17 17 23 24 28

4.7 数字黑洞 6174

已知：一个任意的四位正整数。将数字重新组合成一个最大的数和最小的数相减，重复这个过程，最多七步，必得 6174，将永远出不来。

求证：所有四位数数字（全相同的除外），均能得到 6174。输出掉进黑洞的步数。

输入文件：6174.in 输出文件：6174.out

样例输入 1：

10

样例输出 1：

Input error!

样例输入 2 :

2015

样例输出 2 :

7

4.8 开关问题

有 n 盏关闭的灯, 编号为 $1 \sim n$, 第一个人按下所有灯的开关, 第二个人按下所有编号为 2 的倍数的灯的开关, 第三个人按下所有编号为 3 的倍数的灯的开关, 依次类推, 一共 k 个人。最后哪些灯开着? 输入 n, k , 输出开着的灯的编号。

输入文件: lamp.in 输出文件: lamp.out

样例输入:

7 3

样例输出:

1 5 6 7

4.9 筛选法求素数

用筛选法求素数是一个高效率求素数的方法。

其做法是从 2 开始的自然数列中, 首先确定第一个数 2 是素数, 然后筛去所有 2 的倍数; 接下来找到数列中还剩的第一个数, 它也是素数, 然后筛去它的所有倍数, 如此继续直到无数可筛为止。这是数列中剩下的就全部是素数了。

请输出 200 以内的全部素数。

输出文件: filter.out

样例输出:

略

4.10 约瑟夫问题

n 个人围成一圈, 从第一个人开始报数, 数到 m 的人出列; 再从下一个人开始报数, 如此往复, 输出出列的人的先后顺序。

输入文件: josephu.in 输出文件: josephu.out

样例输入:

7 4 // n 个人; 数到 m

样例输出:

4 1 6 5 7 3 2

4.11 猴子选大王

n 只猴子选大王, 选举办法如下: 从头到尾 1、2、3 报数, 凡报 3 的退出; 余下的从尾到头 1、2、3 报数, 仍是报 3 的退出, 如此往复, 直到剩下 1 只猴子。若想当大王, 则当初应该占据什么位置?

输入文件: monkey.in 输出文件: monkey.out

样例输入:

20

样例输出:

4

4.12 高精度阶乘

对于阶乘函数，即使自变量很小，其函数值也会相当大。例如：

$$10! = 3628800$$

$$25! = 1551121004330985984000000$$

很明显的，无论是 long int 型数据还是 long double 型数据，都无法支持太大的自变量。

请你设计一个程序，当输入一个正整数 n 时，输出 $n!$ 的精确表示法。

输入文件：high-precision.in 输出文件：high-precision.out

样例输入：

20

样例输出：

2432902008176640000

4.13 输入一个 $n \times m$ 的方阵，找出其中的最大值及其位置。

输入文件：matrix.in 输出文件：matrix.out

测试数据：(矩阵) 答案：45 [1] [4]

13	4	14	45	6
10	5	2	6	8
2	23	21	8	10
17	11	7	12	26
9	6	19	20	3

4.14 马鞍数

求一个 $n \times n$ 数阵中的马鞍数，输出它的位置。所谓马鞍数，是指在行上最小而在列上最大的数。

输入文件：saddle.in 输出文件：saddle.out

测试数据：(矩阵 1) 答案：8 [2] [5] (矩阵 2) 答案：No find

15	4	14	25	7	11	4	2	7	8
11	9	23	16	8	5	9	23	1	25
22	3	21	18	5	3	22	21	18	15
17	1	24	12	6	17	16	24	12	6
13	10	19	20	2	13	10	19	20	14

4.15 蛇形矩阵

输入文件：snake.in 输出文件：snake.out

输入正整数 N ，生成一个按蛇形方式排列的矩阵。如下图的蛇形矩阵。

$N=4$

1	3	4	10
2	5	9	11
6	8	12	15
7	13	14	16

$N=5$

1	3	4	10	11
2	5	9	12	19
6	8	13	18	20
7	14	17	21	24
15	16	22	23	25

提示：蛇形矩阵只有四种走向，即向下走、向右上走、向右走、向左下方走，因此可以用 switch 语句控制方向。下面的关键问题是方向改变时的第一个数据的行列坐标，仔细观察当行列坐标分别变化为哪些特殊数时才改变方向，改变方向后的第一个数据的行列的坐标怎样变换。

4.16 螺旋矩阵

输入正整数 N，在 N*N 的方阵中，输出如下图的 N 阶螺旋方阵。

输入文件：screw.in 输出文件：screw.out

N=3		
7	8	1
6	9	2
5	4	3

N=4			
10	11	12	1
9	16	13	2
8	15	14	3
7	6	5	4

提示：螺旋矩阵在每一次大循环内有四个方向，向下、向左、向上、向右。四个方向填充数据可以用小循环控制填数，仔细观察在各个方向的小循环内填充数据时，在什么情况下会结束小循环转到下一个方向填充数据。

另外，可以使用 switch 语句来控制四个方向。

说明：第二部分为在线题，部分有测试数据，以网站通过为正确。部分略有思维难度。数据范围以题意描述为准。

Tyvj1012/Vijos1496 火柴棒等式

Tyvj1020 寻找质因数

Tyvj1026 犁田机器人

Tyvj1093 验证数独

Tyvj1218/Vijos1123 均分纸牌

Tyvj1247 派发数字

Tyvj1269 安全逃离

Tyvj1273 排序集合

Tyvj1398 生命游戏

Tyvj1497 旅游路线

Tyvj1599 锻炼计划

Tyvj1619 高中运动会

Tyvj1658 奶牛同盟

Tyvj2012 矩阵交换