Pupils Plan S25

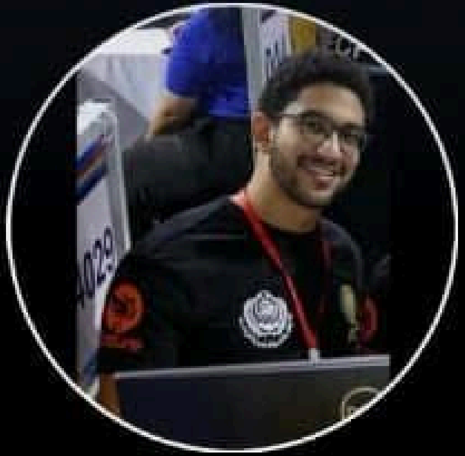
ACM GUCCPC
GUC COLLEGIATE
PROGRAMMING CONTEST

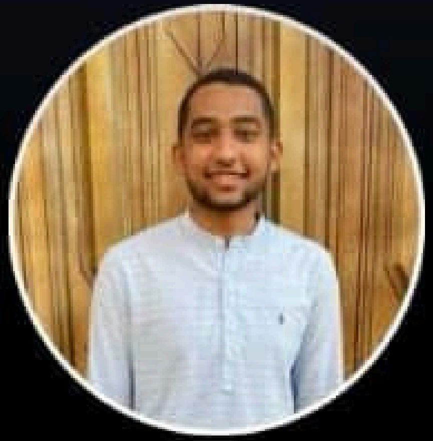
GUC ICPC COMMUNITY

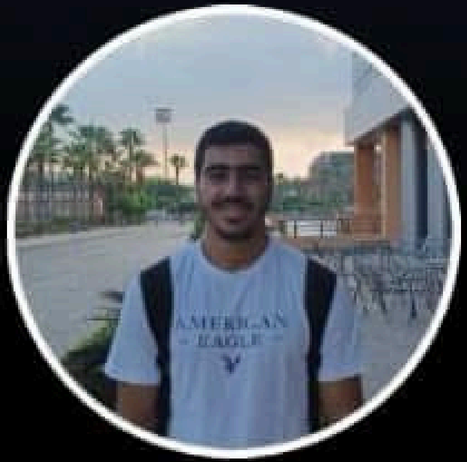
GUC Collegiate Programming Contest

MEET YOUR COACHES

PUPILS' PLAN







$$\begin{tikzcd} \text{PUPIL} \arrow[r] & \text{SPECIALIST} \arrow[r] & \text{EXPERT} \end{tikzcd}$$

📌 Why Competitive Programming ?

🚀 Helps You Get Into FAANG & Top Companies
🧠 Develops Algorithmic Thinking & Problem-Solving Skills
⚡ Makes You Code Faster & More Efficiently
🧮 Improves Mathematical & Logical Thinking
🏅 Builds Your Ranking & Recognition
👥 Builds a Strong Network & Community
🚀 CP is a Game-Changer for Jobs & Problem-Solving

📌 Why Competitive Programming ?
(With Proven Statistics & Sources)

🚀 1. Helps You Get Into FAANG & Top Tech Companies

- Google, Meta, Microsoft, Amazon, Bloomberg, and Adobe actively hire top CP competitors.
- 60%+ of ICPC World Finalists receive job offers from FAANG companies (ICPC Global Report, 2023).
- Google Code Jam winners often receive direct job offers (Google Careers Blog, 2022).
- Top Codeforces Grandmasters (2400+ rating) have been recruited by Google and Meta without a standard interview process (Ex-FAANG Recruiter, LinkedIn, 2023).
- Meta engineers report that CP-trained candidates outperform others in coding interviews by at least 30% (Meta Hiring Insights, 2021).

🧠 2. Develops Algorithmic Thinking & Problem-Solving Skills

- Google's hiring team states that CP helps in structuring logical thinking, a key skill in system design & software architecture (Google Tech Blog, 2022).
- Competitive programmers solve technical interview problems 3× faster than average candidates (HackerRank Developer Skills Report, 2023).
- 94% of developers believe CP improves their problem-solving skills (Stack Overflow Developer Survey, 2022).
- Microsoft reports that CP-trained engineers require 40% less time in debugging real-world applications (Microsoft Engineering Report, 2021).

⚡ 3. Makes You Code Faster & More Efficiently

- CP coders debug 2× faster than non-CP programmers (Google Hiring Report, 2022).
- The average time taken by a competitive programmer to write a working algorithm is 5× less than a regular coder (Codeforces Analysis, 2021).
- Top-ranked CP competitors take 30% less time to pass all test cases in online assessments (HackerRank Data Science Insights, 2023).
- Fast coding and debugging skills in CP directly translate into high productivity in real-world software development (Meta Hiring Blog, 2022).

🧮 4. Improves Mathematical & Logical Thinking

- Graph theory (used in CP) is the backbone of Google Maps, AI search engines, and networking algorithms (Google AI Research Paper, 2022).
- Modular arithmetic, a key CP topic, is widely used in cybersecurity and blockchain encryption (MIT Cryptography Report, 2023).
- Amazon’s AI team actively hires competitive programmers due to their strong mathematical foundation (Amazon AI Research Blog, 2022).
- CP participants perform 25% better in machine learning & AI courses (Harvard CS50 Report, 2021).

🏅 5. Builds Your Ranking & Recognition

- ICPC World Finalists have a 90%+ job placement rate at major tech firms (ICPC Employment Data, 2023).
- Codeforces rankings are used by recruiters to assess candidates quickly (LinkedIn Hiring Data, 2022).
- A Codeforces rating of Expert (1600+) already places you in the top 10% of programmers worldwide (Codeforces Analytics, 2022).
- Top 100 Codeforces Grandmasters have job offers from FAANG companies and hedge funds (Codeforces Employment Trends, 2023).

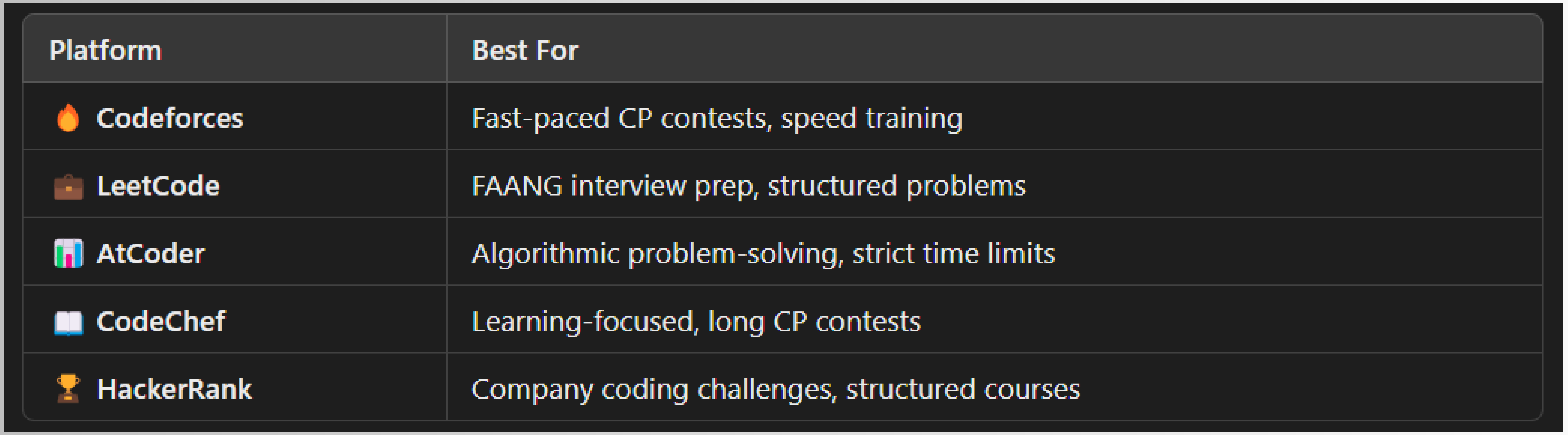
👥 6. Builds a Strong Network & Community

- ICPC winners receive sponsorships and networking opportunities at Google, Microsoft, and IBM (ICPC Industry Report, 2022).
- Competitive programmers are more likely to collaborate on tech startups due to their problem-solving background (Y Combinator Startup Analysis, 2022).
- ICPC, Google Code Jam, and Facebook Hacker Cup finalists are often invited to exclusive recruitment events (Meta Careers Blog, 2023).

🎯 7. CP is a Game-Changer for Jobs & Problem-Solving

- CP-trained candidates outperform others in job assessments & coding interviews by at least 35% (Google & Meta Hiring Reports, 2022).
- Companies like Google, Meta, Microsoft, and Amazon track Codeforces, LeetCode, and ICPC rankings when hiring engineers (LinkedIn Hiring Data, 2023).
- Hedge funds and trading firms (Jane Street, Two Sigma, Citadel) actively hire CP experts due to their algorithmic trading skills (Financial Times Report, 2022).
- CP training makes tech interviews 10× easier by improving coding speed, accuracy, and problem breakdown strategies (HackerRank Interview Statistics, 2023).

