

FINAL REVIEW PROJECT EXHIBITION

GROUP NO. -203

Guided by- Dr Aanjan Kumar

Team presentation



TEAM MEMBERS

ANANT KRISHNA-21BCE10083

TANYA BARSAINYA-21BCE10116

THAKOR SANDIPSINH-21BCE11437

PULKIT KUMAR MATHUR-21BCE11602



ONLINE JUDGE FOR COMPETITIVE PROGRAMMING



TABLE OF CONTENTS



- INTRODUCTION
- EXISTING WORK WITH LIMITATIONS
- PROPOSED WORK AND METHODOLOGY
- NOVELTY OF THE PROJECT
- REAL TIME USAGE
- HARDWARE & SOFTWARE REQUIREMENTS
- OVERALL SYSTEM ARCHITECTURE DIAGRAM.
- LITERATURE REVIEW
- MODULE DESCRIPTION
- MODULE WORK FLOW EXPLANATION.
- IMPLEMENTATION AND CODING.
- DEMO VIDEO.
- SNAP SHOT OF YOUR PROJECT.
- TESTING.
- RESULT AND DISCUSSION. (INPUT AND OUTPUT)
- CONCLUSION.

INTRODUCTION

Competitive programming has become increasingly popular over the years, with many programmers participating in various contests to test their skills against other coders. However, testing and evaluating code can be time-consuming and challenging, especially when dealing with complex algorithms and data structures. This is where our project comes in – an online judge system for competitive programming that provides a fast and efficient platform for testing and evaluating code.

Our project aims to help competitive programmers to test their code against a series of challenging test cases, and to provide accurate verdicts as to whether the code submitted is correct or not.



- The online judge system shows the results in terms of time (sec) and memory (MB) used, which allows coders to optimize their code for better performance.
- Our project provides a user-friendly platform for competitive programmers to evaluate their code quickly and accurately, allowing them to improve their coding skills and performance in competitive programming.
- Overall, our project provides an essential tool for competitive programmers to test and evaluate their code, as well as improve their coding skills and performance.



EXISTING WORK WITH LIMITATIONS

1. Lack of transparency: Online judges don't provide enough information about how the scoring and evaluation process works, which can lead to confusion and frustration for participants.
2. Limited programming languages: Online judges support only a limited number of programming languages, which can be a problem for participants who prefer to use a language that is not supported.
3. Inconsistent performance: The performance of online judges can vary based on the complexity of the problem being evaluated and the number of submissions being processed, which can lead to delays and inconsistencies in results.
4. Cheating: Online judges are vulnerable to cheating, including plagiarism, collaboration, and the use of unauthorized resources.
5. Lack of creativity: Online judges focus on standard algorithms and problems, which can limit the creative potential of competitive programming and discourage experimentation and innovation.





