1 RTC Cargo Services

A state road transport corporation is planning to introduce cargo bus services. The state has identified a few towns that have need for such cargo services.

Each town has a Special Bus Terminal (*SBT*) where cargo is loaded and unloaded. Each SBT has several buses operating as part of cargo services. Here are a few operating rules:

- All towns are interconnected by single, bidirectional, no self-loop routes.
- There would be ***N towns (1 to N)*** and ***M undirected routes***.
- Every SBT shall operate in two modes. (1) *Busy (2) Transit*. In *Busy Mode,* cargo is loaded into the cargo buses and entry for cargo buses coming from other town is blocked and they have to wait till the *Busy Mode* is switched to *Transit Mode*. In *Transit Mode*, cargo buses will be leaving the current SBT for other cities.
- Every ***S hours*** the operating mode of SBTs is **s**witched from *Busy Mode to Transit Mode* and vice versa. *Note: Initially all SBTs will be Transit Mode only.*
- It could take ***T hours*** to move the cargo from one town to another town.
- The primary focus should be on saving fuel, on-time delivery of cargo.

Given information about the number of towns, routes, switching time and time taken to move from source to target, find the lexicographically smallest path which will take the minimum amount of time (in hours) required to move from town-***X*** to town-***Y*** and *print two lines* of output. In the first line, print *total number of towns* that a bus must go through to reach town-Y from town-X and in the next line *print the town-ids* (separated by spaces) denoting the path that takes the minimum amount of time (in hours) required to move from town-X and town-Y.

Input/Output

Input	Output	Comments
6 8 2 4 **1 2** **2 3** **2 4** **2 6** **2 6** **4 5** **5 6** **1 5** **1 6**	**3** **1 2 6**	• The first line contains 4 space separated integers, N, M, S and T. • Next M lines contains two space separated integers each, U and V denoting that there is a bidirectional road between town U and town V. • Next line contains two space separated integers, X and Y. Explanation: • Possible routes : ○ 1->2->4->5->6 ○ 1->2->3->5->6 ○ **1->2->6** ○ 1->5->6 • Fastest path will be **1–>2–>6**. (lexicographically smallest)

		• Reach town-2 in 4 hours. After 4 hours at the SBT in town-2 will change its operating state to Busy and then to operating. So in town-2, no need to wait for the SBT to change its operating state. • So total time will be 4 hours (from town-1 to town-2) + 4 hours (from town-2 to town-6) = **8**
5 5 3 5 **1 2** **1 3** **2 4** **1 4** **2 5** **1 5**	**3** **1 2 5**	• The first line contains 4 space separated integers, N, M, S and T. • Next M lines contains two space separated integers each, U and V denoting that there is a bidirectional road between town U and town V. • Next line contains two space separated integers, X and Y. • Fastest path will be 1–>2–>5. • Reach town-*2* in *5* hours. After *3* hours at the SBT in town-2 will change its operating state to Busy. So in town-2, wait for *1* hour for the SBT to change its operating state. • So total time will be ***5*** hours (from town-1 to town-2) + ***1*** hour (waiting time at town-*2*) + ***5*** hours (from town-2 to town-5) = 11 hours.

2 Haandi breaking competition

In a school, Sri Krishna Shishu Mandir, during Janmaastami, a haandi (pot) breaking competition was organized for kids to break the pots and win chocolates. For the competitions, **N** pots are hanged in row (imagine as an array ***pots[]***). *So, the first pot is at **pots[0]** and last pot in the row would be **pots[N-1].*** **pots[i]** is labelled with number indicating its worth.

The rules of the competition are as follows:

- Every time a pots[i] is broken then the kid will get chocolates equal to products of worth of **pots[i-1] * pots[i] * pots[i+1].**
- After a pots[i] is broken, it is removed and the total number of pots is reduced by one and pots[i-1] and pots[i+1] become adjacent pots.
- If there are no three pots then we may imagine pots[-1] = pots[n] = **1**. They are not real therefore you cannot break them.
- All pots in the row must be broken.

Now, Shyamlal likes chocolates very much so he wants to earn more chocolates in the competition. Given the number of pots and their worth, find out the *maximum number of chocolates* that Shyamlal can win by breaking the pots more intelligently.

Input/Output

Input	Output	Comments
4 3 1 5 8	**167**	• First line 4 corresponds to number of pots • Second line 3 1 5 8 corresponds to worth of each pot. *Round-1* → *Round-2* → *Round-3* → *Round-4* • pots = [3 **1** 5 8] → [3 **5** 8] → [**3** 8] → [**8**] → [] • chocolates = 3*1*5 + 3*5*8 + **1***3*8 + **1***8***1** = **167**

3 Vinta Guha, a Novel Cave

In Vinaynagara kingdom, Devaraya has ordered his education minister to come up with a wonderful amusement park for kids to play. The education minister responded promptly and constructed a beautiful park with lots of funny games.

One of the games was 'Vinta Guha' (Novel Cave). That cave had compartments that form a N x N grid. Each compartment was either *empty (0), mango (1) or closed (-1).* The main task of the game was to enter into the cave and win as many mangoes as possible. There were some interesting rules to get those mangoes from those compartments of the grid.

The rules are as follows:

1. One can pass through an empty compartment.
2. The compartment with a mango becomes empty once the mango is picked up.
3. Compartments either empty or with mangoes are considered valid to pass through.
4. First compartment (0, 0) and Last compartment (N-1, N-1) are always either empty (0) or have mangoes (1). They are never closed (-1)
5. Always start from first compartment (0, 0), move right or down and reach last compartment (N-1, N-1). In way, pick up mangoes (if found) and make that compartment empty.
6. On reaching last compartment (N-1, N-1), resume the journey back to first compartment (0, 0) by moving left or upper valid compartments.
7. Repeat (5) and (6) till is it impossible to pass through compartments due the rules of movement as defined in (5) and (6).

Given the grid, find out the maximum number of mangoes that a kid can win. Print 0 if there is no path from start point to end point.

Input/Output

Input	Output	Comments
3 0 1 -1 1 0 -1 1 1 1	5	• The first line 3 corresponds to the size of the grid. • The player starts at (0, 0) and goes *down, down, right, right* to reach (2, 2). • 4 mangoes were picked up during this single trip, and the matrix becomes [[0 1 -1], [0 0 -1], [0 0 0]]. • Then, the player can go *left, up, up, left* to return home, picking up one more mango. • The total number of mangoes picked up is 5, and this is the maximum possible.
3 1 -1 1 1 -1 0 0 1 1	4	• The player starts at (0, 0) and goes *down, down, right, right* to reach (2, 2). • 4 mangoes were picked up during this single trip, and the matrix becomes [[0 -1 1], [0 -1 0], [0 0 0]].

		• There is a chance of blocking if he tried to pick a mango placed on (0, 2) position while returning home (0, 0) he can't move down and left there is a closed compartment. • So, the player goes *left, left, up, up* to return home without picking any mango. • The total number of mangoes picked up is **4**, and this is the maximum possible.
		•