Top Coding Platforms



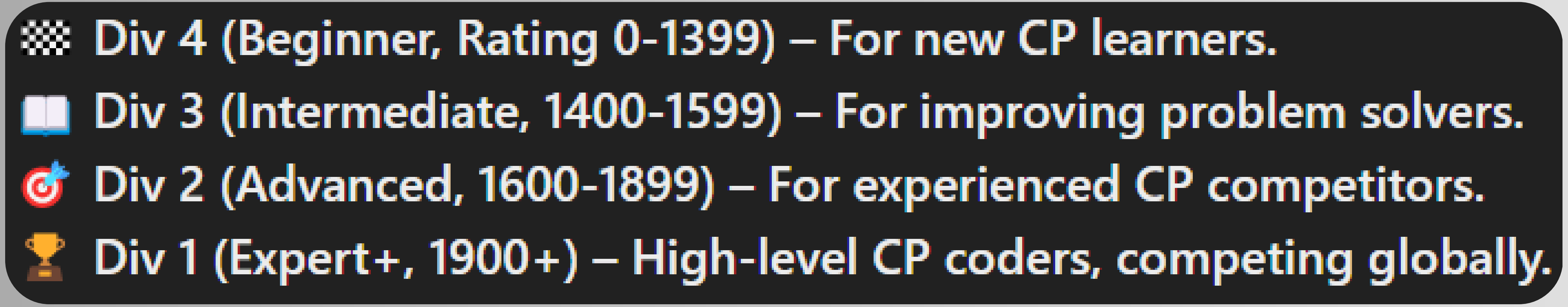
Platform	Best For
🔥 **Codeforces**	Fast-paced CP contests, speed training
💼 **LeetCode**	FAANG interview prep, structured problems
📊 **AtCoder**	Algorithmic problem-solving, strict time limits
📖 **CodeChef**	Learning-focused, long CP contests
🏆 **HackerRank**	Company coding challenges, structured courses

📌 Why Codeforces is the Best for CP?

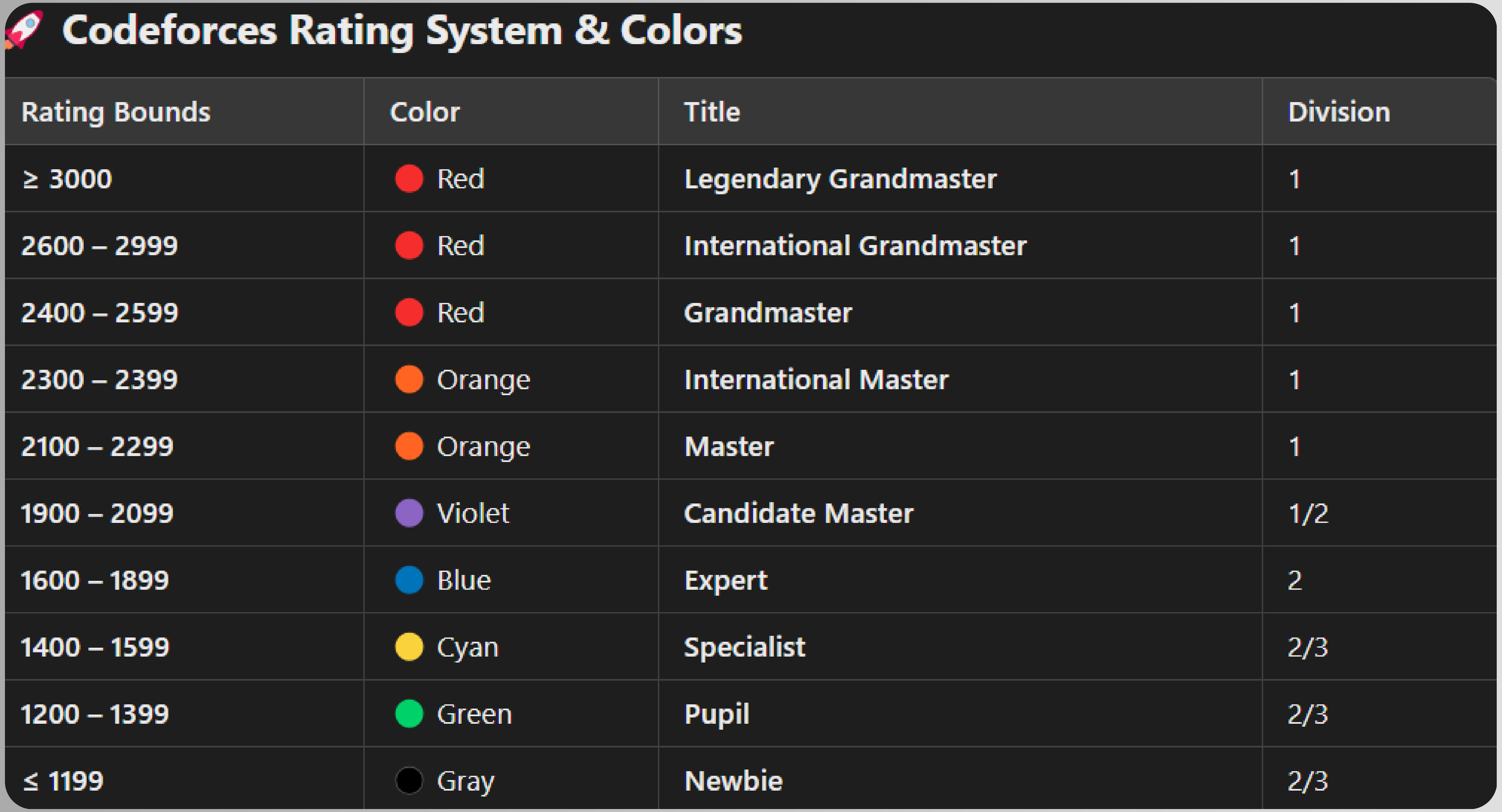
- 🔹 Fast-Paced Contests: Compete against thousands in live timed contests.
- 🔹 Elo-Based Rating System: Track your performance like a competitive game.
- 🔹 Strict Time Limits: Code must be optimized & efficient to pass test cases.
- 🔹 Large Problem Set: Over 8000+ problems sorted by difficulty.
- 🔹 Recognized by FAANG & ICPC: Many top tech companies track Codeforces ratings.

CODEFORCES DIVISIONS & RATINGS

** talk within each div



🏁 Div 4 (Beginner, Rating 0-1399) – For new CP learners.
📖 Div 3 (Intermediate, 1400-1599) – For improving problem solvers.
🎯 Div 2 (Advanced, 1600-1899) – For experienced CP competitors.
🏆 Div 1 (Expert+, 1900+) – High-level CP coders, competing globally.

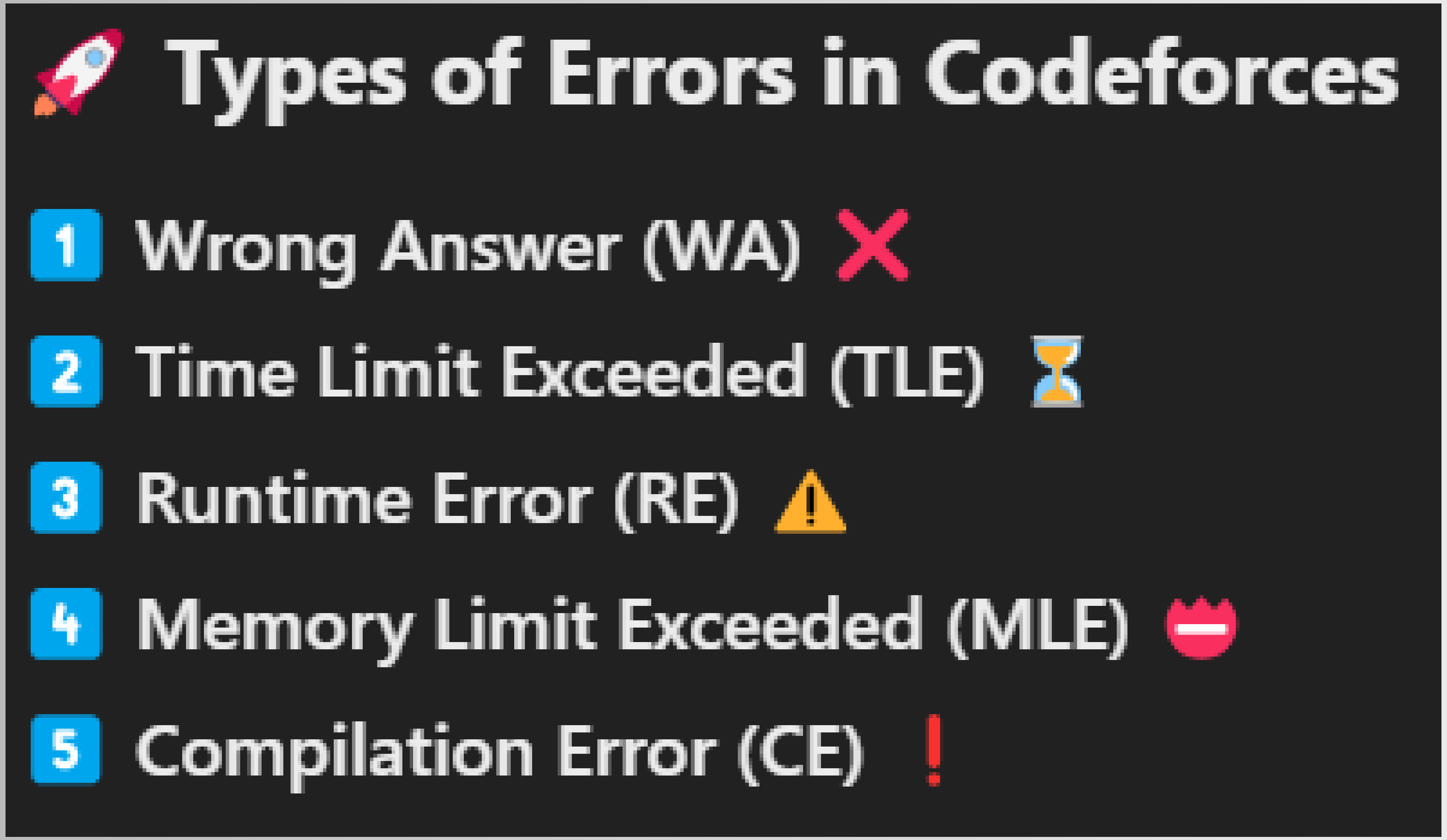


🚀 **Codeforces Rating System & Colors**

Rating Bounds	Color	Title	Division
≥ 3000	🔴 Red	Legendary Grandmaster	1
2600 – 2999	🔴 Red	International Grandmaster	1
2400 – 2599	🔴 Red	Grandmaster	1
2300 – 2399	🟠 Orange	International Master	1
2100 – 2299	🟠 Orange	Master	1
1900 – 2099	🟣 Violet	Candidate Master	1/2
1600 – 1899	🔵 Blue	Expert	2
1400 – 1599	🟡 Cyan	Specialist	2/3
1200 – 1399	🟢 Green	Pupil	2/3
≤ 1199	⚫ Gray	Newbie	2/3