PROPOSED WORK AND METHODOLOGY

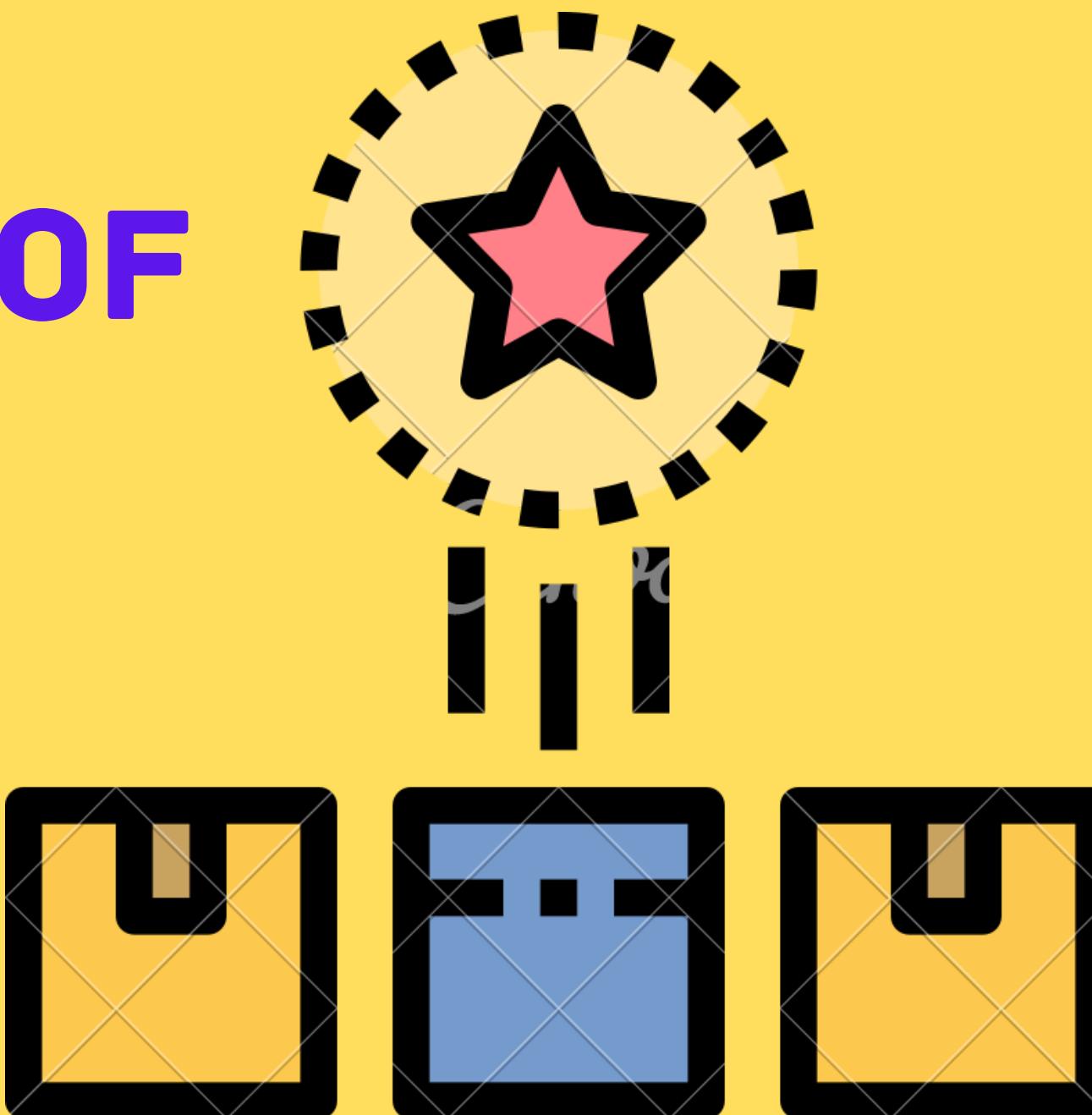
Proposed work:

- 1. Design a user-friendly interface for the online judge.**
- 2. Develop a secure sandbox environment to compile and execute user-submitted code.**
- 3. Implement a feature to compare the output of user-submitted code with expected output to determine correctness.**
- 4. Provide feedback to the user on the correctness and efficiency of their solution, as well as their ranking compared to other users.**
- 5. Limit access to external resources to ensure fairness.**

