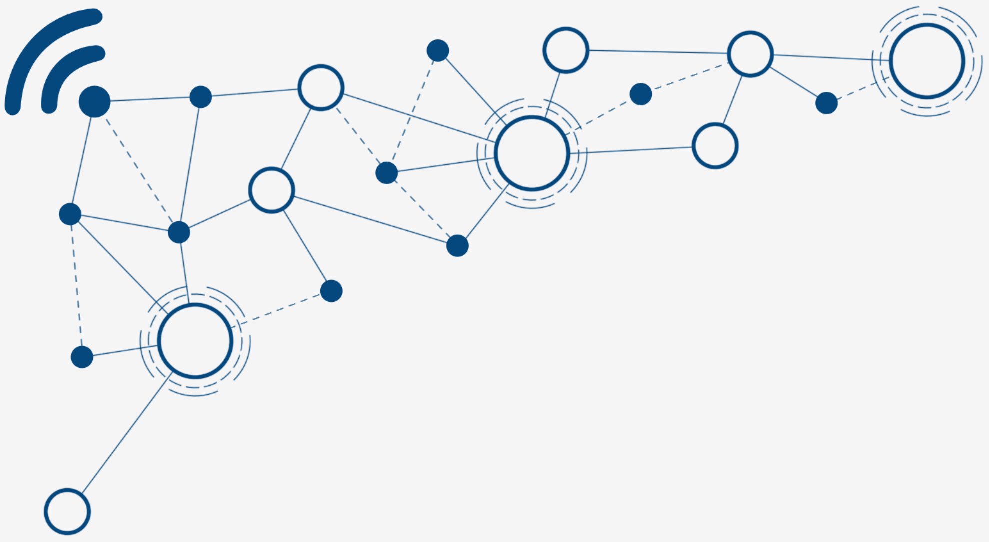


NOI.PH 2025

National Olympiad in Informatics

Finals Round 2

Important! Read the following:

Hidden Test Cases. Your solution will be checked by running it against one or more (usually several) hidden test cases. You will not have access to these cases, but a correct solution is expected to handle them correctly.

Strict Output Format. The output checker is **strict**. Follow these guidelines strictly:

- It is **space sensitive**. Do not output extra leading or trailing spaces. Do not output extra blank lines unless explicitly stated.
- It is **case sensitive**. So, for example, if the problem asks for the output in lowercase, follow it.
- Do not print any tabs. (No tabs will be required in the output.)
- Do not output anything else aside from what's asked for in the Output section. So, do not print things like "`Please enter t`".

Not following the output format strictly and exactly will likely result in the verdict "*Output isn't correct*".

Use Standard I/O. Do not read from, or write to, a file. You must read from the standard input and write to the standard output.

Submit Code Only. Only include **one** file when submitting: the source code (.cpp, .py, etc.) and nothing else.

No Java Package. For Java submissions, do not include a `package` line.

No Weird Filenames. Only use letters, digits and underscores in your filename. Do not use spaces or other special symbols.

Use Fast I/O. Many problems have large input file sizes, so use fast I/O. For example:

- In C/C++, use `scanf` and `printf`.
- In Python, use `sys.stdin.readline()`

Flush On Interactive Problems. On interactive problems, make sure to **flush** your output stream after printing.

- In C++, use `fflush(stdout);` or `cout << endl;`
- In Python, use `sys.stdout.flush()` or `print(flush=True)`
- For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the contest! 😃

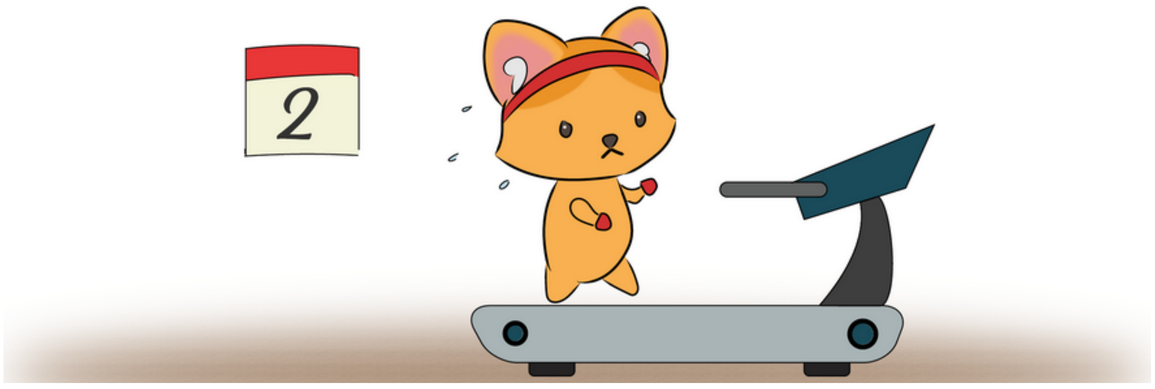
Contents

Notes

- Many problems have large input file sizes, so use fast I/O. For example:
  - In C/C++, use `scanf` and `printf`.
  - In Python, use `sys.stdin.readline()`
- On interactive problems, make sure to **flush** your output stream after printing.
  - In C++, use `fflush(stdout);` or `cout << endl;`
  - In Python, use `sys.stdout.flush()` or `print(flush=True)`
  - For more details, including for other languages, ask a question/clarification through CMS.