Top Individuals Egypt

Top GUCians



🚀 Types of Errors in Codeforces

1️⃣ Wrong Answer (WA) ❌

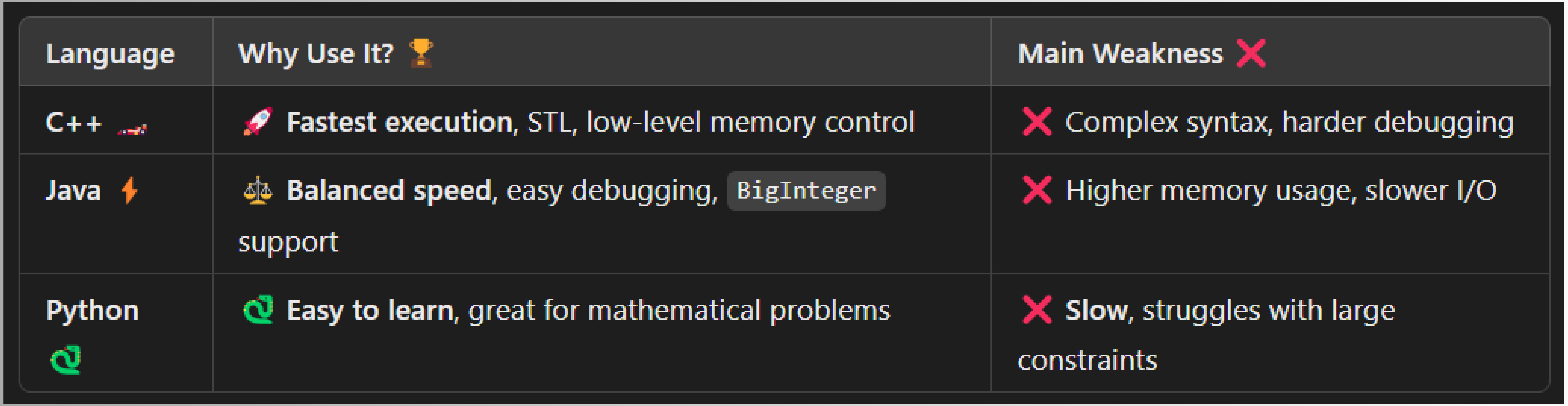
2️⃣ Time Limit Exceeded (TLE) ⏳

3️⃣ Runtime Error (RE) ⚠️

4️⃣ Memory Limit Exceeded (MLE) ⛔

5️⃣ Compilation Error (CE) ❗

Java vs. C++ vs. Python



Language	Why Use It? 🏆	Main Weakness ❌
C++ 🏎️	🚀 **Fastest execution**, STL, low-level memory control	❌ Complex syntax, harder debugging
Java ⚡	⚖️ **Balanced speed**, easy debugging, `BigInteger` support	❌ Higher memory usage, slower I/O
Python 🐍	🐍 **Easy to learn**, great for mathematical problems	❌ **Slow**, struggles with large constraints

✅ **C++ is ideal for large constraints ($N \le 10^8$).**
✅ **Java handles moderate constraints ($N \le 2 \times 10^7$) before hitting TLE.**
✅ **Python struggles with $N > 10^6$ unless optimized (e.g., PyPy).**

Fast Input Templates for Java, C++, and Python 🚀

```java
import java.io.*;
import java.util.*;

/*
* some notes to mention:
*
* 1- the Scanner that we're using is faster the Scanner class of java
* 2- we use PrintWriter instead of System.out because it is faster as well
* 3- don't let the syntax of java be a problem for you. google things, ask chatGPT, or even ask us
*/

public class Main {
final static PrintWriter pw = new PrintWriter(System.out);
final static Scanner sc = new Scanner(System.in);

public static void main(String[] args) throws IOException {
// start your code here
pw.close();
}

static class Scanner {

StringTokenizer st;
BufferedReader br;
public Scanner(InputStream s) {
br = new BufferedReader(new InputStreamReader(s));
}
public Scanner(String file) throws IOException {
br = new BufferedReader(new FileReader(file));
}
public Scanner(FileReader r) {
br = new BufferedReader(r);
}
public String next() throws IOException {
while (st == null || !st.hasMoreTokens())
st = new StringTokenizer(br.readLine());
return st.nextToken();
}
public String readAllLines(BufferedReader reader) throws IOException {
StringBuilder content = new StringBuilder();
String line;
while ((line = reader.readLine()) != null) {
content.append(line);
content.append(System.lineSeparator());
}
return content.toString();
}
public int nextInt() throws IOException {
return Integer.parseInt(next());
}
public long nextLong() throws IOException {
return Long.parseLong(next());
}
public String nextLine() throws IOException {
return br.readLine();
}
public double nextDouble() throws IOException {
String x = next();
StringBuilder sb = new StringBuilder("0");
double res = 0, f = 1;
boolean dec = false, neg = false;
int start = 0;
if (x.charAt(0) == '-') {
neg = true;
start++;
}
for (int i = start; i < x.length(); i++)
if (x.charAt(i) == '.') {
res = Long.parseLong(sb.toString());
sb = new StringBuilder("0");
dec = true;
} else {
sb.append(x.charAt(i));
if (dec)
f *= 10;
}
res += Long.parseLong(sb.toString()) / f;
return res * (neg ? -1 : 1);
}
public long[] nextlongArray(int n) throws IOException {
long[] a = new long[n];
for (int i = 0; i < n; i++)
a[i] = nextLong();
return a;
}
public Long[] nextLongArray(int n) throws IOException {
Long[] a = new Long[n];
for (int i = 0; i < n; i++)
a[i] = nextLong();
return a;
}
public int[] nextIntArray(int n) throws IOException {
int[] a = new int[n];
for (int i = 0; i < n; i++)
a[i] = nextInt();
return a;
}
public Integer[] nextIntegerArray(int n) throws IOException {
Integer[] a = new Integer[n];
for (int i = 0; i < n; i++)
a[i] = nextInt();
return a;
}
public boolean ready() throws IOException {
return br.ready();
}

}
}
```

Java

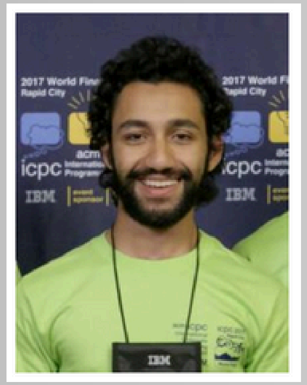
```cpp
#include <bits/stdc++.h>
using namespace std;

#define fast_io
ios::sync_with_stdio(false);
cin.tie(NULL);

int main() {
fast_io;
// Start coding here
return 0;
}
```

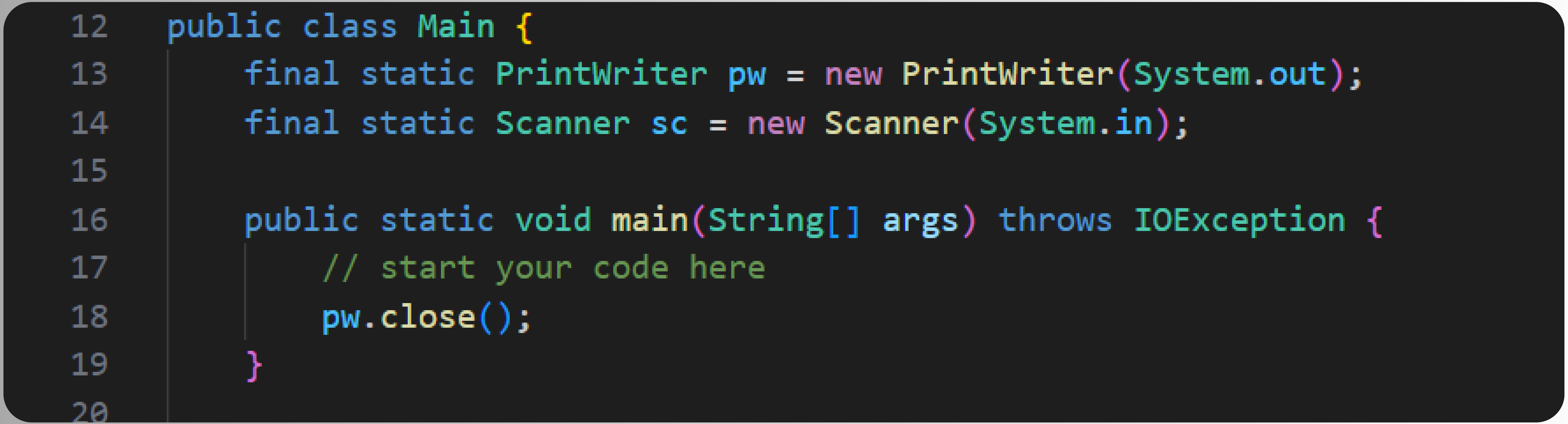
C++

```python
import sys

def fast_input():
return sys.stdin.read().split()

data = fast_input()
sys.stdout.write("\n".join(data) + "\n")
```

