

良心送分赛

zzq

2019.2

题目名称	直径	定价	排序
源程序文件名	diameter.c/cpp/pas	price.c/cpp/pas	sort.c/cpp/pas
输入文件名	diameter.in	price.in	sort.in
输出文件名	diameter.out	price.out	sort.out
时间限制	1s	7s	1s
是否捆绑测试	是	是	是
内存限制	512MB	512MB	512MB
是否有部分分	否	否	否
题目类型	传统	传统	传统
是否有附加文件	否	否	否
编译开关	-O2 -std=c++11	-O2 -std=c++11	-O2 -std=c++11

注意：

- 1.AK 了不要 D 出题人，没 AK 也不要 D 出题人。
2. 按照剧本所有题目时限均为 std 运行最大点用时两倍以上。
3. 题目顺序与难度无关。

Problem A. 直径 (diameter.c/cpp/pas)

Input file: diameter.in
Output file: diameter.out
Time limit: 1 second
Memory limit: 512 megabytes

上午讲了构造题，下午就放一道构造题。你需要构造一棵至少有两个顶点的树，树上的每条边有一个非负整数边权。树上两点 i, j 的距离 $dis(i, j)$ 定义为树上连接 i 和 j 这两点的简单路径上的边权和。

我们定义这棵树的直径为，所有满足 $1 \leq i < j \leq n$ 的 (i, j) 中， $dis(i, j)$ 最大的。如果有多个这样的 (i, j) ，那么均为直径。

作为一个构造题，你需要构造一个恰有 k 个直径的树。可以证明在给定的限制下一定有解。

Input

一行一个正整数 k ，表示你需要构造出一个恰有 k 个直径的树。

Output

第一行一个正整数 n ，表示你构造的树的点数。

接下来 $n - 1$ 行，每行三个整数 i, j, w ，表示一条连接点 i 和 j （点的编号为 $1, 2 \dots n$ ）的树边，边权为 w 。

Examples

diameter.in	diameter.out
3	5 1 2 2 3 2 2 2 5 3 4 2 2

这只是一种符合题意的输出，可能还有其他输出。在这个输出中，直径为 $(1, 5), (3, 5), (4, 5)$ 。

Notes

对于所有数据， $1 \leq k \leq 5000000$ 。

你构造的树需要保证 $2 \leq n \leq 5000$ ，且每条边的边权满足 $0 \leq w \leq 10^5$ 。

Subtask 1 (10pts): $1 \leq k \leq 5$ 。

Subtask 2 (20pts): $1 \leq k \leq 2000$ 。

Subtask 3 (30pts): $1 \leq k \leq 200000$ 。

Subtask 4 (20pts): $\sqrt{1 + 8k}$ 为整数。

Subtask 5 (20pts): 无特殊限制。

Problem B. 定价 (price.c/cpp/pas)

Input file: price.in
Output file: price.out
Time limit: 7 seconds
Memory limit: 512 megabytes

作为 ByteLand 中最出色的开发者，你刚刚开发了一款新游戏。

为了赚更多的钱，这个游戏的主线剧情分为 n 个 DLC 出售。你打算发行 n 种预购套装，第 i ($1 \leq i \leq n$) 种套装包含前 i 个 DLC。当然，第 i ($2 \leq i \leq n$) 个套装的价格需要严格高于第 $i-1$ 个套装的价格，并且第 1 个套装的价格大于 0。现在你需要确定这 n 个价格。

在 Byteland，所有物品的价格都只能为 m 位二进制非负整数。此外，政府限制了每个价格的某些二进制位必须为 0。为了体现你是良心开发商，你要求出所有 **套装** 的价格和的最小值。由于这个值可能太大了，你只需要输出它 $\text{mod } 10^9 + 7$ 后在十进制下的值。

天有不测风云，政策瞬息万变。一开始政府限制了每个价格的每个二进制位都必须为 0，后来随着时间推移，政府可能会允许某个价格的某个二进制位可以为 1，或重新限制某个二进制位必须为 0，你必须高效地对政策的变化作出反应，并随时支持求出价格和的最小值。

Input

第一行三个正整数 n, m, q ， q 表示操作个数。

接下来 q 行，每行为两者之一：

- 1 r c: 表示第 r 个套装的价格的从高到低第 c 位（即位权为 2^{m-c} 的一位）由于政策调整发生了变化。如果原来必须为 0 则现在可以为 1，否则现在必须为 0。
- 2: 表示询问所有 **套装** 的价格和的最小值，详见输出格式。