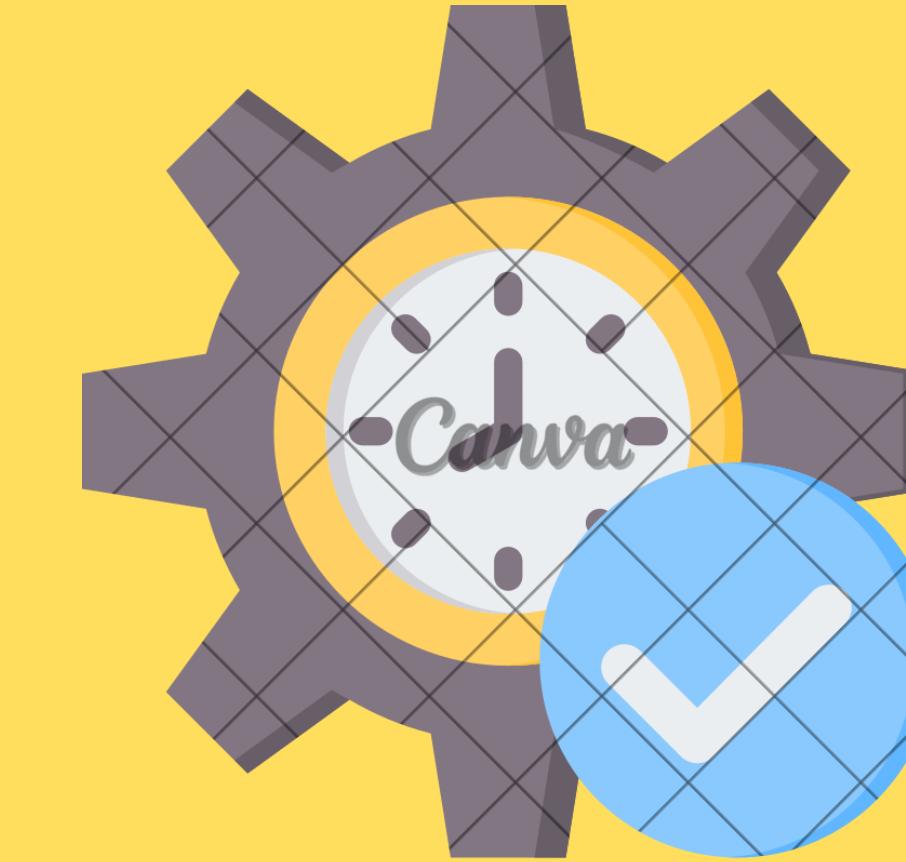
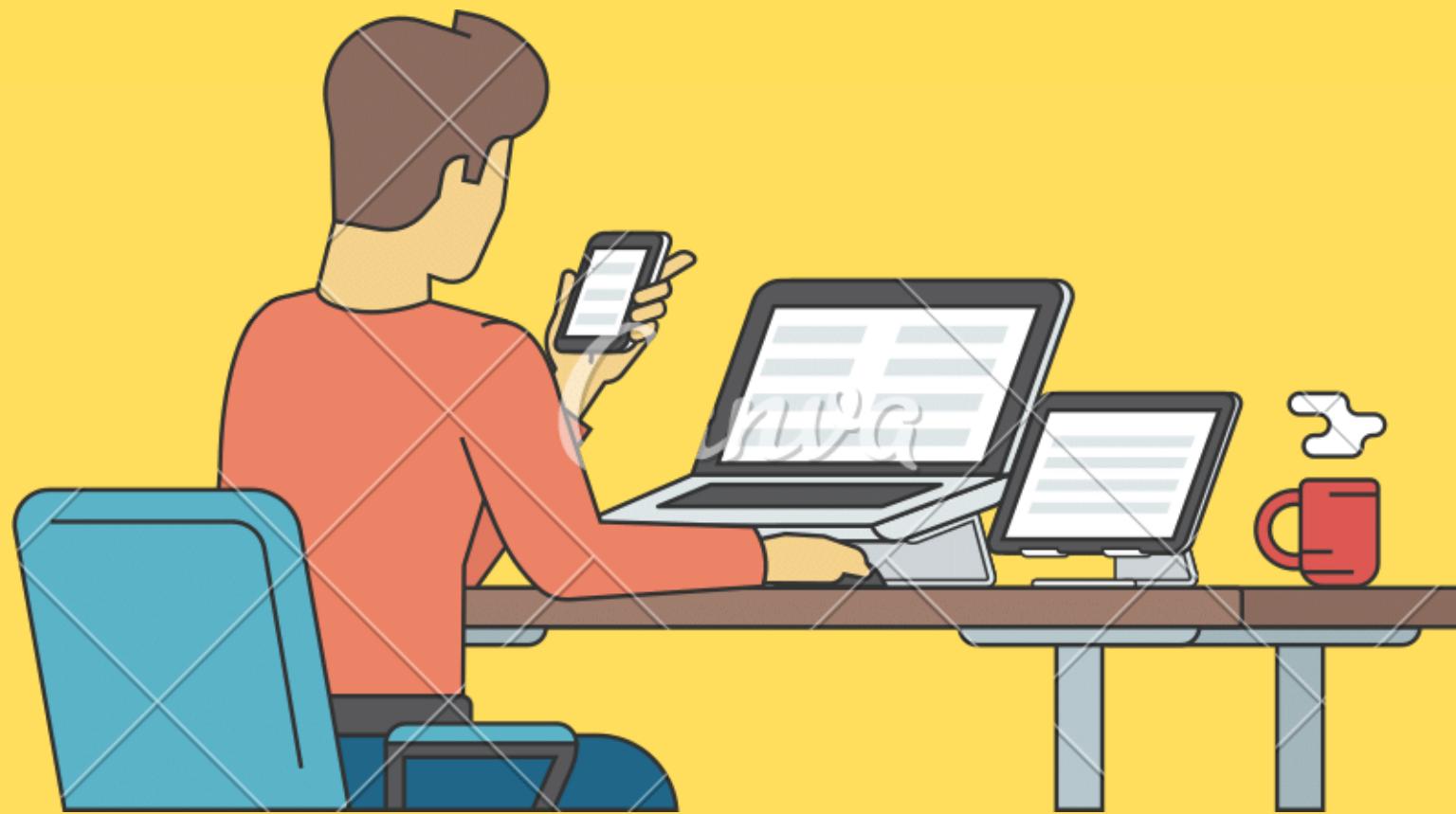
Methodology:

