Problem 8: Bay Area's Revolutionary Train
4+2+2=8 Points

Problem ID: `subway`

Rank: `2+2+3`

Bounty: A Limited Edition [BART 50th anniversary](#) [Clipper Card](#) _for the first team to solve any test set for this problem and fill out this [Google Form](#)!_

Introduction

The setup of this problem is very similar to `crosstown`! Key changes in `subway` are highlighted.

The year is [2048](#) and the [BART (Bay Area Rapid Transit)](#) has completed its long-awaited [Silicon Valley Extension](#)! They've also acquired [Caltrain](#), creating the [very first circular train line in the Bay Area](#)! Despite only one train car working, [CALICO's Biggest Railfan](#) excitedly rushes to film the occasion, but realizes he can't get any clear footage due to nonstop passenger flow at every single station. How long will it be until traffic finally clears up?

Problem Statement

There is a circular subway loop with M stations numbered 1 to M. There are N passengers numbered 1 to N spread across these stations. Each passenger begins at their starting station $S_1, S_2, \ldots, S_N$, and needs to go to their ending station $E_1, E_2, \ldots, E_N$. To get there, they board a single subway that goes around the loop that can carry at most K passengers.

The stations are arranged clockwise on a circle by their numbers in ascending order. The subway begins at station 1 and travels clockwise, stopping at each station along the way. After stopping at station M, it loops back to station 1.

At each station (including the initial station 1), passengers already on the subway whose E_i is the current station will exit the subway as they have arrived at their ending station. Then, passengers whose S_i is the current station will enter the subway in order of ascending passenger number i (smaller passenger numbers enter first) until there are no more passengers at the station or the number of passengers on the subway reaches the maximum of K. Then, the subway travels to the next station.

Each station is 1 mile apart. Find the total distance in miles the subway must travel until all passengers have arrived at their ending station.

Input Format

The first line of the input contains a single integer T denoting the number of test cases that follow. For each test case:

- The first line contains three space-separated integers N M K denoting the number of passengers, the number of stations, and the maximum number of passengers the subway can carry.
- The second line contains N space-separated integers S_1 S_2 ... S_N, denoting the starting station for each passenger in ascending order of passenger numbers.
- The third line contains N space-separated integers E_1 E_2 ..., E_N, denoting the ending station for each passenger in ascending order of passenger numbers.

Output Format

For each test case, output a single line containing an integer denoting the distance the subway must travel before all passengers arrive at their ending station.

*Careful! For the **second bonus test set only**, if you are a Java or C/C++ programmer, be aware that the int variable type may be too small to contain the final answer! Java programmers can use variable types long or float instead, and likewise long long or float for C/C++.*

Constraints

$1 \leq T \leq 100$
$1 \leq S_i, E_i \leq M$
$S_i \neq E_i$

Main Test Set

$1 \leq N, K \leq 10$
$2 \leq M \leq 10$

Bonus Test Set 1

$1 \leq N, K \leq 250$
$2 \leq M \leq 10^4$

Bonus Test Set 2

$1 \leq N, K \leq 10^5$
$2 \leq M \leq 10^9$
The sum of N across all test cases in a test file does not exceed 10^5.

Sample Test Cases

Sample Input   Download

```
6
1 6 1
3
5
2 6 2
1 4
5 6
2 6 1
1 4
5 6
2 8 2
2 3
6 5
4 5 2
1 3 3 5
4 5 1 2
7 7 1
1 1 1 1 1 1 1
7 7 7 7 7 7 7
```

**Sample Output**   Download

```
4
5
11
5
10
48
```

Sample Explanations

For test case #1, we have a single passenger starting at station 3 who wants to get to station 5. The subway starts at station 1, then travels to station 2, then station 3. The passenger gets on at station 3. Then, the subway travels to station 4, then station 5. The passenger gets off at station 5, and all passengers have now arrived. The total distance traveled was 4 miles.

For test case #2, the subway picks up a passenger at station 1, picks up another passenger at station 4, drops off a passenger at station 5, and drops off another passenger at station 6, for a total of 5 miles.

For test case #3, the subway picks up a passenger at station 1, stops at station 4 but is unable to pick up another passenger because the maximum number of passengers has been reached, drops off the passenger at station 5, then loops around to pick up and drop off the last passenger, for a total of 11 miles.

For test case #5, there are multiple passengers starting at station 3. We start with the sole passenger at station 1 boarding the train. When we get to station 3, the subway only has one seat left. The passenger that gets on is the one with $\mathbf{E}_i = 5$ because their passenger number is smaller than the other passenger. The second passenger in line waits for the subway to loop around. A timeline of events is shown below.
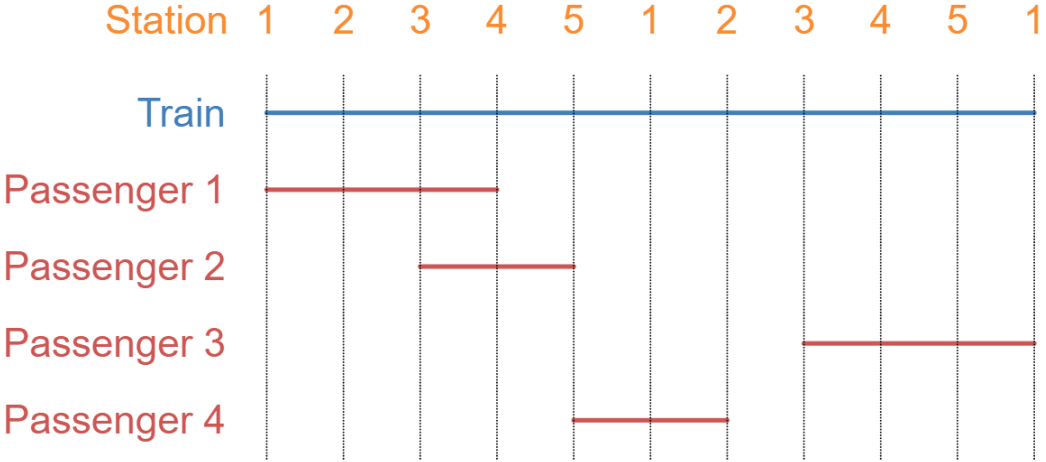
第 8 题：湾区铁路系统大改造
4+2+2=8 分