Output

对所有 2 操作输出一行：如果不存在合法的定价方案输出 -1 ，否则输出所有 **套装** 的价格和的最小值 $\text{mod } 10^9 + 7$ 后十进制下的值。

Examples

price.in	price.out
3 4 6 1 2 3 1 3 2 2 1 2 4 1 1 4 2	-1 7
1 1000000000 3 2 1 1 1 2	-1 570312504

Notes

对于所有数据， $1 \leq n \leq 1000$ ， $1 \leq m \leq 10^9$ ， $1 \leq q \leq 500000$ 。2 操作的个数不超过 1000。

Subtask 1 (10pts): $n = 1$ 。

Subtask 2 (15pts): $1 \leq n, m \leq 300$ 。

Subtask 3 (25pts): $1 \leq m \leq 10000$ 。

Subtask 4 (25pts): $1 \leq q \leq 5000$ 。

Subtask 5 (25pts): 无特殊限制。

由于输入输出量较大，下发文件中有 `io.cpp`，选手可以参考。注意评测时并不会附加该文件，如果你需要使用它可以直接拷贝其源代码。

Problem C. 排序 (sort.c/cpp/pas)

Input file: **sort.in**
Output file: **sort.out**
Time limit: 1 second
Memory limit: 512 megabytes

zzq 近日发明了一种最新的排序算法 **FastSort**，它的伪代码大致如下：

```
function FASTSORT( $a, n$ )  
   $cnt \leftarrow 0$   
  for  $i \leftarrow 1$  to  $n$  do  
    for  $j \leftarrow i + 1$  to  $n$  do  
      if  $a_j < a_i$  then  
         $tmp \leftarrow a_j$   
         $a_j \leftarrow a_i$   
         $a_i \leftarrow tmp$   
      end if  
       $cnt \leftarrow cnt + 1$   
    end for  
  end for  
  return  $a$   
end function
```

当然，这个伪代码里的 cnt 对排序来说没什么用，它只是用来指示这个算法的运行效果的。

zzq 很喜欢这个算法，于是他打算用它来给一个 $1, 2 \dots n$ 的排列 a 排序。为了清楚地获取算法的运行过程，zzq 决定在每次 cnt 改变时记录下当前的 cnt 和 a 便于分析。

zzq 运行了这个算法之后就睡觉去了。第二天早上，zzq 发现昨天停电了，日志里只剩下了记录下的最后一个 cnt 和 a 。zzq 不想再运行一遍算法了，于是他想让你确认一下记录是否正确。

一句话题意：对给定的 $1, 2 \dots n$ 的排列 a 执行算法 **FastSort**，问当 cnt 刚刚变成输入中给定的值时的 a 序列。

Input

第一行两个整数， n 和 cnt 。

第二行 n 个整数，表示 $a_1, a_2 \dots a_n$ 。保证是一个 $1, 2 \dots n$ 的排列。

Output

一行 n 个整数，表示 cnt 恰好变成给定值时的 a 序列。

Examples

sort.in	sort.out
5 3 4 3 2 5 1	2 4 3 5 1
9 16 1 9 8 2 3 7 5 4 6	1 2 8 9 3 7 5 4 6

Notes

对于所有数据， $2 \leq n \leq 10^6, 1 \leq cnt \leq \frac{n(n-1)}{2}$ 。

Subtask 1 (10pts): $cnt \leq 10^7$ 。

Subtask 2 (10pts): $a_i = n + 1 - i$ 。

Subtask 3 (20pts): 数据生成方式为，先选定 $n \leq 100000$ 和 cnt ，并初始化 $a_i = i$ ($\forall i \in [1, n]$)，然后重复 10 次，每次在 $[1, n]$ 中独立均匀随机两个不同的整数 i 和 j ，并交换 a_i 和 a_j 。

Subtask 4 (20pts): $n \leq 50000$ 。

Subtask 5 (20pts): $n \leq 200000$ 。

Subtask 6 (20pts): 无特殊限制。

由于输入输出量较大，下发文件中有 `io.cpp`，选手可以参考。注意评测时并不会附加该文件，如果你需要使用它可以直接拷贝其源代码。