- 1. Use test cases to verify the correctness of user-submitted code.**
- 2. Measure the execution time and memory usage of the submitted code to evaluate efficiency.**
- 3. Provide feedback to the user on their performance in terms of correctness and efficiency.**
- 4. Implement a system for reporting and handling cheating or plagiarism.**
- 5. Continuously update the online judge with new problems and features to keep users engaged and challenged.**
- 6. Test and debug the online judge thoroughly to ensure it functions as intended.**

NOVELTY OF THE PROJECT



- Incorporating machine learning or artificial intelligence to automatically generate test cases or evaluate code efficiency.**
- Developing new types of problem sets that utilize emerging technologies or focus on social impact and sustainability.**
- Integrating gamification elements to increase engagement and motivation among users.**
- Providing personalized feedback and guidance to help users improve their programming skills and tackle more challenging problems.**



REAL TIME USAGE

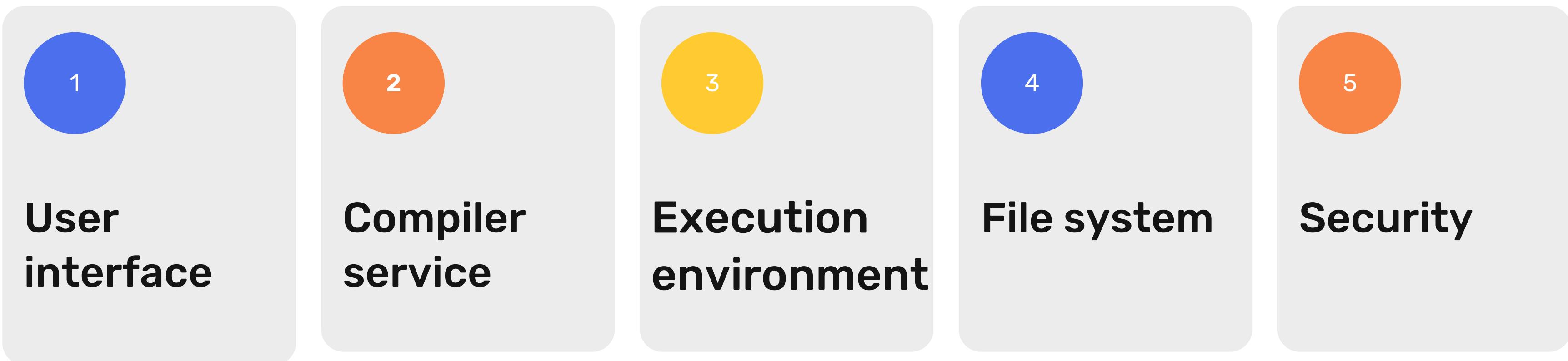
- **Code submission and evaluation:** Competitors can submit their code to the online judge, which then evaluates the code and returns a verdict indicating whether the code is correct or not.
- **Contest hosting:** Online judges are often used to host programming contests, where participants can compete against each other by solving a set of programming problems within a given time limit.
- **Practice and learning:** Many online judges offer a wide range of programming problems with varying difficulty levels, allowing users to practice their skills and learn new programming concepts.
- **Performance benchmarking:** Online judges can be used to benchmark the performance of different algorithms and programming languages, helping programmers to choose the most efficient approach for a given problem.