问题标识符：subway

难度等级：2+2+3

奖励： 首个通过此问题任意测试集并填写*此表格*的团队可赢得*湾区捷运系统限量版50周年纪念公交卡*！

问题背景

<mark>此题的背景设定与"crosstown"很相似！关键变化已高亮显示。</mark>

*2048*年，*旧金山湾区捷运系统*期盼已久的硅谷轨交延伸段竣工！同时它收购了*加州专列*，从而建成了湾区首条环形列车线路！虽然只有一辆列车投入运行，但CALICO的头号铁路粉丝迫不及待地要前去拍摄记录通车瞬间。但是由于每个站点都客流不止，他完全无法拍摄出清晰的照片。要过多长时间站台上才会没有乘客呢？

问题描述

一条环线地铁有 M 个站点，编号为 1 到 M 。这些站点中共有 N 位乘客，编号为 1 到 N。每位乘客在各自的起始站 S_1，S_2，...，S_N 搭乘环线地铁列车前往其下车站点 E_1，E_2，...，E_N。该列车<mark>最多可承载 K 位乘客</mark>。

所有站点按序号升序顺时针排列。地铁列车从站点 1 出发，顺时针行驶，途经每个站点停车。在站点 **M** 停车后，列车返回站点 1 再次开始环形运行。

在每一个站点（包括起始站点 1），其 E_i 为当前站点的在列车上的乘客会下车。然后，其 S_i 为当前站点的乘客会<mark>按照编号 i 升序的顺序上车（编号较小的乘客先上车），直到站内全部乘客上车或者列车达到最大承载量 K 时停止上车</mark>。地铁接着开往下一站点。

每一站点间隔 1 英里。<mark>请计算出所有乘客都到站下车时地铁总共行驶的距离，单位为英里。</mark>

输入格式

输入的第一行包含一个整数 T，表示测试用例的数量。

对于每一个测试用例：

- 第一行包括三个用空格隔开的整数 **N M K**，分别表示乘客数量、站点数量、<mark>地铁最大承载量</mark>。

- 第二行包含 **N** 个用空格隔开的整数 $S_1, S_2, ..., S_N$，表示按照乘客编号的升序，每位乘客的起始站点。

- 第三行包含 **N** 个用空格隔开的整数 $E_1, E_2, ..., E_N$，表示按照乘客编号的升序，每位乘客的下车站点。

输出格式

对于每一个测试用例，单独输出一行，包含一个整数，<mark>表示所有乘客都到达其下车站点时，地铁列车总共行驶的距离。</mark>

注意！仅对于第二个附加测试集，如果你是 Java 或 C/C++ 程序员，请注意 int 变量类型可能过小，不足以包含最终答案！Java 程序员可以改用 long 或 float 类型变量，C/C++ 程序员同样可以使用 long long 或 float。

数据范围

$1 \leq T \leq 100$

$1 \leq S_i, E_i \leq M$

$S_i \neq E_i$

主要测试集

$1 \leq N, K \leq 10$

$2 \leq M \leq 10$

附加测试集1

<mark>$1 \leq N, K \leq 250$</mark>

<mark>$2 \leq M \leq 10^4$</mark>

附加测试集2

<mark>$1 \leq N, K \leq 10^5$</mark>

<mark>$2 \leq M \leq 10^9$</mark>

<mark>一个测试文件中所有测试用例的 N 之和不超过 10^5.</mark>

测试样例

```
6
1 6 1
3
5
2 6 2
1 4
5 6
2 6 1
1 4
5 6
2 8 2
2 3
6 5
4 5 2
1 3 3 5
4 5 1 2
7 7 1
1 1 1 1 1 1 1
7 7 7 7 7 7 7
```

```
4
5
11
5
10
48
```

主样例解释

对于测试用例#1，一位乘客从地铁站点 3 上车，想在站点 5 下车。地铁列车从站点 1 出发，经过站点 2，到达站点 3 时，该乘客上车。随后，地铁开往站点 4，到达站点 5 时，该乘客下车。此时所有乘客到站。总行驶距离为 4 英里。

对于测试用例#2，一位乘客在地铁站点 1 上车，另一位乘客在站点 4 上车。随后，一位乘客在站点 5 下车，另一位乘客在站点 6 下车。总行驶距离为 5 英里。

对于测试用例#3，一位乘客在地铁站点 1 上车。列车接下来在站点 4 停车，但由于地铁已达到最大承载量，因此无法搭载另一位乘客。站点 1 上车的乘客在站点 6 下车，随后地铁再次环形运行接送最后一位乘客，总行驶距离为 11 英里。

对于测试用例#5，多位乘客需要在站点 3 上车。站点 1 有一位乘客上车。在到达站点 3 时，车内仅剩 1 个座位。此时上车的是 $E_i = 5$ 的乘客，因为他的编号比其他等待上车的乘客的小。其他乘客则需要等待地铁再行驶一圈回来上车。下图为整个行程的时间线。