



Task 2: Journey

Kuno Kunibertson is moving house from Amman to Brasília. His new employer pays the trip and allows him to make as many stop-overs as he wants; the only two conditions are: (1) the i -th stop-over city of the trip should be nearer to Brasília than the $(i - 1)$ -th stop-over city and (2) the trip should not take more than m nights. Furthermore, as Brasília is a transport hub, there is a direct flight from each city to Brasília. Kuno Kunibertson has to supply the following information to the trip evaluation:

1. Three integers n, m, h where n is the number of cities which might be visited on the trip, m is an upper bound on the number of nights the trip can take and h is the number of follow-on flights per stop-over city (for the ease of notation, the cities are numbers $0, 1, \dots, n - 1$ where 0 is Amman and $n - 1$ is Brasília).
2. For each stop-over city, a list of exactly h possible follow-on flights and each such flight together is specified by the next city together with a minimum number of nights for a hotel stay at the next city. Only those flights which go nearer to Brasília are relevant, the others should be ignored by the program.

The trip evaluation agency will then plan out a suitable trip for him with some artificial intelligence program, which optimises the trips according to a data base of tourist attractions. However, in order not to overload its computers, the trip evaluation agency has the following requirement:

(*) The number of itineraries which use up k nights ($k < m$) should not be more than 500000000 (that is, 5×10^8).

Note that for counting, two itineraries are considered to be different, if one of the following conditions applies:

- one itinerary contains the stopover in city i and the other one does not;
- the departure day from city i is different in both itineraries;
- both itineraries go from city i to city j , but using different flights (this can be the case if Kuno Kunibertson lists two or more flights from city i to city j in his data).

For example, if the itinerary is Amman - Cairo - Dakar - Brasília, then an itinerary which departs from Cairo on day 2 differs from another itinerary which departs from Cairo on day 3. Furthermore, if there are two flights from Dakar to Brasília (even if they depart and arrive on the same days), the itineraries are different since they use these different flights.



As there is a penalty when the condition (*) is violated, Kuno Kunibertson asks his friend Herbert Hercules to write for him a computer program which computes the number of itineraries which use up to k days.

Input format

Your program must read from standard input.

The first row of the input consists of three integers n , m and h , where n is the number of cities, m is the strict upper bound of the number of nights used and h is the number of follow-on flights per city. We assume city 0 is Amman and city $n - 1$ is Brasília. We also assume city i is closer to Brasília than city j if $i > j$.

Then, the input contains $n - 1$ rows, where the i th row, $i = 0, 1, \dots, n - 2$, consists of h pairs of integers (j, k) , which means that Kuno Kunibertson can fly from city i to city j and stays at least k nights. If there are several flights to the same city, they are considered different. All $2h$ integers are listed in one row separated by spaces. Below is an example.

```
4 4 3
1 2 2 2 3 0
2 2 2 3 3 0
3 1 3 3 3 0
```

Here city 0 stands for Amman, city 1 might stand for Cairo, city 2 for Dakar and city 3 for Brasília. So the data stands for the following information.

City	Flight a To City	Min. Stay	Flight b To City	Min. Stay	Flight c To City	Min. Stay
Amman	Cairo	2	Dakar	2	Brasília	0
Cairo	Dakar	2	Dakar	3	Brasília	0
Dakar	Brasília	1	Brasília	3	Brasília	0

Output format

Your program must output to standard output only.

The output is a list of m integers between 0 and 500000001 which are, respectively, the number of itineraries from Amman to Brasília taking 0, 1, \dots , $m - 1$ nights, respectively. If the number of itineraries is 500000001 or more, the output should be 500000001.

The correct answer for the above input example is as follows:



1 1 3 6

Explanation

The itineraries are the following, where flight a is the first, flight b the second and flight c the third flight out of a city; the overall number of itineraries is 11.

Total Nights	Itinerary	Day	From	To	Flight	Nights at Next City
0	0	0	0	3	c	0
1	0	0	0	3	c	1
2	0	0	0	1	a	2
		2	1	3	c	0
	1	0	0	2	b	2
		2	2	3	c	0
2	0	0	3	c	2	
3	0	0	0	1	a	2
		2	1	3	c	1
	1	0	0	1	a	3
		3	1	3	c	0
	2	0	0	2	b	2
		2	2	3	a	1
	3	0	0	2	b	3
3		2	3	c	0	
4	0	0	2	b	2	
	2	2	3	c	1	
5	0	0	3	c	3	

Subtasks

There are four subtasks; each subtask will be tested on several instances and the maximum execution time on each instance is 1.0 seconds. The subtasks and their marks are as follows.

Subtask	Marks	Criteria
1	20	at most one stop-over, $n, m, h \leq 10$
2	23	no minimum-stay, $n, m, h \leq 20$
3	26	$n, m, h \leq 100$
4	31	$n \leq 10000, m \leq 400, h \leq 100$



At most one stop-over means that all outgoing flights from a stop-over location go directly to Brasília. No minimum stay means that the stay in each stop-over can be any number (including 0) and that Kuno Kunibert can report to work on arrival in Brasília.

Sample Testcase 1

This testcase consists of test data with at most one stop over. It is valid for subtasks 1, 3 and 4.

Input	Output
4 8 3 1 0 2 1 3 0 3 1 3 2 3 0 3 1 3 1 3 0	2 5 11 17 23 29 35 41

Sample Testcase 2

This testcase consists of test data with no minimum stay. It is valid for subtasks 2, 3 and 4.

Input	Output
4 11 3 1 0 2 0 3 0 0 0 2 0 3 0 0 0 0 0 3 0	4 8 13 19 26 34 43 53 64 76 89

Sample Testcase 3

This testcase consists of test data which generate large output numbers so that the rule comes into place which limits the maximum of an output number to 500000001. The test case is valid for subtasks 2, 3 and 4.



Input	Output
8 8 8	960800 6702725 26746084 80092734
1 0 1 0 1 0 1 0 1 0 1 0 1 0 7 0	199948848 439388874 500000001
2 0 2 0 2 0 2 0 2 0 2 0 2 0 7 0	500000001
3 0 3 0 3 0 3 0 3 0 3 0 3 0 7 0	
4 0 4 0 4 0 4 0 4 0 4 0 4 0 7 0	
5 0 5 0 5 0 5 0 5 0 5 0 5 0 7 0	
6 0 6 0 6 0 6 0 6 0 6 0 6 0 7 0	
7 0 7 0 7 0 7 0 7 0 7 0 7 0 7 0	

Note that artificial linebreaks have been inserted into the output for this example to avoid an overlong line. These linebreaks are for display-purposes only and not required in the output.