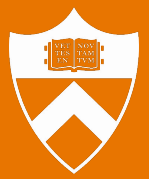
ACM Student Chapter

PEDRO PAREDES

Princeton Competitive Programming

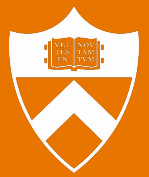
Fall 2023



Competitive Programming

Fall 2023

- Welcome to Fall 23
- Intro to Competitive Programming
- Problem Solving session



A few words about competitive programming

What is competitive programming?

- Solving complex problems with algorithms and math under tight time limits
- Participants use programming languages to create fast and correct solutions

Why should I do competitive programming?

- Learn problem-solving and coding skills, readying you for industry challenges
- Improve efficiency and creativity, tackling complex problems under a time limit
- Boost your resume, unlocks job and internship opportunities, and practice for interviews

What if I don’t want to compete?

- Our events prioritize problem-solving and learning, with optional competition
- Look out for upcoming interview prep sessions in the fall semester, designed to specifically address interview aspects (details coming soon)

Competitive programming at Princeton

We have a website: https://competitive-programming.cs.princeton.edu/

We have a discord: _____

"Division II":

- For people with little or no competitive programming experience
- Problems will often have associated readings for you to learn new concepts

"Division I":

- For people with at least some competitive programming experience
- Problems of varying difficulties to target different levels of experience.

Divisions are informal and you can switch whenever you want

Problem solving sessions are supposed to be interactive and collaborative

Last 30 minutes of each session will be used for solutions discussion

Most weeks will start with a short (~30 minute) talk on some topic (some advanced some beginner)

Logistics

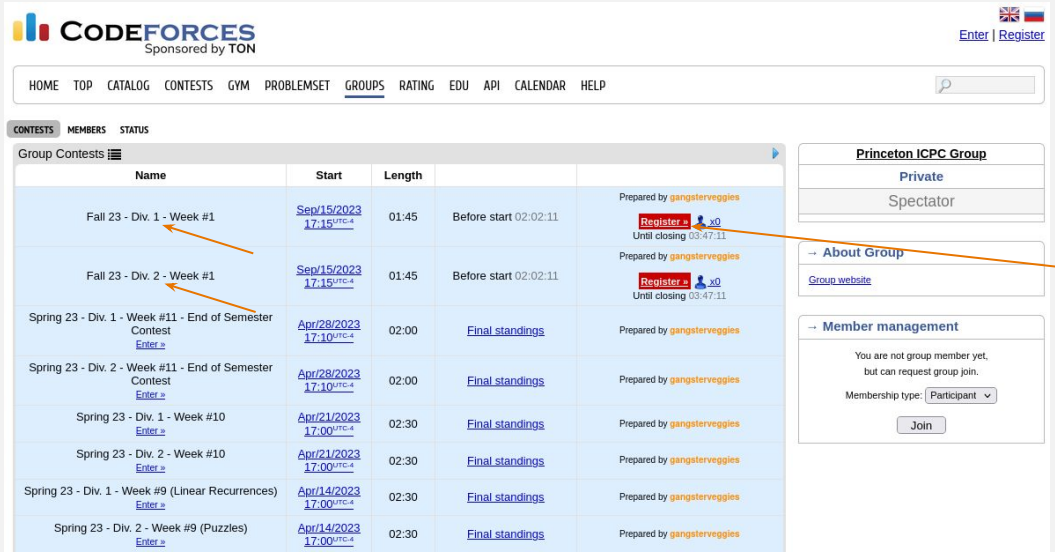
CODEFORCES Sponsored by TON

https://codeforces.com

Go to our group on codeforces and click on “Join”

This week’s problems

Past problems (you can try to solve any problem at any time)



Click here to be able to submit

ICPC and Competing

- **Local Qualifier:**
  Local selection contest at Princeton in the **first week of October**
  Participate as a team or individually in this contest
  5 teams will be selected to represent Princeton at the next stage.