HARDWARE & SOFTWARE REQUIREMENTS

- **HTML**
- **CSS**
- **JavaScript**
- **Python**
- **Django**
- **Tailwind**
- **Editor(vs Code)**



Overall system architecture diagram.





LITERATURE REVIEW

Online judges are widely used platforms for hosting programming competitions and practice sessions. They provide a convenient way for programmers to improve their algorithmic skills and track their progress over time. Several online judges are available, and each has its strengths and weaknesses.

Some of the most popular online judges include Codeforces, HackerRank, CodeChef, Topcoder, and AtCoder. These platforms offer a wide range of programming challenges, from beginner-level problems to advanced algorithmic puzzles. They also provide a supportive community of programmers who can help solve problems and provide feedback on solutions.

Online judges typically support a wide range of programming languages, including C++, Java, Python, and many others. They also provide tools for testing and debugging code, as well as resources for learning new algorithms and data structures.

One of the key features of online judges is the ability to submit code and receive feedback on its correctness and performance. This can be a powerful way to learn from mistakes and improve programming skills over time. Some online judges also offer rewards and rankings for top performers, adding an element of competition and motivation to the learning process.

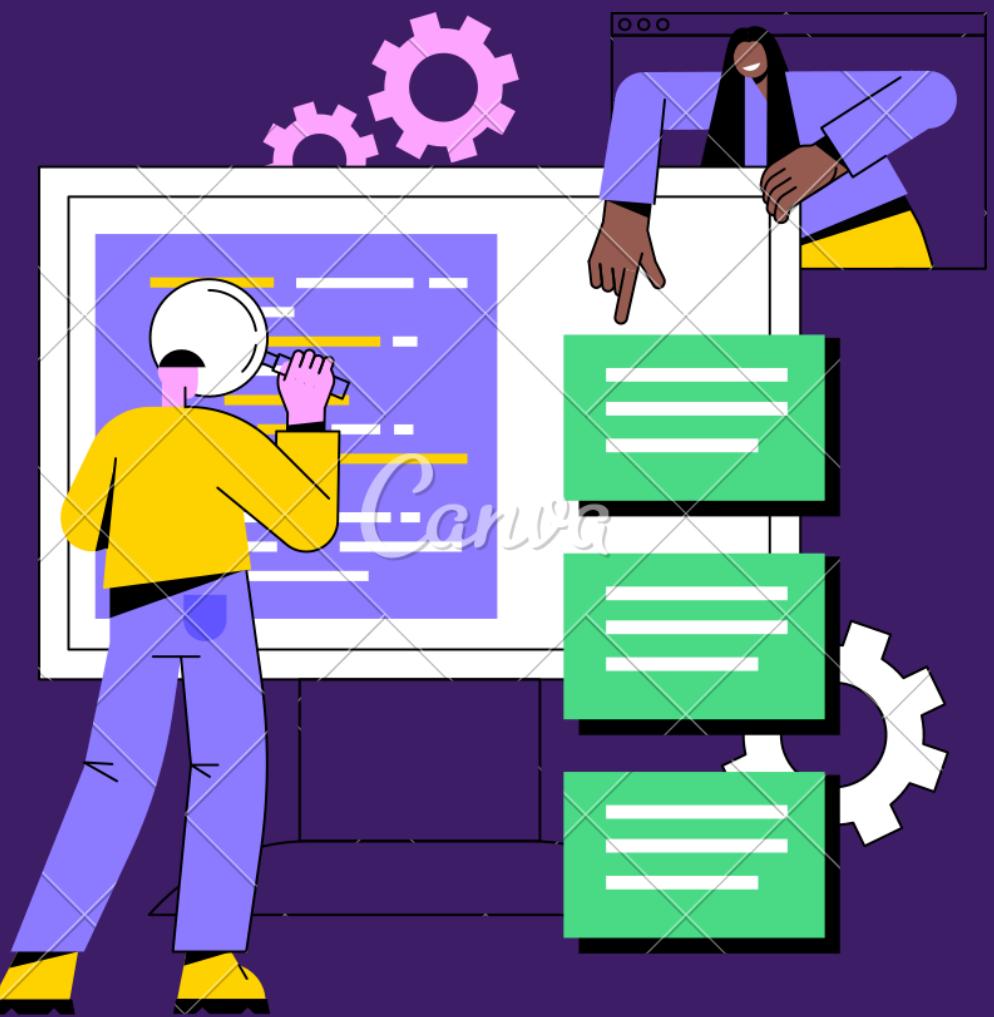
Overall, online judges are an invaluable resource for programmers looking to improve their algorithmic skills and compete in programming competitions. They provide a supportive community, challenging problems, and tools for feedback and improvement, all in a convenient online format.