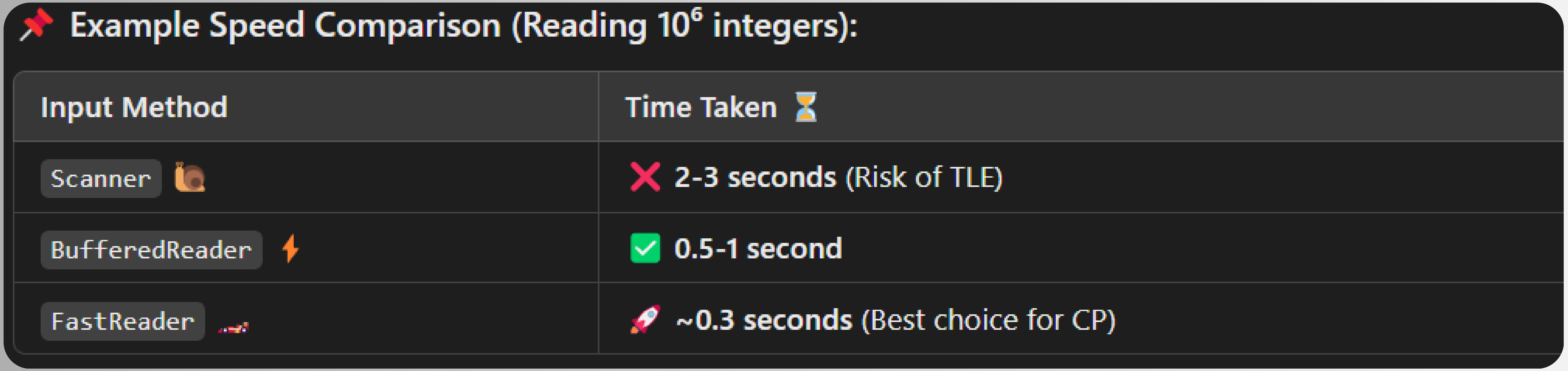
Python

By : Ahmed ElSagheer





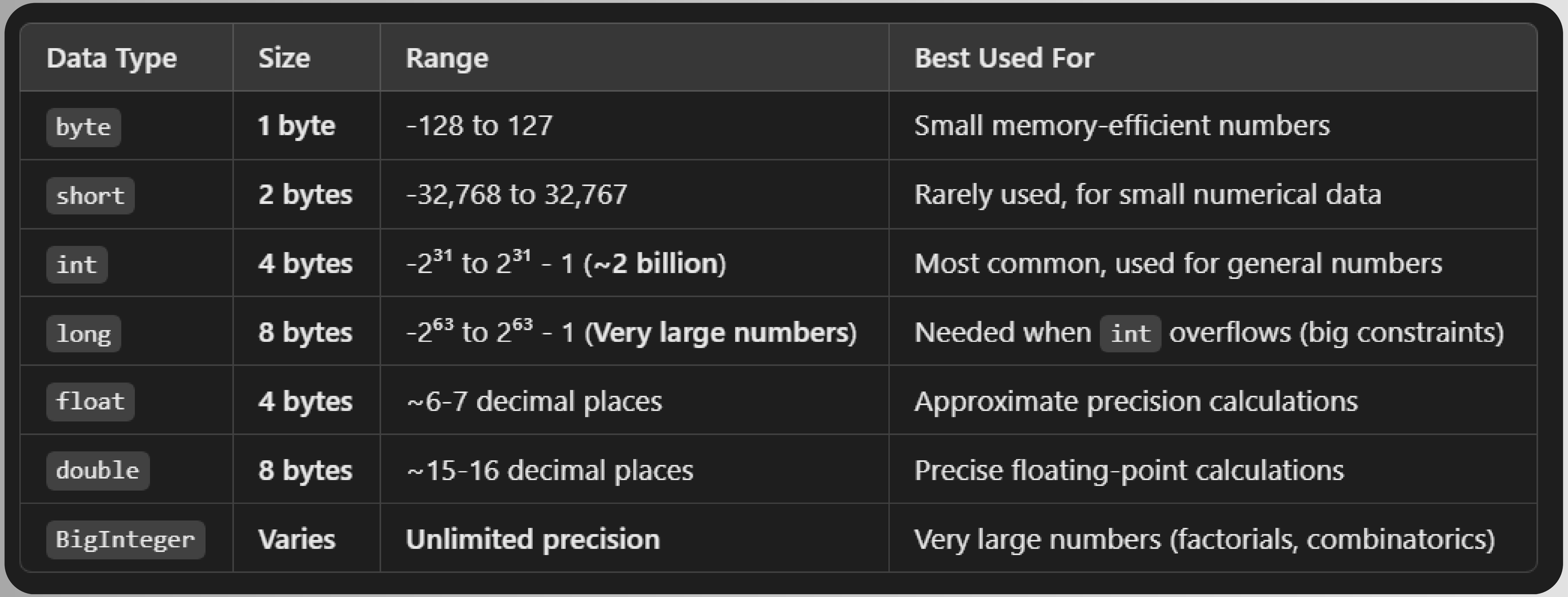
```java
public class Main {
    final static PrintWriter pw = new PrintWriter(System.out);
    final static Scanner sc = new Scanner(System.in);

    public static void main(String[] args) throws IOException {
        // start your code here
        pw.close();
    }
```



📌 Example Speed Comparison (Reading 10^6 integers):

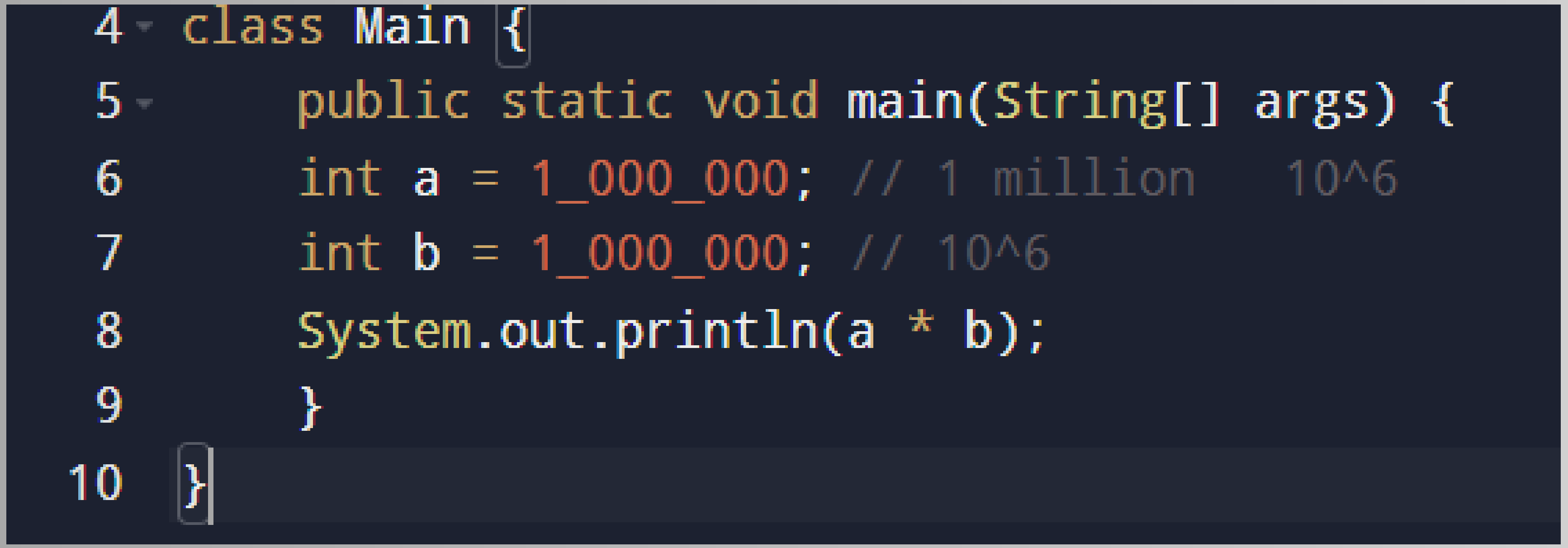
Input Method	Time Taken ⏳
`Scanner` 🐌	❌ **2-3 seconds** (Risk of TLE)
`BufferedReader` ⚡	✅ **0.5-1 second**
`FastReader` 🏎️	🚀 **~0.3 seconds** (Best choice for CP)

📌 Understanding Numbers in Java – Data Types & Differences



Data Type	Size	Range	Best Used For
`byte`	**1 byte**	-128 to 127	Small memory-efficient numbers
`short`	**2 bytes**	-32,768 to 32,767	Rarely used, for small numerical data
`int`	**4 bytes**	-2^{31} to 2^{31} - 1 (**~2 billion**)	Most common, used for general numbers
`long`	**8 bytes**	-2^{63} to 2^{63} - 1 (**Very large numbers**)	Needed when `int` overflows (big constraints)
`float`	**4 bytes**	~6-7 decimal places	Approximate precision calculations
`double`	**8 bytes**	~15-16 decimal places	Precise floating-point calculations
`BigInteger`	**Varies**	**Unlimited precision**	Very large numbers (factorials, combinatorics)

