

国庆集训第1次模拟测试

题目名称	分糖果	开心消消乐	像素块	数列
题目类型	传统型	传统型	传统型	传统型
目录	candy	eliminate	pixel	sequence
可执行文件名	candy	eliminate	pixel	sequence
输入文件名	candy.in	eliminate.in	pixel.in	sequence.in
输出文件名	candy.out	eliminate.out	pixel.out	sequence.out
每个测试点时限	1.0秒	1.0秒	1.0秒	1.0秒
内存限制	512 MB	512 MB	512 MB	512 MB
测试点数目	10	10	10	10
每个测试点分值	10	10	10	10

提交源程序文件名

对于C++ 语言	candy.cpp	eliminate.cpp	pixel.cpp	sequence.cpp
对于C 语言	candy.c	eliminate.c	pixel.c	sequence.c
对于pascal 语言	candy.pas	eliminate.pas	pixel.pas	sequence.pas

编译选项

对于C++ 语言	-lm	-lm	-lm	-lm
对于C 语言	-lm	-lm	-lm	-lm
对于pascal 语言				

注意事项：

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 除非特殊说明, 结果比较方式均为忽略行末空格及文末回车的全文比较。
3. 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。
4. 全国统一评测时采用的机器配置为：Cpu AMD Athlon(tm) II x2 240 processor, 2.8GHz, 内存4G, 上述时限以此配置为准。
5. 只提供Linux格式附加样例文件。
6. 评测在NOI Linux下进行。
7. 编译时不打开任何优化选项。

分糖果 (candy)

【问题描述】

排排坐，分糖果。

我们买了一些糖果candies，打算把它们分给排好队的 $n = \text{num_people}$ 个小朋友。

给第一个小朋友 1 颗糖果，第二个小朋友 2 颗，依此类推，直到给最后一个小朋友 n 颗糖果。

然后，我们再回到队伍的起点，给第一个小朋友 $n + 1$ 颗糖果，第二个小朋友 $n + 2$ 颗，依此类推，直到给最后一个小朋友 $2 * n$ 颗糖果。

重复上述过程（每次都比上一次多给出一颗糖果，当到达队伍终点后再次从队伍起点开始），直到我们分完所有的糖果。注意，就算我们手中的剩下糖果数不够（不比前一次发出的糖果多），这些糖果也会全部发给当前的小朋友。

返回一个长度为 num_people 、元素之和为 candies 的数组，以表示糖果的最终分发情况（即 $\text{ans}[i]$ 表示第 i 个小朋友分到的糖果数）。

【输入格式】

从文件***candy.in*** 中读入数据。

输入的第一行包含两个正整数candies和 num_people ，表示糖果的总数和排好队的小朋友个数。

【输出格式】

输出到文件***candy.out*** 中。

输出一行一个数组ans，表示每个小朋友分到的糖果数目。

【样例1输入】

7 4

【样例1输出】

1 2 3 1

【样例1说明】

第一次，`ans[0] += 1`，数组变为 `[1, 0, 0, 0]`。

第二次，`ans[1] += 2`，数组变为 `[1, 2, 0, 0]`。

第三次，`ans[2] += 3`，数组变为 `[1, 2, 3, 0]`。

第四次，`ans[3] += 1`（因为此时只剩下 1 颗糖果），最终数组变为 `[1, 2, 3, 1]`。

【样例2输入】

10 3

【样例2输出】

5 2 3

【数据范围】

$1 \leq \text{candies} \leq 10^9$

$1 \leq \text{num_people} \leq 1000$

开心消消乐 (eliminate)

【问题描述】

现在给你一个字符串，你要做的就是当这个字符串中存在两个挨着的字符是相同的时就将这两个字符消除。需要注意的是，当把这两个字符消除后，可能又产生一对新的挨着的字符是相同的。

比如，初始的字符串是abcddc，dd是两个挨着的相同的字符，当把“dd”消除后，得到的字符串是abcc，这时cc又是两个挨着的相同的字符，所以又应该把cc消除。重复以上操作直到剩下的串中不存在两个挨着的字符是相同的为止，输出最终剩下的串。

另需要注意的是，多对相同字符的消除顺序是不会对答案产生影响的，可以证明最后他们都会达到唯一的结果，比如，对于初始字符串adccdeed，无论是adccdeed→addeed→aeed→ad还是adccdeed→adccdd→adcc→ad，最终的输出结果都是ad。

【输入格式】

从文件`eliminate.in` 中读入数据。

输入的第一行，包含一个字符串，为初始字符串，所有的字符均为小写字母。

【输出格式】

输出到文件`eliminate.out` 中。

输出为一行，包含一个字符串，为执行多次消除操作后最终剩下的字符串。

【提示】

对于100%的数据，字符串的长度在1到200000之间。

【样例1输入】

adccdeed

【样例1输出】

ad

像素块 (pixel)

【问题描述】

现在有一个 $n \times m$ 的像素矩阵，每一个像素点只有黑或白两种颜色，你的任务是把给定的像素矩阵转化成满足要求的矩阵。

要求为：

1. 每一列的所有像素点颜色相同
2. 相同颜色的连续像素列长度应在 x 到 y 之间。

请计算最小需要改变多少个像素点的颜色。

【输入格式】

从文件`pixel.in` 中读入数据。

第一行有四个整数 n, m, x, y ，分别矩阵长宽和连续像素列的长度限制。

接下来 n 行每行 m 个字符 ‘#’ 或 ‘.’ 表示像素点颜色

【输出格式】

输出到文件`pixel.out` 中。

输出一个整数，即最少需要改变的像素点颜色

【样例1输入】

6 5 1 2

##.#.

.###.

###..

#...#

.##.#

###..

【样例1输出】

11

【样例2输入】

2 5 1 1

#####

.....

【样例2输出】

5

【数据范围】

对于20%数据点， $x=1, y=1$

对于50%数据点， $1 \leq n, m \leq 20$

对于100%数据点， $1 \leq n, m, x, y \leq 1000, x \leq y$

数列 (sequence)

【问题描述】

$$a[1]=a[2]=a[3]=1$$

$$a[x]=a[x-3]+a[x-1] \quad (x>3)$$

求a数列的第n项对1000000007 (10^9+7) 取余的值。

【输入格式】

从文件`sequence.in` 中读入数据。

第一行一个整数T，表示询问个数。

以下T行，每行一个正整数n。

【输出格式】

输出到文件`sequence.out` 中。

每行输出一个非负整数表示答案。

【样例1输入】

3
6
8
10

【样例1输出】

4
9
19

【数据范围】

对于30%的数据 $n \leq 100$;

对于60%的数据 $n \leq 2 \times 10^7$;

对于100%的数据 $T \leq 100$, $n \leq 2 \times 10^9$;

