

NOIP2016 模拟试题

提高组day2

(请选手务必仔细阅读本页内容)

一. 题目概况

中文题目名称	第K小数	abcd	好数
英文题目与子目录名	number	abcd	good
可执行文件名	number	abcd	good
输入文件名	number.in	abcd.in	good.in
输出文件名	number.out	abcd.out	good.out
每个测试点时限	1 秒	1 秒	1 秒
测试点数目	20	10	10
每个测试点分值	5	10	10
附加样例文件	有	有	有
结果比较方式	全文比较（过滤行末空格及文末回车）		
题目类型	传统	传统	传统
运行内存上限	128M	128M	128M

二. 提交源程序文件名

对于C++语言	number.cpp	abcd.cpp	good.cpp
对于C语言	number.c	abcd.c	good.c
对于pascal语言	number.pas	abcd.pas	good.pas

三. 编译命令（不包含任何优化开关）

对于C++语言	g++ -o number number.cpp -lm	g++ -o abcd abcd.cpp -lm	g++ -o good good.cpp -lm
对于C语言	gcc -o number number.c -lm	gcc -o abcd abcd.c -lm	gcc -o good good.c -lm
对于pascal语言	fpc number.pas	fpc abcd.pas	fpc good.pas

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、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、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。
- 4、只提供 Linux 格式附加样例文件。
- 5、特别提醒：评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以其为准。

1. 第K小数

(number.cpp/c/pas)

【问题描述】

有两个正整数数列，元素个数分别为N和M。从两个数列中分别任取一个数相乘，这样一共可以得到N*M个数，询问这N*M个数中第K小数是多少。

【输入格式】

输入文件名为number.in。

输入文件包含三行。

第一行为三个正整数N, M和K。

第二行为N个正整数，表示第一个数列。

第三行为M个正整数，表示第二个数列。

【输出格式】

输出文件名为number.out。

输出文件包含一行，一个正整数表示第K小数。

【输入输出样例1】

number.in	number.out
2 3 4 1 2 2 1 3	3

【输入输出样例2】

number.in	number.out
5 5 18 7 2 3 5 8 3 1 3 2 5	16

【数据规模与约定】

样例点编号	N	M	K	元素大小(\leq)
1	20	20	150	10^4
2	50	50	2000	10^4
3	100	80	5000	10^9
4	200	200	26000	10^9
5	10000	10000	50050000	10^4
6	1000	20000	9500000	10^4
7	1000	20000	10000500	10^9
8	2000	20000	190000	10^9
9	2000	20000	199000	10^9
10	20000	20000	210005000	10^4
11	20000	20000	210000	10^5
12	20000	20000	200000	10^9

13	20000	20000	220000500	10^5
14	20000	20000	199000500	10^9
15	200000	200000	180000	10^4
16	200000	200000	200000	10^9
17	2000	200000	100001500	10^9
18	200000	180000	19550000000	10^5
19	200000	200000	19900010000	10^9
20	200000	200000	20000010000	10^9

2. abcd
(abcd. cpp/c/pas)

【问题描述】

有4个长度为N的数组a, b, c, d。现在需要你选择N个数构成数组e，数组e满足 $a[i] \leq e[i] \leq b[i]$ 以及

$$\sum_{i=1}^N e[i] * c[i] = 0$$

并且使得

$$\sum_{i=1}^N e[i] * d[i]$$

最大。

【输入格式】

输入文件名为abcd. in。

输入文件共 N+1 行。

第 1 行包含1个正整数N。

第 i+1 行包含4个整数a[i], b[i], c[i], d[i]。

【输出格式】

输出文件名为abcd. out。

输出共1行，包含1个整数，表示所给出公式的最大值。输入数据保证一定有解。

【输入输出样例1】

abcd. in	abcd. out
5 -1 1 2 5 -2 2 1 2 0 1 1 3 -2 -1 3 10 -2 2 3 9	2

【输入输出样例2】

abcd. in	abcd. out
10 1 10 1 7 -10 10 2 0 -10 10 2 2 -10 10 2 0 1 10 1 0 -10 10 2 0 10 10 2 0 1 10 1 0 -10 10 2 0	90

1 10 1 0	
----------	--

【输入输出样例3】

abcd. in	abcd. out
10	-4
1 10 1 0	
-10 10 2 2	
-10 10 2 2	
-10 10 2 2	
1 10 1 0	
-10 10 2 2	
-10 10 2 2	
1 10 1 0	
-10 10 2 2	
1 10 1 0	

【数据规模与约定】

对于 20%的数据， $N \leq 10$ ， $-2 \leq a[i] < b[i] \leq 2$ ；

对于 60%的数据， $N \leq 50$ ， $-20 \leq a[i] < b[i] \leq 20$ ；

对于 100%的数据，

$N \leq 200$ ， $-25 \leq a[i] < b[i] \leq 25$ ， $1 \leq c[i] \leq 20$ ， $0 \leq p[i] \leq 100000$ 。

3. 好数

(good. cpp/c/pas)

【问题描述】

我们定义一个非负整数是“好数”，当且仅当它符合以下条件之一：

1. 这个数是0或1
2. 所有小于这个数且与它互质的正整数可以排成一个等差数列

例如，8就是一个好数，因为1, 3, 5, 7排成了等差数列。

给出N个非负整数，然后进行如下三个操作：

1. 询问区间[L, R]有多少个好数
2. 将区间[L, R]内所有数对S取余 ($S \leq 1000000$)
3. 将第C个数更改为X

提示：如果你不知道如何判断一个数是否为好数，你可以打个表找找规律。

【输入格式】

输入文件名为good. in。

第一行包含两个正整数N和M，M表示操作数目

第二行包含N个非负整数。

接下来的M行每行表示1个操作：“1 L R”表示第1个操作，“2 L R S”表示第2个操作，“3 C X”表示第3个操作。

【输出格式】

输出文件名为color. out。

对每个操作1，输出一个非负整数，表示区间内好数的个数。

【输入输出样例1】

good. in	good. out
3 6	2
4 6 9	0
1 1 3	2
1 3 3	2
2 1 1 10	
1 1 3	
3 2 4	
1 1 3	

【输入输出样例2】

good. in	good. out
8 5	3
12 24 17 31 16 21 18 30	6
1 2 5	4
2 4 7 7	
3 2 13	
1 1 8	
1 3 6	

【数据规模与约定】

样例点编号	N	M	N个数大小(\leq)	具有的操作
1, 2	100	100	100	1, 2, 3
3, 4	1000	1000	1000000	1, 2, 3
5, 6, 7	100000	100000	1000000	1, 3
8, 9, 10	100000	100000	1000000	1, 2, 3