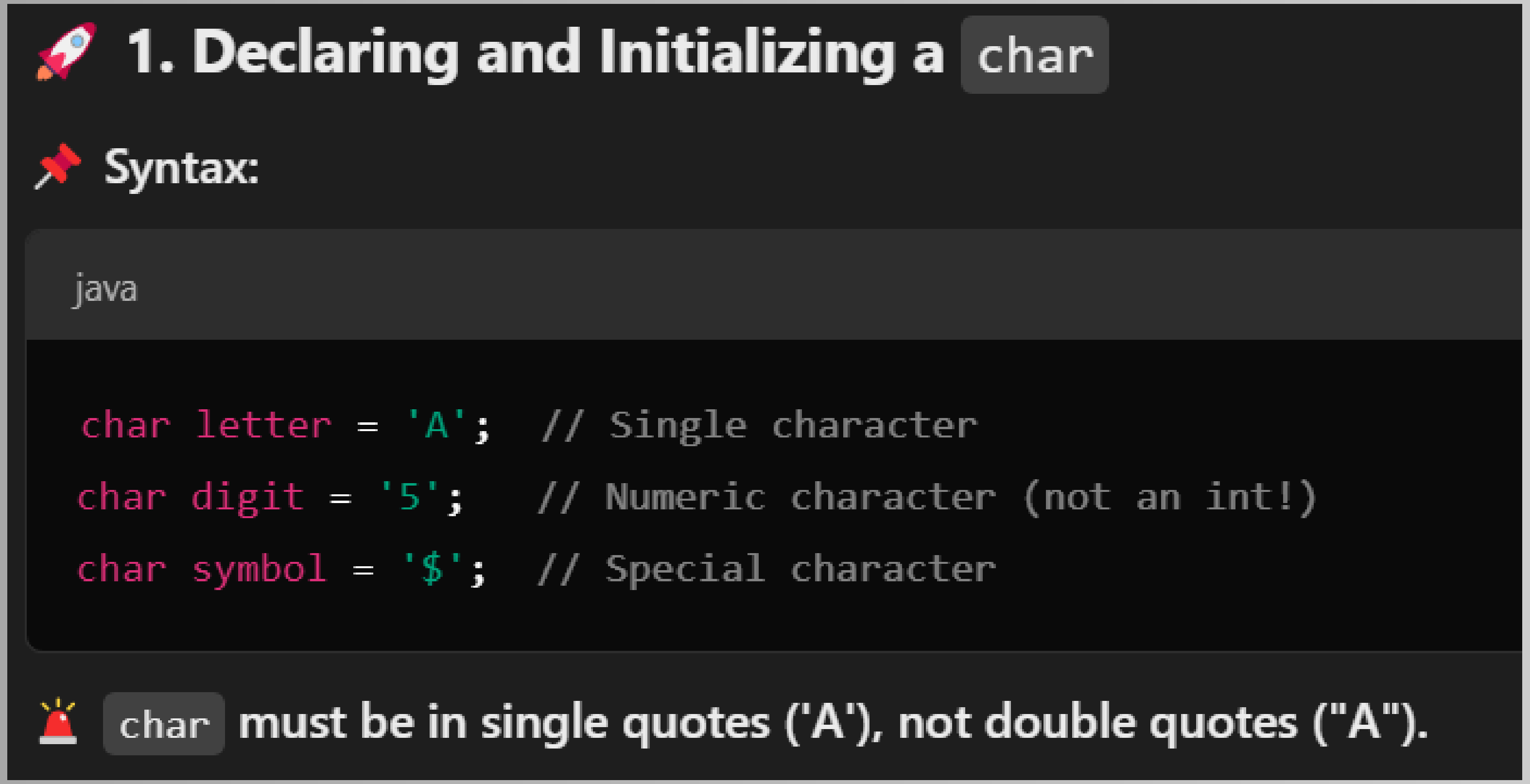
Quick Quiz !



```java
class Main {
    public static void main(String[] args) {
    int a = 1_000_000; // 1 million   10^6
    int b = 1_000_000; // 10^6
    System.out.println(a * b);
    }
}
```

Reference to read :
https://www.cs.cornell.edu/~tomf/notes/cps104/twoscomp.html

📌 Understanding Chars in Java

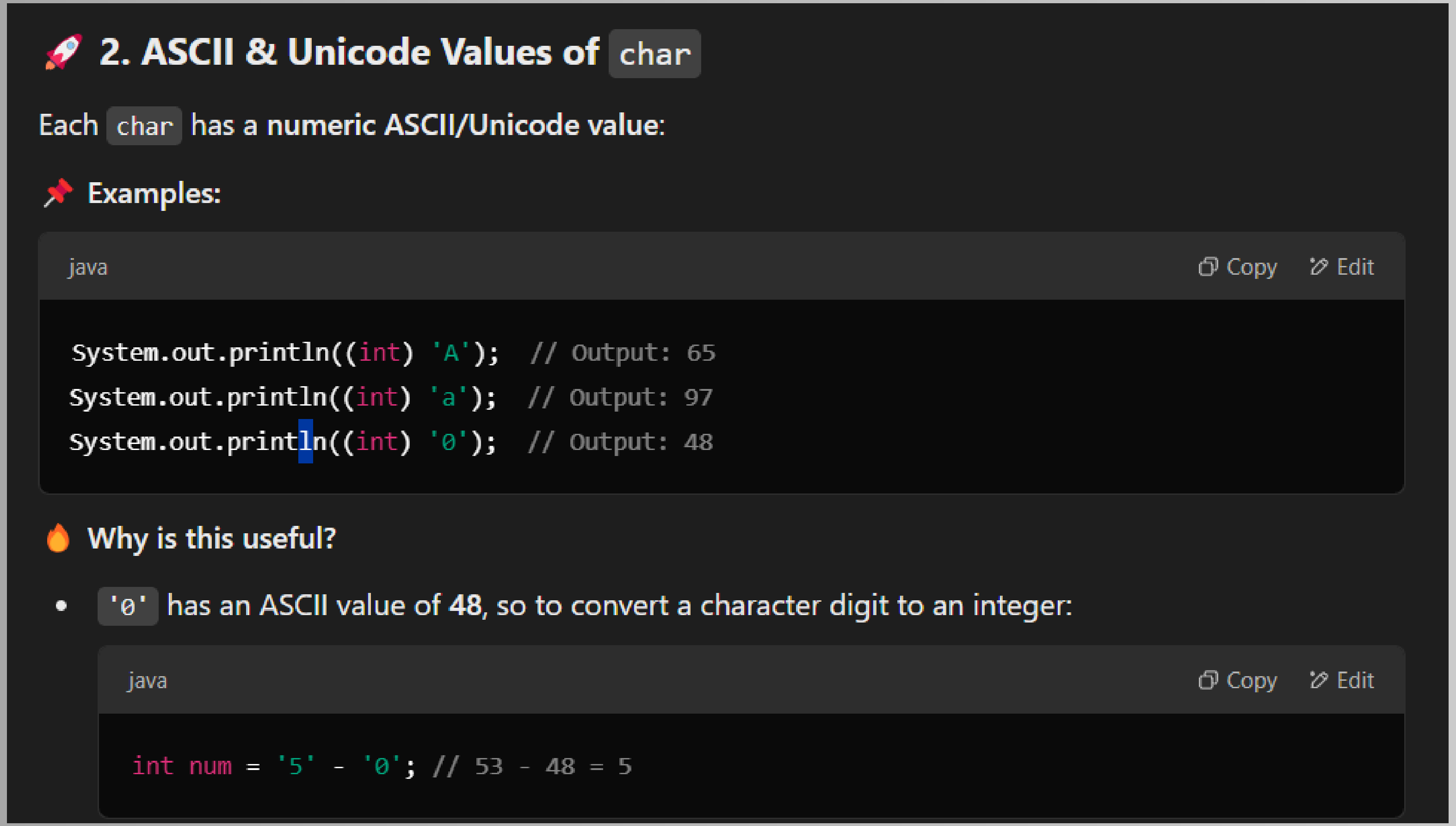


🚀 1. Declaring and Initializing a `char`

📌 **Syntax:**

```java
char letter = 'A';  // Single character
char digit = '5';   // Numeric character (not an int!)
char symbol = '$';  // Special character
```

🚨 `char` **must be in single quotes ('A'), not double quotes ("A").**



🚀 2. ASCII & Unicode Values of `char`

Each `char` has a numeric ASCII/Unicode value:

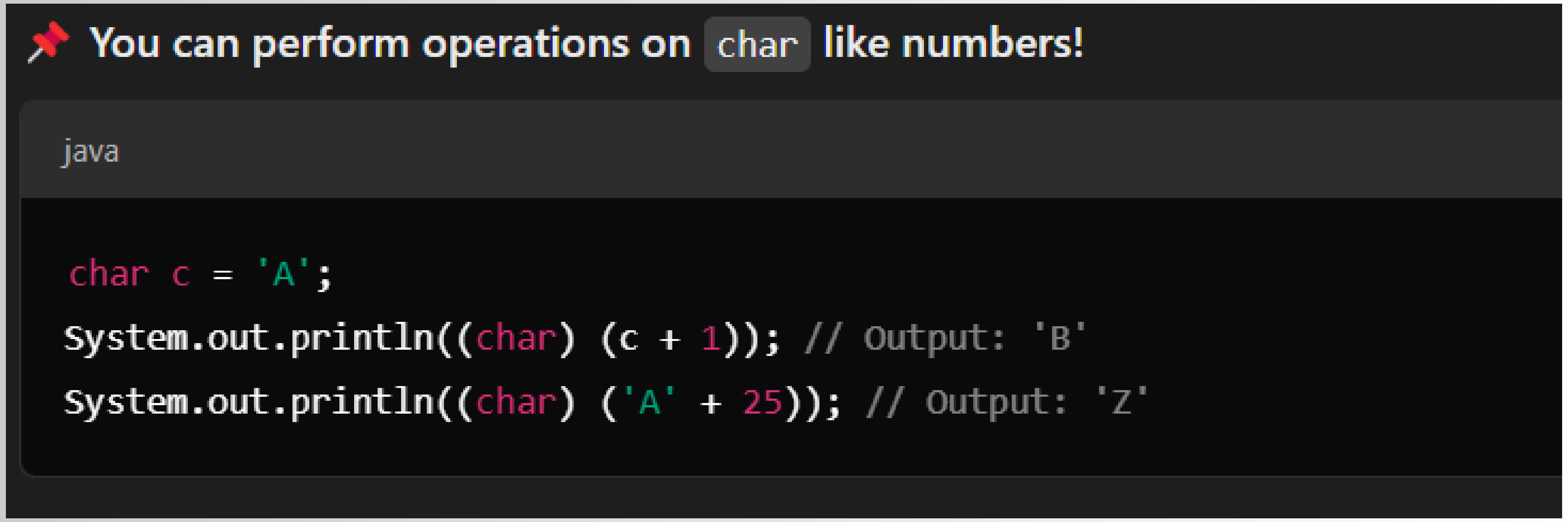
📌 **Examples:**

```java
System.out.println((int) 'A');  // Output: 65
System.out.println((int) 'a');  // Output: 97
System.out.println((int) '0');  // Output: 48
```

🔥 **Why is this useful?**

- `'0'` has an ASCII value of **48**, so to convert a character digit to an integer:

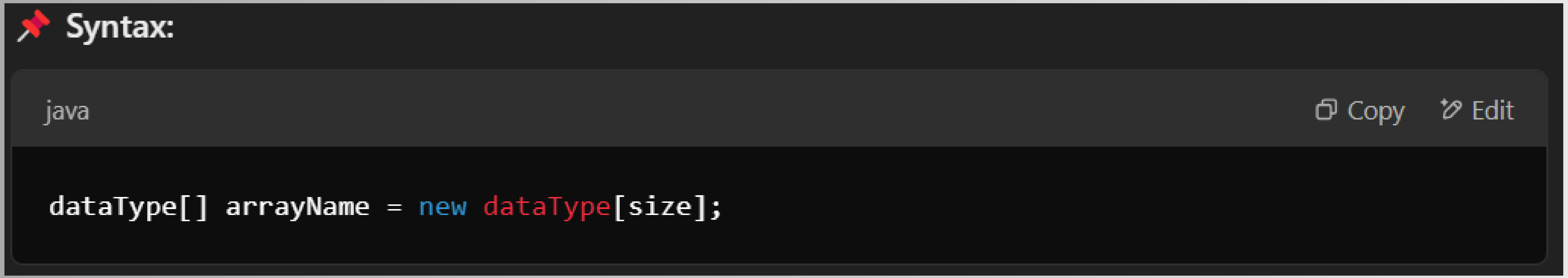
```java
int num = '5' - '0'; // 53 - 48 = 5
```



📌 **You can perform operations on `char` like numbers!**

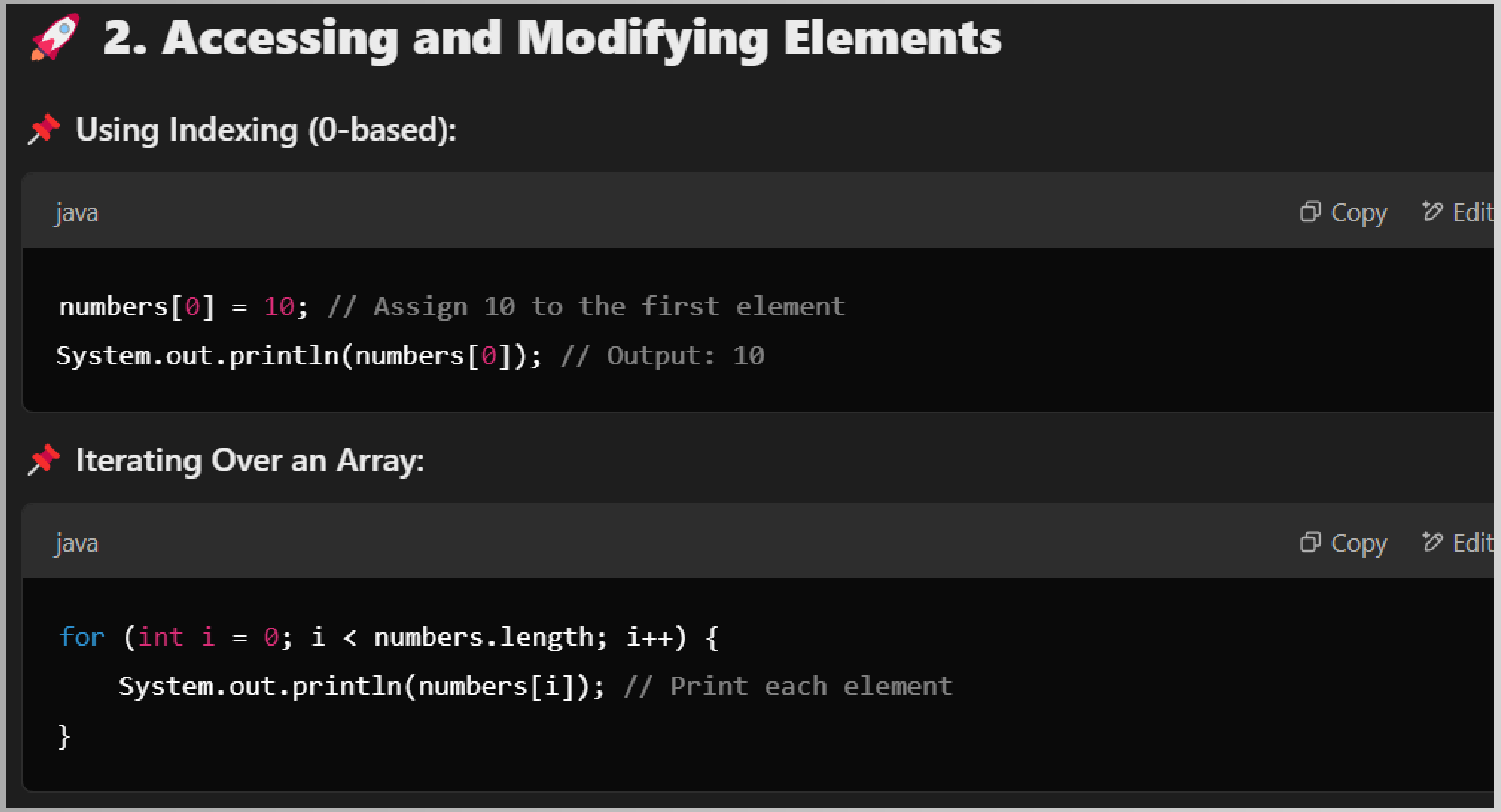
```java
char c = 'A';
System.out.println((char) (c + 1)); // Output: 'B'
System.out.println((char) ('A' + 25)); // Output: 'Z'
```

Array(static)



📌 Syntax:

```java
dataType[] arrayName = new dataType[size];
```



🚀 2. Accessing and Modifying Elements

📌 Using Indexing (0-based):

```java
numbers[0] = 10; // Assign 10 to the first element
System.out.println(numbers[0]); // Output: 10
```

📌 Iterating Over an Array:

```java
for (int i = 0; i < numbers.length; i++) {
    System.out.println(numbers[i]); // Print each element
}
```

Highly Recommend If no Java experience :

https://www.geeksforgeeks.org/arrays-in-java/

https://codeforces.com/group/GFvHTagAfJ/contest/502939/problem/C

time limit per test: 1 second

memory limit per test: 256 megabytes

The story began when $Nour$ was playing a game. On one of its levels, there was a huge door with a huge machine beside it. The door was locked and it seemed that $Nour$ had to use the machine to open the door. When $Nour$ approached the machine, it gave him a number and asked him to count the number of even, odd and prime digits in that number. $Nour$ is a 5 years old young boy who loved the game but he couldn't unlock this door and that's why he is about to hate it. Can you help him and let him continue play the game peacefully?

Input

The first and only line contains an integer x $(1 \leq x \leq 10^{18})$ — The number that the machine game $Nour$.

Output

Print 3 integers. The first integer is the number of even digits in x, the second integer is the number of odd digits in x and the third integer is the number of prime digits in x.