- **Greater New York Regional Contest:**
  Will be held on **Sunday, October 29** at Columbia University
  Top ~5 teams qualify for the next stage, but only one per university

- **North American Championship:**
  Will happen at some point at the **end of the Spring semester** in the USA
  The first ~18 teams qualify for the next stage

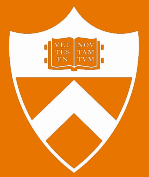
- **ICPC World Finals:**
  The last stage, date and location is yet to be announced

Note: You don't have to compete in order to benefit from these sessions!

Competitive Programming

Fall 2023

- Welcome to Fall 23
- Intro to Competitive Programming
- Problem Solving session



Intro to Competitive Programming

Problem Statement: Detailed problem statement describing the task

Input Format: Input will be provided in specific format. You need to read from standard input

Output Format: Expected format for the output also specified. Write to standard output

Testing: Output must exactly match the expected output. Even a single character difference or extra whitespace can result in a “Wrong Answer” verdict

Multiple Test Cases: Each problem has multiple test cases to evaluate correctness and efficiency

Samples: You’ll get sample input and output to help you understand the problem's requirements

Hidden Test Cases: Hidden test cases evaluate your solution. You do not have access to these test cases, so your code should handle all possible inputs

Languages and Libraries: You can use any programming language and any default library

126/226 Libraries: If you are used to the special 126/226 libraries, these are **NOT** available on codeforces

Intro to Competitive Programming

A. To My Critics

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Suneet has three digits a, b, and c.

Since math isn't his strongest point, he asks you to determine if you can choose any two digits to make a sum greater or equal to 10.

Output "YES" if there is such a pair, and "NO" otherwise.

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The only line of each test case contains three digits a, b, c ($0 \le a, b, c \le 9$).

Output

For each test case, output "YES" if such a pair exists, and "NO" otherwise.

You can output the answer in any case (for example, the strings "yEs", "yes", "Yes" and "YES" will be recognized as a positive answer).

Example

input Copy

```
5
8 1 2
4 4 5
9 9 9
0 0 0
8 5 3
```

output Copy

```
YES
NO
YES
NO
YES
```

Note

For the first test case, by choosing the digits 8 and 2 we can obtain a sum of $8 + 2 = 10$ which satisfies the condition, thus the output should be "YES".

For the second test case, any combination of chosen digits won't be at least 10, thus the output should be "NO" (note that we can not choose the digit on the same position twice).

For the third test case, any combination of chosen digits will have a sum equal to 18, thus the output should be "YES".

Intro to Competitive Programming

B. Balanced Round

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are the author of a Codeforces round and have prepared n problems you are going to set, problem i having difficulty a_i. You will do the following process:

- remove some (possibly zero) problems from the list;
- rearrange the remaining problems in any order you wish.

A round is considered *balanced* if and only if the absolute difference between the difficulty of any two consecutive problems is at most k (less or equal than k).

What is the minimum number of problems you have to remove so that an arrangement of problems is balanced?

Input

The first line contains a single integer t ($1 \le t \le 1000$) — the number of test cases.

The first line of each test case contains two positive integers n ($1 \le n \le 2 \cdot 10^5$) and k ($1 \le k \le 10^9$) — the number of problems, and the maximum allowed absolute difference between consecutive problems.

The second line of each test case contains n space-separated integers a_i ($1 \le a_i \le 10^9$) — the difficulty of each problem.

Note that the sum of n over all test cases doesn't exceed $2 \cdot 10^5$.

Output

For each test case, output a single integer — the minimum number of problems you have to remove so that an arrangement of problems is balanced.

Example

input Copy

```
7
5 1
1 2 4 5 6
1 2
10
8 3
17 3 1 20 12 5 17 12
4 2
2 4 6 8
5 3
2 3 19 10 8
3 4
1 10 5
8 1
8 3 1 4 5 10 7 3
```

output Copy

```
2
0
5
0
3
1
4
```

Note

For the first test case, we can remove the first 2 problems and construct a set using problems with the difficulties $[4, 5, 6]$, with difficulties between adjacent problems equal to $|5 - 4| = 1 \le 1$ and $|6 - 5| = 1 \le 1$.

For the second test case, we can take the single problem and compose a round using the problem with difficulty 10.

Competitive Programming

Fall 2023

- Welcome to Fall 23
- Intro to Competitive Programming
- Problem Solving session