Good luck and enjoy the problems!

Problem B
The Astounding Race



Abby is working hard to live an Active Lifestyle, so that she might lower her Body Mass Index. Why? Well you see...

We are proud to announce our new contest, the National Olympiad in Informatics and PHysicaleducation - PHilippines (NOI.PH.PH). Rather than being confined to one seat for five hours, participants in the NOI.PH.PH must traverse perilous terrain and complete various physical challenges, in between the coding tasks!

This innovation is sure to make comp prog more exciting as a spectator sport!

NOI.PH.PH will involve n problems, which we label from 1 to n. We will also set up n *workstations*, arranged in a row and labeled 1 to n from left to right.

Obstacle courses have been set up between workstations such that traveling between adjacent workstations (that is, between workstations i and $i+1$, for $1 \leq i < n$) always takes *exactly one minute* in either direction. Contestants may *not* move between workstations by any means other than through the obstacle courses.

Contestants are *forced* to answer the problems in the order $1, 2, 3, \ldots, n$—that is, you may not initiate problem i until all of problems 1 through $i-1$ have already been answered. Also, you aren't allowed to just answer any problem anywhere. For each i from 1 to n, problem i was assigned a distinct workstation w_i, meaning you can only solve problem i when you're at workstation w_i.

Your time starts when you start the first problem, and your time ends when you finish the last problem. A contestant's *physical time penalty* is computed by totalling the amount of time they spent (in minutes) moving between workstations.

The minimum possible physical time penalty depends on how the judges decide to assign the workstations to the problems.

For example, suppose $n = 5$ and $w = [3, 1, 4, 5, 2]$. Then, the minimum physical time penalty would be 9:

- We can take our sweet time getting to workstation $w_1 = 3$.
- After that, we spend 2 minutes moving to workstation $w_2 = 1$.
- After that, we spend 3 minutes moving to workstation $w_3 = 4$.
- After that, we spend 1 minute moving to workstation $w_4 = 5$.
- Finally, we spend 3 minutes moving to workstation $w_5 = 2$.
- Once we finish the last problem, our time ends.

Since $2 + 3 + 1 + 3 = 9$, that is the minimum possible physical time penalty.

Given n and k, determine if there exist workstation assignments $w_1, w_2, \ldots, w_n$ which make it so that the minimum physical time penalty is exactly equal to k. If yes, we would *prefer* if you are also able to actually construct such an assignment. If not, but your Yes/No answers were all correct, then you can still earn partial points.

Also, there will be T test cases.

Input Format

The first line of input contains a single integer T, the number of test cases. The descriptions of the T test cases follow.

Each test case is described by a line containing the two space-separated integers n and k for that test case.

Output Format

For each test case:

- Output a line containing either `YES` or `NO`, depending on if the task is possible.
- If `YES`, also output a line containing n space-separated integers, your proposed values for $w_1, w_2, w_3, \ldots, w_n$, which should be some permutation of the integers from 1 to n.

Note that if you want the partial points, you **must** output a permutation of 1 to n after every `YES` response, even if it's incorrect.

Constraints

Let N be the sum of n across all test cases.

For all subtasks
$1 \le T \le 10^5$ $2 \le n \le 5 \times 10^5$ $0 \le k \le 10^{18}$ $N \le 10^6$

Subtask	Points	Constraints
1	**12**	$n \le 9$
2	**19**	$n \le 15$
3	**22**	$k \le 2n$
4	**35**	$N \le 5000$
5	**12**	No further constraints.

There is also partial scoring:

- You get 0 points for this subtask if there exists a test file under this subtask for which you gave an incorrect `YES` or `NO` answer in some test case.
- If not, you get 40% of the points for this subtask if there exists a test file under this subtask such that: all your `YES` and `NO` answers were valid, but there exists some `YES` test case where the provided assignments would not yield a minimum physical time penalty of k.
- Otherwise, you get 100% of the points for this subtask.

Sample I/O

Input 1	Output 1

`3`
`12 38`
`7 14`
`2 1000`
	`YES`
	`12 10 9 11 6 1 8 5 2 7 3 4`
	`YES`
	`3 1 2 7 4 6 5`
	`NO`