

题目概况

中文题目名称	第一题	第二题	第三题
英文题目与子目录名	problem1	problem2	problem3
可执行文件名	problem1	problem2	problem3
输入文件名	problem1t.in	problem2 .in	problem3 .in
输出文件名	problem1.out	problem2 .out	problem3 .out
每个测试点时限	1s	1s	1s
测试点数目	10	10	10
每个测试点分数	10	10	10
附加样例文件	有	无	无
题目类型	传统	传统	传统
运行内存上限	512MB	512MB	256MB

第一题 (problem1.cpp/c/pas)

题目描述

给定一个正整数 n ，求有多少个大小为 n 的排列是偶排列。输出答案模 $109 + 7$ 的值。

其中，偶排列指的是逆序对数为偶数的排列。

输入格式

从文件 `problem1.in` 中读入数据。

输入一行，包含一个正整数 n 。

输出格式

输出到文件 `problem1.out` 中。

输出一行，按照以下格式输出答案：

```
The [nth] answer is [m].
```

其中，`[nth]` 应被替换为 n 在英文中的序数词，包含阿拉伯数字 n 紧跟着两个小

写英文字母；

[m] 应被替换为具体的答案。

你可以观察样例输出来确保对格式的正确理解。请. 注. 意. 行. 末. 有. 一. 个. 句. 号. 。

样例1输入

2

样例1输出

The 2nd answer is 1.

样例2输入

3

样例2输出

The 3rd answer is 3.

样例3输入

1000

样例3输出

The 1000th answer is 320709854.

子任务

对于 40% 的数据， $n \leq 10$

对于 100% 的数据， $1 \leq n \leq 10^5$

第二题 (problem2.cpp/c/pas)

题目描述

有 n 个初始为空的序列，按 1 到 n 编号。它们的元素是模 $10^9 + 7$ 意义下的整数。

请设计一种数据结构维护它们，支持以下六种操作：

- `1 i x`，其中 `i` 是一个序列的编号，`x` 是一个元素（即 `[0; 109 + 6]` 中的整数）。在序列 `i` 的左侧插入元素 `x`。
- `2 i`，其中 `i` 是一个序列的编号，保证此时序列 `i` 非空。删除序列 `i` 最左侧的元素，同时输出这个元素的值。
- `3 i x`。在序列 `i` 的右侧插入元素 `x`。
- `4 i`。删除序列 `i` 最右侧的元素，同时输出这个元素的值。
- `5 i d`。对于序列 `i` 的每一个元素 `x`，将它的值改为 `x + d`。
- `6 i j`，其中 `i, j` 是两个不同的序列编号。将序列 `j` 拼接至序列 `i` 的右侧，并清空序列 `j`。

输入格式

从文件 `problem2.in` 中读入数据。

第一行包含两个正整数 `n, m`，分别表示序列的数量和操作的数量。

接下来 `m` 行描述操作，符合题目描述中的格式，其中所有的编号都是 `[1; n]` 中的整数，所有值都是 `[0; 109 + 6]` 中的整数。

输出格式

输出到文件 `problem2.out` 中。

对于每一个操作 `2` 或操作 `4`，输出一行，包含一个 `[0; 109 + 6]` 中的整数，表示答案。

样例输入

```
2 10
1 1 100
3 1 200
3 2 10
1 2 20
5 2 40
6 1 2
2 1
2 1
2 1
2 1
```

样例输出

```
100
200
60
50
```

子任务

对于 前 40% 的数据， $1 \leq n, m \leq 100$ 。

对于 60% 的数据， $1 \leq n, m \leq 10^5$, 不存在操作6。

对于 100% 的数据， $1 \leq n, m \leq 10^6$ 。