Examples

input Copy

```
12345
```

output Copy

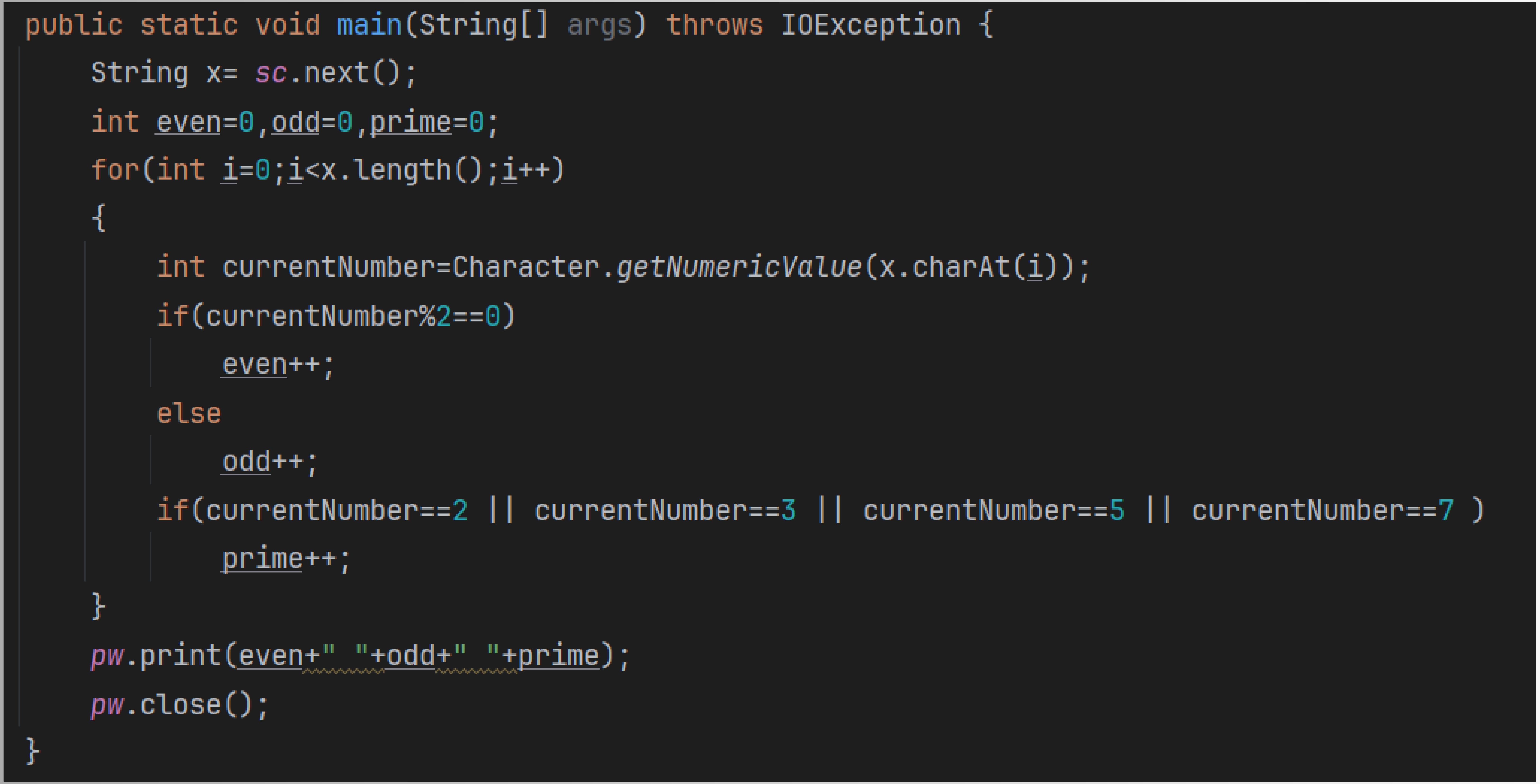
```
2 3 3
```

input

```
913
```

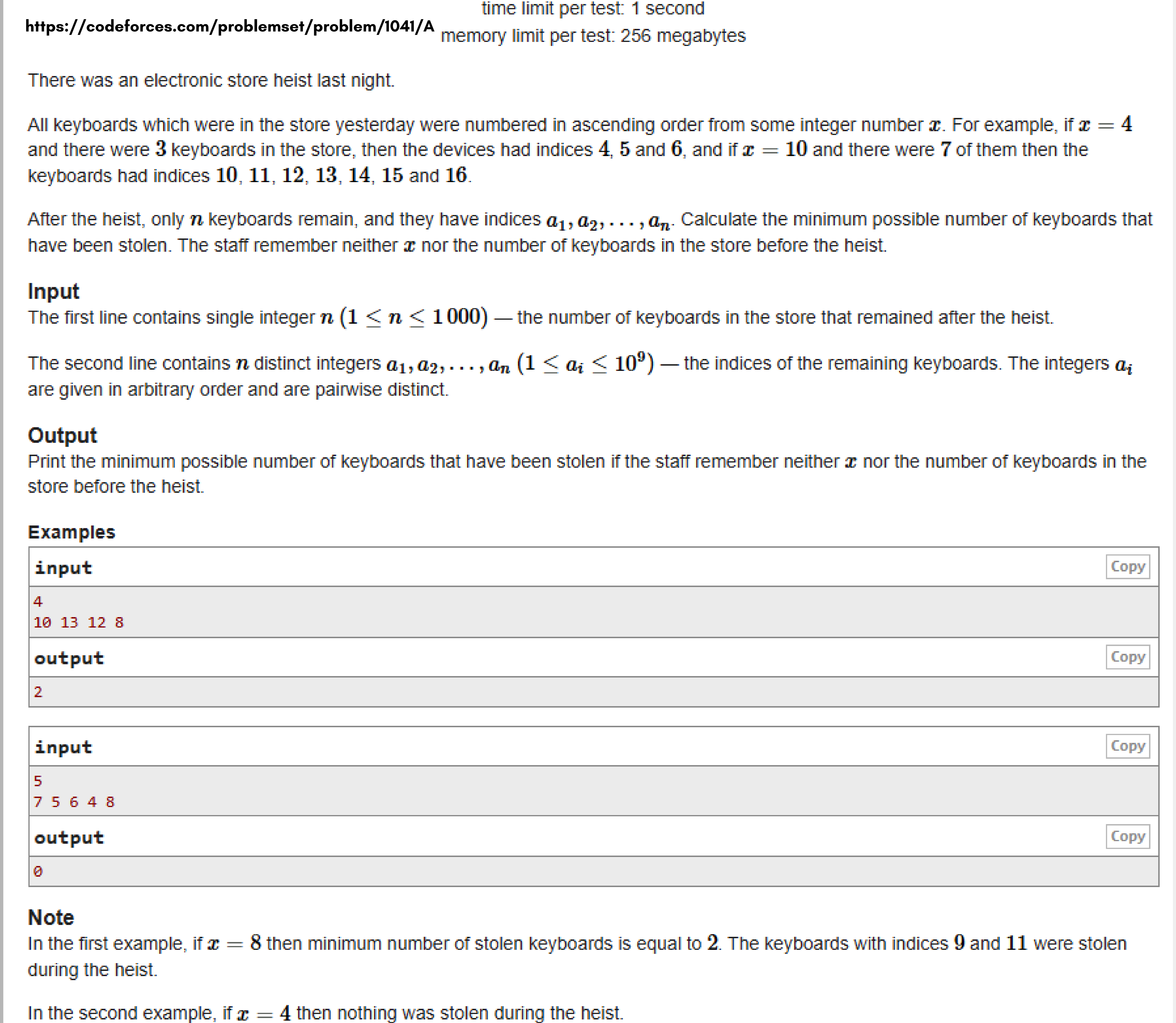
output Copy

```
0 3 1
```



```java
public static void main(String[] args) throws IOException {
    String x= sc.next();
    int even=0,odd=0,prime=0;
    for(int i=0;i<x.length();i++)
    {
        int currentNumber=Character.getNumericValue(x.charAt(i));
        if(currentNumber%2==0)
            even++;
        else
            odd++;
        if(currentNumber==2 || currentNumber==3 || currentNumber==5 || currentNumber==7 )
            prime++;
    }
    pw.print(even+" "+odd+" "+prime);
    pw.close();
}
```

https://codeforces.com/problemset/problem/1041/A



time limit per test: 1 second
memory limit per test: 256 megabytes

There was an electronic store heist last night.

All keyboards which were in the store yesterday were numbered in ascending order from some integer number x. For example, if $x = 4$ and there were 3 keyboards in the store, then the devices had indices 4, 5 and 6, and if $x = 10$ and there were 7 of them then the keyboards had indices 10, 11, 12, 13, 14, 15 and 16.

After the heist, only n keyboards remain, and they have indices $a_1, a_2, \dots, a_n$. Calculate the minimum possible number of keyboards that have been stolen. The staff remember neither x nor the number of keyboards in the store before the heist.

Input

The first line contains single integer n $(1 \le n \le 1\,000)$ — the number of keyboards in the store that remained after the heist.

The second line contains n distinct integers $a_1, a_2, \dots, a_n$ $(1 \le a_i \le 10^9)$ — the indices of the remaining keyboards. The integers a_i are given in arbitrary order and are pairwise distinct.

Output

Print the minimum possible number of keyboards that have been stolen if the staff remember neither x nor the number of keyboards in the store before the heist.

