

NNEZ 2020 入门测试

一、题目概况

题目名称	第一题	第二题	第三题
可执行文件名	magic	mode	query
输入文件名	magic.in	mode.in	query.in
输出文件名	magic.out	mode.out	query.out
时间限制	1秒	1秒	1秒
空间限制	256MB	4MB	512MB
测试点数量	20	20	20
单个测试点分值	5	5	5
题目类型	传统	传统	传统

二、提交源程序文件名

对于 C++ 语言	magic.cpp	mode.cpp	query.cpp

三、编译命令

对于 C++ 语言	-lm	-lm	-lm

注意事项

- 文件名（程序名和输入输出文件名）必须使用英文小写。
- C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时返回值必须是 0。
- 评测时采用的机器配置为 *Intel Core i3 - 7100@3.9GHz* 内存 *4GB*，上述时限以此配置为准
- 仅提供 *Linux* 格式附加样例文件
- 不建议使用万能头文件

考前提示

考试难度相对较难且时间较少，请合理安排考试时间。

对于题目有任何疑问的，可随时举手提出。

如果题目没有办法想出正解，请考虑拿部分分。

如果样例不足以满足测试的需求，可以自己手动写一个例子来检查。过了样例并不代表你这题可以得满分。

如果不会文件输入输出，可以参考以下模板：

```
#include<cstdio>
#include<iostream>
using namespace std;
int main()
{
    freopen("题目要求的输入文件名","r",stdin);
    freopen("题目要求的输出文件名","w",stdout);
    //中间的代码不需要改变 可以直接使用 cin cout

    fclose(stdin);
    fclose(stdout);
    return 0;
}
```

T1 magic

1.1 题目描述

给出 n 个数，与一个函数 $f(x) = x^3 + 13x^2 + 3x \pmod{20200920}$ ，你需要求出每个数对应的函数值。

1.2 输入描述

输入文件名为 *magic.in*。

输入文件包含 2 行。

第一行为一个整数 n ，即给出的数的个数。

接下来一行，为 n 个整数 a_i 。

1.3 输出描述

输出文件名为 *magic.out*。

输出文件中应包含 n 行。

一行一个整数 $f(a_i)$ ，即计算后的函数值。

1.4 输入样例 & 输出样例

magic.in	magic.out
3	66
2 15 17	6345
	8721

1.5 数据范围及提示

对于 50% 的数据，满足 $n \leq 10^3$ ， $1 \leq a_i \leq 120$

对于 80% 的数据，满足 $n \leq 10^4$ ， $1 \leq a_i \leq 1280$

对于 100% 的数据，满足 $n \leq 10^5$ ， $1 \leq a_i \leq 5000$

请注意算术符的优先级

T2 mode

1.1 题目描述

给出一个长为 n 的数列 a ，求 a 的众数。保证众数出现次数 $\geq \frac{n}{2}$ 。

1.2 输入描述

输入文件名为 *mode.in*。

输入文件包含 $n + 1$ 行。

第一行一个整数 n

接下来 n 行，每行一个整数 a_i 。

1.3 输出描述

输出文件名为 *mode.out*。

输出文件中应包含 1 行。

一个整数，即为众数。

1.4 输入样例&输出样例

mode.in	mode.out
5 2 3 3 3 3	3

1.5 数据范围及提示

对于 10% 的数据，满足 $n \leq 3$, $0 \leq a_i \leq 1000$

对于另 30% 的数据，满足 $n \leq 1000$, $0 \leq a_i \leq 1000$

对于另 20% 的数据，满足 $n \leq 1000$, $|a_i| \leq 2^{31} - 1$

对于 100% 的数据，满足 $n \leq 2000000$, $|a_i| \leq 2^{31} - 1$

请注意程序对内存的占用，根据测试，4MB 内存大约可以创建一个长度为 100000 的 *int* 数组。

T3 query

1.1 题目描述

已知一个数列，其中每一个数的初始值为 1000000。你需要维护以下操作。

- 1 将某区间 α 里的每一个数加上 k 。
- 2 将某区间 α 里的每一个数减去 k 。
- 3 求出某区间 α 每一个数的和。
- 4 令 A 表示某区间 α 每一个数的和， B 表示某区间 β 每一个数的和。求出 A 与 B 的最大公约数。

对最大公约数的求法如有疑问可以参见 1.6 题目补充。

1.2 输入描述

输入文件名为 `query.in`。

输入文件包含 $m + 1$ 行

第一行包含两个整数 n, m 分别表示数列的长度和操作总次数。

接下来 m 行每行包含 3 或 4 或 5 个整数，表示一个操作，具体如下：

- 1 x y k：见操作 1
- 2 x y k：见操作 2
- 3 x y：见操作 3
- 4 x y l r：见操作 4

1.3 输出描述

输出文件名为 `query.out`。

输出包括若干行，即所有操作 3 和 操作 4 的结果。

1.4 输入样例&输出样例

query.in	query.out
10 7 1 3 7 5 1 5 10 15 3 5 8 2 1 9 4 3 3 6 1 2 9 3 4 1 3 5 7	4000075 4000034 1

1.5 数据范围及提示

对于 20% 的数据, $n \leq 1000$ 且无操作 4。

对于 40% 的数据, $n \leq 1000$ 。

对于 60% 的数据, $n \leq 50000$ 。

对于 100% 的数据, $n \leq 100000$ 。

因为数据的原因, 请全程变量定义为 `long long`

注: 对于前 20 分, 仅用完成操作 1 2 3。对于前 40 分, 需要在前面的基础上完成操作 4。

1.6 题目补充

以下摘自人教版《数学·必修三》。

辗转相除法

.....辗转相除法中包含重复操作的步骤, 因此可以用循环结构来构造算法, 算法步骤如下:

第一步, 给定两个正整数 m, n 。

第二步, 计算 m 除以 n 所得的余数 r 。

第三步, $m = n, n = r$ 。

第四步, 若 $r = 0$, 则 m, n 的最大公约数等于 m ; 否则, 返回第二步。

更相减损数

.....翻译为现代语言如下:

第一步, 任意给定两个正整数, 判断它们是否都是偶数。若是, 用 2 约减; 若不是, 执行第二步。

第二步, 以较大的数减去较小的数, 接着把所得的差与较小的数比较, 并以大数减小数, 继续这个操作, 直到所得的数相等为止, 则这个数 (等数) 或这个数与约简的数的乘积就是所求的最大公约数。