CODEFORCES$^{\beta}$
Sponsored by Telegram

Pinely Treasure Hunt Contest

A. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \le t \le 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \le u, v \le n$, $u \ne v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \dots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i $(1 \le i \le d)$, which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution

1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \leq base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \leq 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

B. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \le t \le 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \le u, v \le n$, $u \ne v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i $(1 \le i \le d)$, which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	

1
AC

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

C. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t ($1 \le t \le 5$) — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ ($2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000$) — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v ($1 \le u, v \le n$, $u \ne v$). It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \dots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \leq base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \leq 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

D. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \leq t \leq 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \leq n \leq 300$, $1 \leq m \leq min(\frac{n(n-1)}{2}, 25n)$, $1 \leq start \leq n$, $1000 \leq base\_move\_count \leq 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \leq u, v \leq n$, $u \neq v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i $(1 \leq i \leq d)$, which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

E. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t ($1 \le t \le 5$) — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ ($2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000$) — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v ($1 \le u, v \le n$, $u \ne v$). It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1

R 2 2 0 2 1
2
R 2 2 0 2 1
1
AC

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

F. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t ($1 \le t \le 5$) — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ ($2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000$) — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v ($1 \le u, v \le n$, $u \ne v$). It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

G. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \le t \le 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \le u, v \le n$, $u \ne v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

H. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t ($1 \le t \le 5$) — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ ($2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000$) — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v ($1 \le u, v \le n$, $u \ne v$). It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	

2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

I. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \le t \le 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \le u, v \le n$, $u \ne v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i $(1 \le i \le d)$, which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

J. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t $(1 \le t \le 5)$ — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ $(2 \le n \le 300$, $1 \le m \le min(\frac{n(n-1)}{2}, 25n)$, $1 \le start \le n$, $1000 \le base\_move\_count \le 30000)$ — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v $(1 \le u, v \le n$, $u \ne v)$. It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \le i \le d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution
1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.

- if $base\_move\_count < moves \leq 2 \cdot base\_move\_count$, you get $20 - \frac{10 \cdot (moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.

K. Treasure Hunt

time limit per test: 5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

All problems in this contest share the same statement, the only difference is the test your solution runs on. For further information on scoring please refer to "Scoring" section of the statement.

This is an interactive problem.

Imagine you are a treasure hunter, a very skillful one. One day you came across an ancient map which could help you to become rich. The map shows multiple forestry roads, and at each junction there is a treasure. So, you start your journey hoping to retrieve all the hidden treasures, but you don't know yet that there is a wicked wizard standing against you and craving to tangle up these roads and impede your achievements.

The treasure map is represented as an undirected graph in which vertices correspond to junctions and edges correspond to roads. Your path begins at a certain fixed vertex with a label known to you. Every time you come to a vertex that you have not been to before, you dig up a treasure chest and put a flag in this vertex. At the initial vertex you'll find a treasure chest immediately and, consequently, you'll put a flag there immediately as well.

When you are standing at the junction you can see for each of the **adjacent** vertices its **degree** and if there is a **flag** there. There are no other things you can see from there. Besides, the power of the wicked wizard is so great that he is able to change the location of the roads and junctions on the map without changing the graph structure. Therefore, the sequence of the roads coming from the junction v might be different each time you come in the junction v. However, keep in mind that the set of adjacent crossroads does not change, and you are well aware of previously dug treasures at each adjacent to v vertex.

Your goal is to collect treasures from all vertices of the graph as fast as you can. Good luck in hunting!

Interaction

On the first line the interactor prints an integer t ($1 \leq t \leq 5$) — the number of maps for which you need to solve the problem.

Then for each of the t maps interactor firstly prints the graph description and after that interaction on this map starts. The map description is given in the following format.

The first line contains four integers $n, m, start, base\_move\_count$ ($2 \leq n \leq 300$, $1 \leq m \leq min(\frac{n(n-1)}{2}, 25n)$, $1 \leq start \leq n$, $1000 \leq base\_move\_count \leq 30000$) — the number of vertices and edges in the graph, the label of the vertex where you start the journey and some base move count, on which the score for the map depends. It's guaranteed that the jury has a solution traversing the map using not more than $base\_move\_count$ moves with high probability.

The next m lines contain descriptions of the edges of the graph u, v ($1 \leq u, v \leq n$, $u \neq v$). It is guaranteed that the graph is connected, does not contain multiple edges and the degrees of all vertices do not exceed 50.

You can find map descriptions for all tests in the archive 'map_descriptions.zip' which you can find in any of the archives 'problem-X-materials.zip' in the section "Contest materials" — they are all the same.

After that the interaction begins. The interactor prints vertex descriptions in the following format:

$$R\ d\ deg_1\ flag_1\ deg_2\ flag_2 \ldots deg_d\ flag_d,$$

where d is the degree of the current vertex, deg_i is the degree of the i-th vertex adjacent to the current one, and $flag_i$ (0 or 1) is an indicator if the i-th adjacent vertex contains a flag. The order of neighbors in the vertex description is chosen by the interactor uniformly at random independently each time. Please keep in mind that you are not given the actual labels of adajacent vertices.

In response to the description of the vertex you should print a single integer i ($1 \leq i \leq d$), which means that you have chosen the vertex with description $deg_i\ flag_i$. Remember to use the flush operation after each output. If your output is invalid, you will get a "Wrong Answer" verdict.

When you have visited all the vertices of the graph at least once, the interactor prints the string "AC" instead of the vertex description. If you use more than $2 \cdot base\_move\_count$ moves, the interactor prints the string "F". In both cases, you have to either start reading the description of the next map, or terminate the program, if it has been the last map in the test.

Below the example of interaction is presented. Note that the test from the example is not the same that your solution will be tested on.

Interactor	Solution

1	
3 3 1 1000	
1 2	
2 3	
3 1	
R 2 2 0 2 0	
	1
R 2 2 0 2 1	
	2
R 2 2 0 2 1	
	1
AC	

Scoring

All problems of this contest share the same statement and differ only in the test. Each problem contains one test. Map descriptions for the test are available to look at in the archive.

Each test consists of several maps.

The solution score for the problem will be 0 in case your solution for some map failed to get an answer "AC" or "F". If the solution correctly interacted with the interactor on all t maps, the score for the task will be equal to the sum of the scores for each of the maps.

If you have successfully passed all vertices of a graph with n vertices using $moves$ moves, the score for the map is calculated as follows. Denote

$$base\_fraction = \frac{base\_move\_count + 1}{n},$$

$$sol\_fraction = \frac{moves + 1}{n},$$

$$c = \frac{90}{\sqrt{base\_fraction - 1}}.$$

Then:

- if $moves \le base\_move\_count$, you get $100 - c\sqrt{sol\_fraction - 1}$ points.
- if $base\_move\_count < moves \le 2 \cdot base\_move\_count$, you get $20 - \frac{10\cdot(moves+1)}{base\_move\_count+1}$ points.

If you use more than $2 \cdot base\_move\_count$ moves, you get 0 points for the map.
For each problem the solution with the **highest** score is chosen. Please note that the maximum is chosen for the entire test as a whole, not for each separate map.

The final result of the participant is the sum of points for each of the problems.