

碎梦 (broken)

【题目背景】

「从我又小又拥挤的房间」
 「从我堆满了贪念的床边」
 「飞向总有容身处的世界」
 「却碎裂 却碎裂」

——「碎梦」COP，洛天依

多年以后，算法竞赛的一切对于忆哀来说是什么呢？一个光辉而没有结局的梦罢了。

永恒的梦，失落的梦，无望的梦，破碎的梦。

仿佛是想到了什么，她又从纸箱里拿出落了几层灰的笔记本，好不容易找出旧时代的充电插头。随着散热风扇的嗡鸣声，一个尘封的世界再次展现在她眼前，还觉得自己只要稍微看看问题，就有可能抽丝剥茧理清问题的思路，那只是逞强罢了。

“第二类斯特林数怎么求来着？”这个问题突然从她脑中浮现。

是啊，她曾经深深扎入各种组合计算中流连忘返，但如今已经连这些基础也不甚记得了。

【题目描述】

第二类斯特林数，是说将 n 个数划分进 m 个非空集合的方案数。我们接下来记为 $f(n, m)$ 。

忆哀的程序使用的是一个很朴素的递推式： $f(n, m) = f(n - 1, m - 1) + mf(n - 1, m)$ ，初值为 $f(0, 0) = 1, f(0, m) = 0 (m \neq 0)$ ，这个递推式的意义是不难解释的：要么第 n 个元素是自成一个集合，要么就将其分配给已有的 m 个集合之一。

忆哀的程序是这样的：

```
for (int i = 1; i <= n; ++i)
  for (int j = 0; j <= min(i, m); ++j)
    f[i][j] = (f[i - 1][j - 1] +
              j * (long long)f[i - 1][j]) % 998244353;
```

(你可以认为数组越界的部分值为 0，并且电脑开的下 $(n + 1) \times (m + 1)$ 的数组)

最后忆哀的程序理应输出 $f(n, m) \bmod 998244353$ ，但出了一个问題：由于种种原因，在计算完 $f(x, y)$ 后，内存的写入产生了意外，在内存中 $f(x, y)$ 被赋值为了一个数 z ，这样的事件对于不同的 (x, y) 总共发生了 k 次。

请你输出经过了这 k 次意外后，实际上忆哀的程序给出的 $f(n, m) \bmod 998244353$ 为多少。

【输入格式】

从文件 *broken.in* 中读入数据。

第一行输入三个整数 n, m, k ，意义如题目描述所示。

接下来 k 行每行输入三个数 x_i, y_i, z_i ，表示 $f(x_i, y_i)$ 在计算完成后实际写入的值为 z_i 。

【输出格式】

输出到文件 *broken.out* 中。

输出一个整数，表示计算得到的实际 $f(n, m)$ 结果。

【样例 1 输入】

```
5 3 1
1 0 1
```

【样例 1 输出】

```
31
```

【样例 2 输入】

```
1000 100 0
```

【样例 2 输出】

```
958221900
```

【样例 3】

见选手目录下的 *broken/broken3.in* 与 *broken/broken3.ans*。

【子任务】

对于 100% 的数据，保证 $1 \leq x_i \leq n \leq 9 \times 10^8, 0 \leq y_i \leq m \leq \min(n, 10^5), 0 \leq k \leq 20, 0 \leq y_i \leq x_i, 0 \leq z_i < 998244353, (x_i < x_{i+1}) \vee (x_i = x_{i+1} \wedge y_i < y_{i+1})$ 。

测试点	$n \leq$	$m \leq$	$k =$
1,2,3,4,5,6	10^3	500	20
7,8,9	9×10^8	10	
10,11		10^2	0
12,13,14		500	20
15,16,17			
18		10^5	0
19,20			20