Module Description

Problem statement and test cases: This module is responsible for creating, uploading, and managing the problems that will be used in the contests.

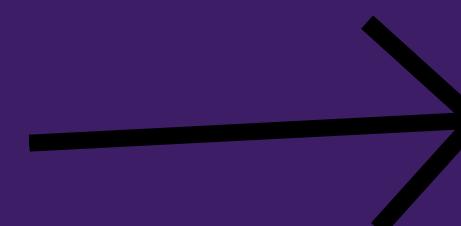
Submissions and judging: This module handles the submission of solutions by users, and runs the test cases to determine the correctness of the solutions.

Administration: This module provides tools for the administrators of the platform to manage and monitor the system, such as the ability to create and manage contests, and to view and manage submissions.



Module Splitup

Creating frontend
for website



using python and
importing Django in
backend for Online
Judge

integrating the
Online Judge
in the website

IMPLEMENTATION AND CODING

File Edit Selection View Go Run Terminal Help problemset.html - Frontend - Visual Studio Code

DIRECTORY

FRONTEND

- assets
- node_modules
- How-to-Get-Started-with-Compet...
- index.html
- logo.jpeg
- main.css
- package-lock.json
- package.json
- postcss.config.js
- problemset.html
- python-2.gif
- question1.html
- question2.html
- question3.html
- question4.html
- question5.html
- tailwind.config.js

problemset.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta http-equiv="X-UA-Compatible" content="IE=edge">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Algo Arenac</title>
    <link rel="stylesheet" href="main.css">
<style>
    .navbar li{
        position: relative;
    }
    .navbar li a{
        color: #f1f1f1;
        transition: 0.3s ease;
    }
    .navbar li a:hover,
    .navbar li a.active{
        color: #e08817;
    }
    .navbar li a.active::after,
    .navbar li a:hover::after{
        content: "";
        width: 100%;
        height: 2px;
        background: #e08817;
        position: absolute;
        bottom: -4px;
        left: 0px;
    }
    table{
        border-collapse: collapse;
        width: 100%;
    }
    th,td{
        border: 2px solid green;
        padding: 15px;
    }
</style>
```

✓ Minimap

✓ Render Characters

Vertical size

Slider

OUTLINE

TIMELINE

Line 42, Col 7 Spaces: 4 UTF-8 CR LF HTML Go Live

ENGLISH IN 11:35 PM 14-02-2023

The screenshot shows the PyCharm IDE interface with the following details:

- Project Tree:** On the left, the project structure is displayed under "Compiler". It includes "Compiler", "OnlineCompiler", "pycompiler", "migrations", "templates", "db.sqlite3", "file.txt", and "manage.py".
- Code Editor:** The main window shows the file "views.py" with the following Python code:

```
from django.shortcuts import render
import sys
import io
import time
import tracemalloc

def index(request):
    return render(request, 'index.html')

