Problem A. Um_nik's Algorithm

Input file:	*standard input*
Output file:	*standard output*
Time limit:	4 seconds
Memory limit:	512 mebibytes

Can you replicate my bachelor thesis in 5 hours?

I give you an undirected bipartite graph. Let K be the size of its maximum cardinality matching. Devise an *algorithm* to find a matching of size at least $0.95 \cdot K$.

If you want to get Accepted, I suggest you to optimize your code as good as you can.

Input

The first line contains three positive integers n_1, n_2 and m ($1 \le n_1, n_2, m \le 2 \cdot 10^6$) — the number of vertices in the first part, the number of vertices in the second part and the number of edges in the graph, respectively.

The next m lines describe edges, one per line. Description of each edge is two integers u and v ($1 \le u \le n_1$, $1 \le v \le n_2$) — the ids of vertices in first and second parts that are connected by the edge. There is no pair of edges connecting the same vertices.

Output

In the first line print one integer L — the size of the matching you found. The inequality $0.95 \cdot K \le L$ should hold.

In the next L lines print the ids of the edges in your matching. Edges are numbered from 1 to m in the order they are given in input.

Examples

standard input	standard output
3 2 4 1 1 2 1 3 1 3 2	2 1 4
20 20 20 1 1 2 2 3 3 4 4 5 5 6 6 7 7 8 8 9 9 10 10 11 11 12 12 13 13 14 14 15 15 16 16 17 17 18 18 19 19 20 20	19 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19