Examples

input Copy

```
4
10 13 12 8
```

output Copy

```
2
```

input Copy

```
5
7 5 6 4 8
```

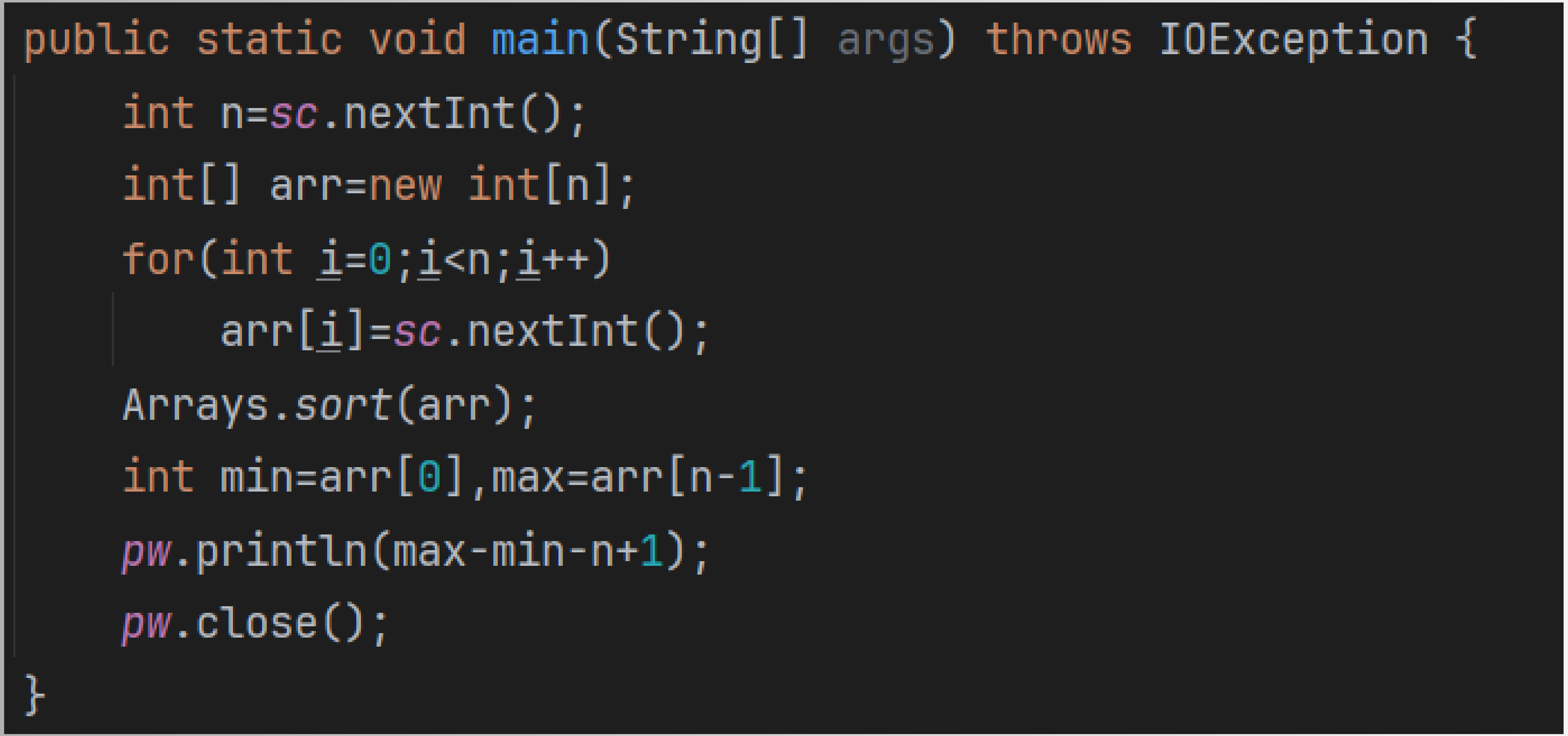
output Copy

```
0
```

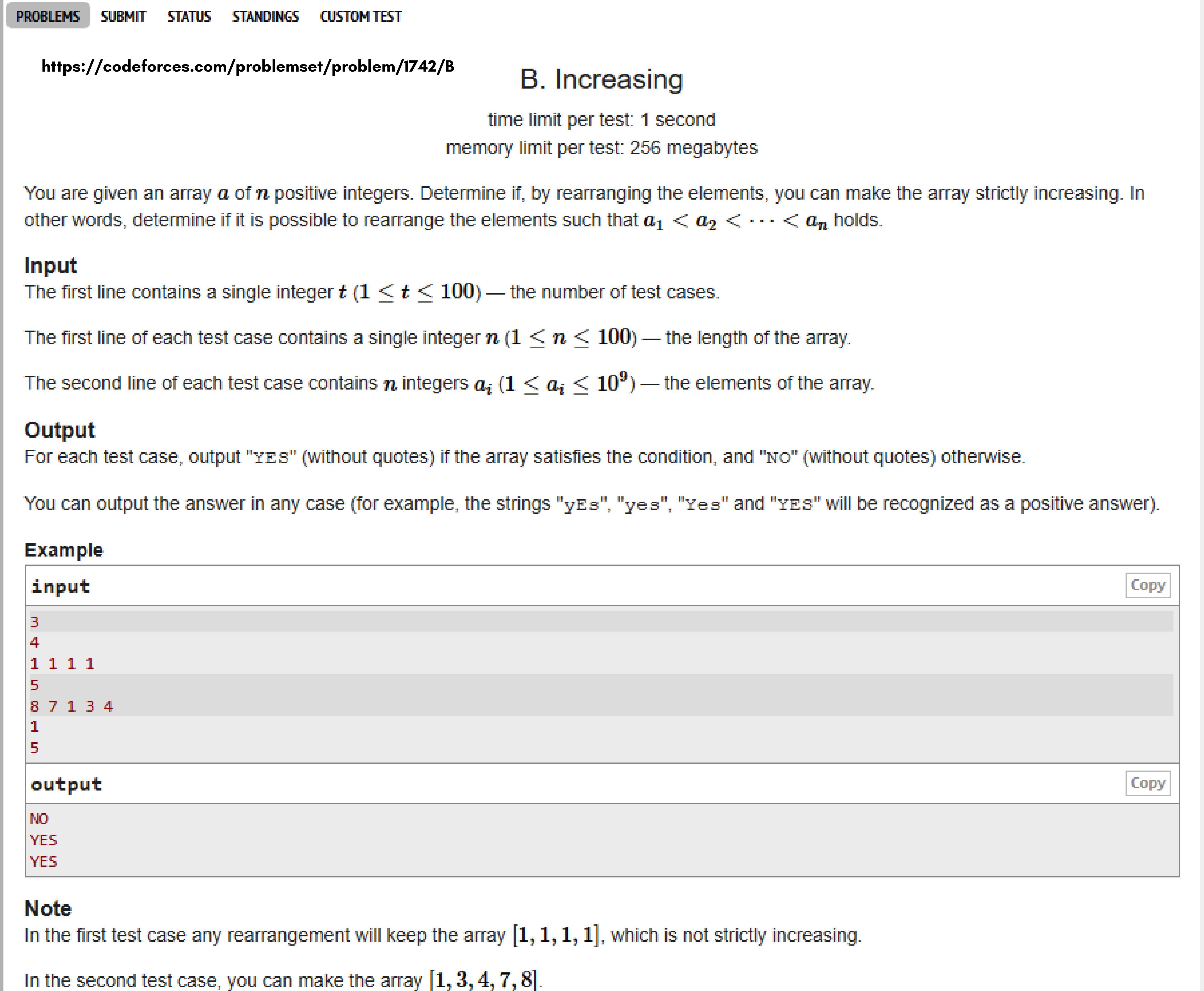
Note

In the first example, if $x = 8$ then minimum number of stolen keyboards is equal to 2. The keyboards with indices 9 and 11 were stolen during the heist.

In the second example, if $x = 4$ then nothing was stolen during the heist.



```java
public static void main(String[] args) throws IOException {
    int n=sc.nextInt();
    int[] arr=new int[n];
    for(int i=0;i<n;i++)
        arr[i]=sc.nextInt();
    Arrays.sort(arr);
    int min=arr[0],max=arr[n-1];
    pw.println(max-min-n+1);
    pw.close();
}
```



https://codeforces.com/problemset/problem/1742/B

B. Increasing

time limit per test: 1 second

memory limit per test: 256 megabytes

You are given an array a of n positive integers. Determine if, by rearranging the elements, you can make the array strictly increasing. In other words, determine if it is possible to rearrange the elements such that $a_1 < a_2 < \cdots < a_n$ holds.

Input

The first line contains a single integer t ($1 \leq t \leq 100$) — the number of test cases.

The first line of each test case contains a single integer n ($1 \leq n \leq 100$) — the length of the array.

The second line of each test case contains n integers a_i ($1 \leq a_i \leq 10^9$) — the elements of the array.

Output

For each test case, output "`YES`" (without quotes) if the array satisfies the condition, and "`NO`" (without quotes) 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

```
3
4
1 1 1 1
5
8 7 1 3 4
1
5
```

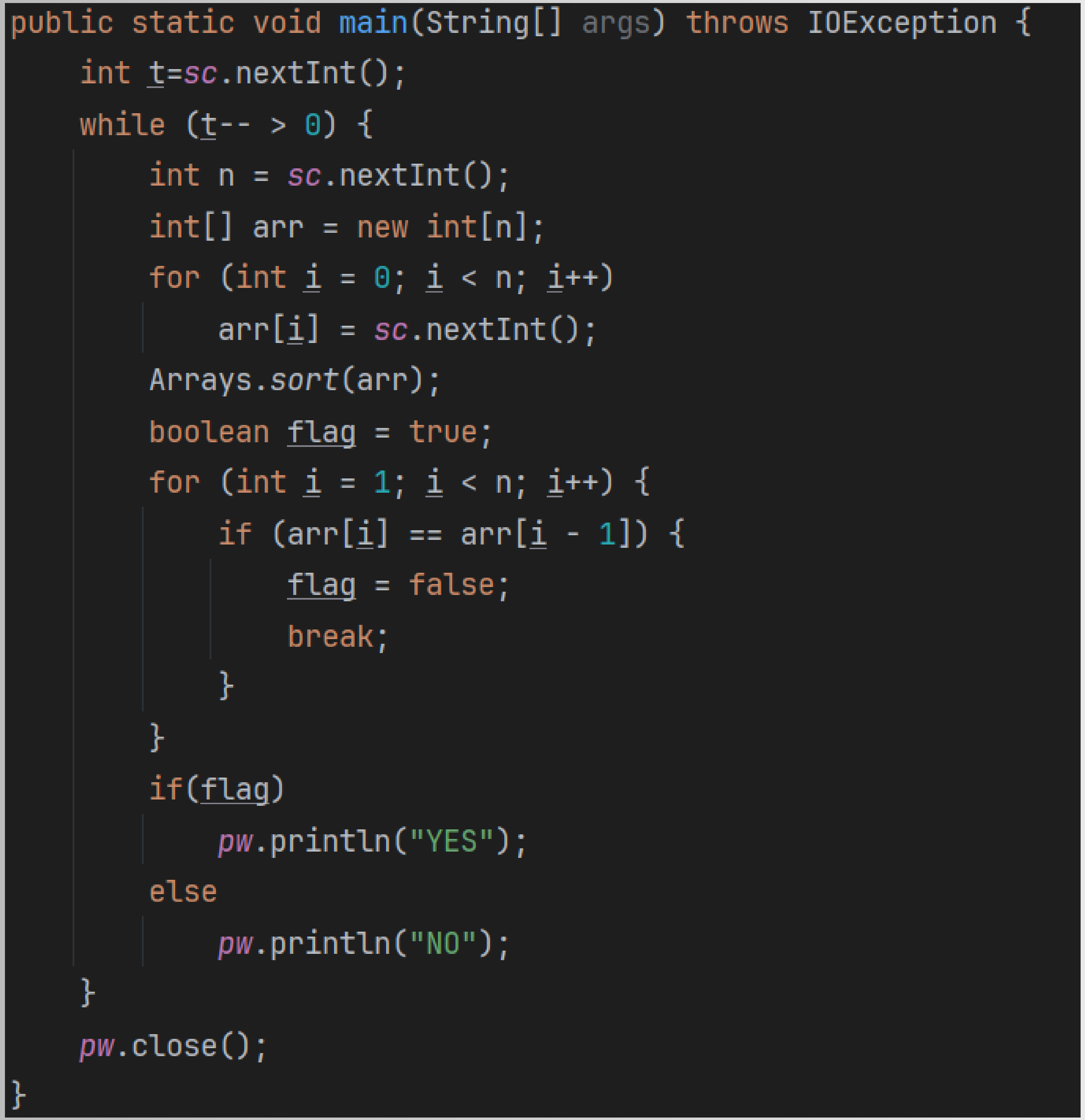
output

```
NO
YES
YES
```

Note

In the first test case any rearrangement will keep the array $[1, 1, 1, 1]$, which is not strictly increasing.

In the second test case, you can make the array $[1, 3, 4, 7, 8]$.


```java
public static void main(String[] args) throws IOException {
    int t=sc.nextInt();
    while (t-- > 0) {
        int n = sc.nextInt();
        int[] arr = new int[n];
        for (int i = 0; i < n; i++)
            arr[i] = sc.nextInt();
        Arrays.sort(arr);
        boolean flag = true;
        for (int i = 1; i < n; i++) {
            if (arr[i] == arr[i - 1]) {
                flag = false;
                break;
            }
        }
        if(flag)
            pw.println("YES");
        else
            pw.println("NO");
    }
    pw.close();
}
```

References :

Sessions - Winter 2024

Sheet 1 (recommneded : do not spoil)

Dr. Mostafa Saad - ACM/ICPC - Newcomers