1. 给定N个数字，每次从中取走2个数，然后计算它们的乘积；重复M次，求最后这M个乘积和的最小值。取走的数字不能被放回。

**输入：**

第一行输入两个整数N和M，N表示数字的个数，M表示进行游戏的轮次。

（2<=N<=100,000，0<=M<=N/2）

第二行为N个整数Ai，表示进行游戏的N个数字。

（0<=Ai<=10,000）

**输出：**

一个整数，表示乘积和的最小值。

示例1：

输入：

4 1

1 2 3 4

输出：

2

示例2：

输入：

4 2

1 2 3 4

输出：

10

1. 植树：一条路上有N个植树点，分别编号为1~N，而每个植树点上最多只能种一棵树。有M个小伙伴，第i个小伙伴一次种[Li, Ri]区间的树。问：每个小伙伴种完树以后，路上一共有多少棵树。

**输入：**

第一行输入两个整数N和M，N表示植树点数，M表示小伙伴的数量。

（1<=N<=100,000，1<=M<=100,000）

第二行后共有M行，每行有两个整数Li和Ri，表示第i个小伙伴负责种树的区间。

（1<=Li<=Ri<=N）

**输出：**

共M行，每行一个整数，表示第i个小伙伴种树后，路上一共有多少棵树。

示例1：

输入：

4 3

1 2

3 3

4 4

输出：

2

3

4

示例2：

输入：

4 3

1 2

2 3

3 4

输出：

2

3

4

备注：

对于50%的测试数据有：1<=N<=10,000, 1<=M<=10,000

对于100%的测试数据有：1<=N<=100,000, 1<=M<=100,000

1. 字典序的第K个数：一个字符串，里面所有单词都只由a和b组成，每个单词的组成里，a的数量不能超过N个，b的数量不能超过M个。

找出所有满足条件的单词里的字典序第K小的单词。

**输入：**

共一行，三个整数N, M, K。（0<N, M<50，0<K<1,000,000,000,000,000）

**输出：**

共一行，为字典序第K小的单词。

示例1：

输入：

2 1 4

输出：

ab

说明：

满足条件的单词里，按照字典序从小到大排列的结果是

a

aa

aab

ab

aba

b

ba

baa

备注：

对于40%的数据：0<K<100,000

对于100%的数据：0<K<1,000,000,000,000,000

题目保证第K小的单词一定存在。

1. 多多最近在玩一款游戏，游戏里一共有N个周末，每个周末玩家都有4种行动可以选择：散步、讲习、出击、修养。玩家每个周末只能选择一种行动，每种行动会获得一些不同维度的收益：提升伙伴支援值、提升人物好感度、提升伙伴干劲、提升自己的技能等。另外，其中有M个周末，玩家由于剧情安排只能选择某一种特定的行动。具体来说，对于第Wi（1<=i<=M, 1<=Wi<=N， Wi-1<Wi）个周末，玩家只能选择第Ci（1<=Ci<=4）种选择。

根据每周选择的行动不同，会形成不同的“游戏路线”。具体来说，序列A={a1, a2, a3, …, aN}(1<=ai<=4)构成了一个“游戏路线”。若两个序列A和B种存在一个整数k，使得ak!=bk，则认为序列A和B是2条不同的游戏路线。比如有4个周末，那么序列{1, 2, 1, 2}和序列{1, 3, 1, 2}不相同；{1, 2, 1, 2}和{2, 1, 2, 1}也不相同。多多觉得连续两个周末选择相同的行动是一件很无聊的事，所以一个游戏路线种存在连续2周都选择了相同的行动，则多多认为这个游戏路线**不快乐**。比如{1, 2, 2, 1}是一个不快乐的游戏路线，因为第2周和第3周这连续的2周都选择了相同的行动(2)；而{1, 2, 1, 3}是一个快乐的游戏路线。

现在多多想计算一下一共有多少种**不同的快乐的游戏路线**，请你帮助他。

**输入：**

输入的第一行是一个正整数T，表示测试用例的数量。接下来是T个测试用例。

每个测试用例的第一行包含2个整数N, M，分别表示游戏总的周末数和剧情安排的周末数。

第二行包含M个整数Wi（1<=i<=M, 1<=Wi<=N, Wi-1<Wi）：第三行包含M个整数Ci（1<=Ci<=4）。具体含义见题目描述。

数据范围：

1<=T<=100

对于20%的数据：1<=N<=106, M=0

对于60%的数据：1<=N<=106, 0<=M<=1000

对于100%的数据：1<=N<=1012, 0<=M<=1000

**输出描述：**

对于每个测试用例，输出一行，仅包含一个整数，表示有多少种不同快乐的游戏路线（对1,000,000,007取模后的结果）。

示例1：

输入：

3

3 0

10 0

1000 0

输出：

36

78732

75850924

示例2：

输入：

3

5 3

1 3 5

1 3 1

2 2

1 2

1 1

12 3

4 8 12

3 3 3

输出：

4

0

11907