Henry and Hetty were recently hired by a networking company from Piatra Neamț. Their first project is to create a new type of router, the revolutionary *Connect Ethernet Operating Interface 2016*, comprised of:

- N input nodes, numbered from 1 to N;
- N output nodes, numbered from N+1 to 2*N;
- K internal nodes, numbered from 2*N+1 to 2*N+K;
- M **unidirectional** direct connections between pairs of distinct nodes.

A node X can send data to a node Y (and hence Y can receive data from X) if:

- X = Y, or
- there exists a node Z such that X can send data to Z and there is a direct connection from node Z to node Y.

If a node X can send data to a node Y, and $X \neq Y$, then we define a data path from X to Y as a set of direct connections $\{(A_1, A_2), (A_2, A_3), \ldots (A_{L-1}, A_L)\}$ for some $L \geq 2$, such that $A_1 = X$ and $A_L = Y$.

A router works properly if:

- Each input node can send data to each output node;
- Each input node can receive data only from itself;
- Each output node can send data only to itself;
- For any two nodes X and Y, if $X \neq Y$ and X can send data to Y, then Y cannot send data to X;
- For any two nodes X and Y, if $X \neq Y$ and X can send data to Y, then **the data path from X to Y is unique**. In particular, any two nodes X and Y should be connected by at most 1 direct connection.

Like any other electronic device, a router needs electricity to work. Let's define the power needed to operate a node X as $P_X = IN_X * OUT_X$, where IN_X is the number of input nodes that can send data to X, and OUT_X is the number of output nodes that can receive data from X. Let's define the maximum power used by the router as $P_{max} = max(P_1, P_2, \ldots P_{2*N+K})$.

The project manager has given Henry and Hetty the technical specifications for building a few test routers, listed in the table below. For each of these specifications, the manager wants a router which:

- has exactly N input nodes and N output nodes;
- uses at most M_{lim} direct connections;
- uses a maximum power which is at most P_{lim};
- uses at most 500 000 nodes overall (Total nodes = $N_{tot} = 2*N + K \leq 500\ 000$).

Test number	N	M_{lim}	P_{lim}	Score
1	118	1 000 000	1 000 000	4
2	223	1 000 000	1 000 000	5
3	1250	500 000	500 000	6
4	5101	500 000	500 000	6
5	9934	500 000	500 000	26
6	9955	500 000	100 000	30
7	9978	100 000	100 000	23

For each router they successfully build, Henry and Hetty will receive a certain score, listed in the table.

Description of input

You are not supposed to submit the program that solves the described task. Instead, in the archive you download from the grading system, you will find the files `1-router.in, 2-router.in ... 7-router.in`. The files are the input data for each of the tests. You can use the command `unzip router.zip` to extract the input files from the zip archive.

Each of the input files `1-router.in, 2-router.in ... 7-router.in` describes a single test. Each file contains, on the first and only line, three integers: `N`, the number of input and output nodes; M_{lim}, the maximum number of direct connections allowed; and P_{lim}, the maximum power the router uses.

Description of output

For each input file, you should create the corresponding output file `1-router.out, 2-router.out ... 7-router.out`. Place all these files in a directory called `router-out` and create a zip archive containing this directory. You can use the command `zip –r router-out.zip router-out` to create an archive `router-out.zip`. You should submit this archive as the solution.

In each of the output files `1-router.out, 2-router.out ... 7-router.out` you will output two integer numbers separated by a space: $N_{tot} = 2*N + K$, representing the total number of nodes used to build the router; and `M`, representing the total number of direct connections used. On each of the following `M` lines you should output a pair of integers `X` and `Y`, meaning a direct connection from node `X` to node `Y` was built.

Helper scripts

In the archive you download from the grading system you will also find two scripts, `gen-out.sh` and `check.sh`, and an executable called `verif_contestant`. If you place these three files, the input files and an executable named `router` in the same directory, you can use the command `bash gen-out.sh` to generate the output files produced by your executable for each of the input files. You can then use the command `bash check.sh` to test the correctness of your outputs relative to the task and input constraints. The executable `router` should be produced from a source file you created which reads the input data from a file called `router.in`, and prints the output to a file called `router.out`.

Examples

router.in	router.out	Explanation
3 100 200	9 8 1 7 2 7 3 8 7 8 8 4 8 9 9 5 9 6	Henry and Hetty must build a router with 3 input nodes and 3 output nodes, which uses at most 100 direct connections and uses a maximum power of 200. They use 9 nodes overall (input nodes 1, 2 and 3, output nodes 4, 5 and 6, and internal nodes 7, 8 and 9), and 8 direct connections. The maximum power the router uses is 9 and is given by the power of vertex 8, which can receive data from $IN_8=3$ input nodes and can send data to $OUT_8=3$ output nodes.

3 100 200	6 9 1 4 1 5 1 6 2 4 2 5 2 6 3 4 3 5 3 6	Another valid router for the same specifications uses 6 nodes overall (the three input nodes and the three output nodes). The maximum power this router uses is 3: each input node can receive data only from itself and can send data to all three output nodes. Similarly, each output node can receive data from all 3 input nodes and can send data only to itself.