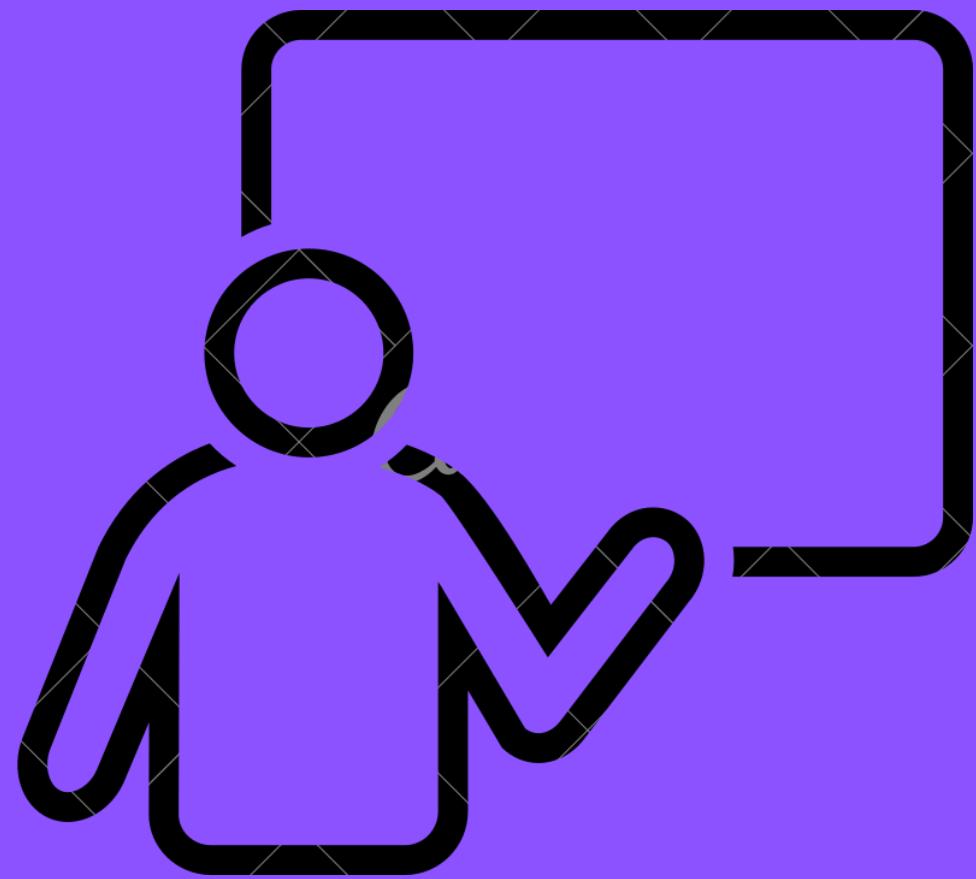
def runcode(request):
    original_stdout = None
    output = None
    execution_time = 0
    memory_usage = 0
    memory_usage_mb = 0

    if request.method == "POST":
        codeareaadata = request.POST['codearea']
        inputdata = request.POST.get('input', '')

    try:
        # Save original standard output reference
        original_stdout = sys.stdout
        sys.stdout = open('file.txt', 'w') # Change the standard output to the file we created

        # Redirect stdin to the input string
        sys.stdin = io.StringIO(inputdata)

runcode()
```
- Toolbars and Status Bar:** The top bar includes "File", "Edit", "View", "Navigate", "Code", "Refactor", "Run", "Tools", "VCS", "Window", and "Help". The status bar at the bottom shows "14:18 CRU UTF-8 4 spaces Python 3.11 (Compiler)".
- Bottom Navigation:** Icons for "Version Control", "Python Packages", "1000", "Python Console", "Problems", "Terminal", and "Services".
- System Tray:** Icons for "ENG IN", "Wi-Fi", "Signal", and "Battery" status.



DEMO VIDEO.

File Edit Selection View Go Run Terminal Help package.json - Frontend - Visual Studio Code

EXPLORER problemset.html question1.html question2.html package.json X question3.html question4.html question5.html

FRONTEND assets node_modules How-to-Get-Started-with-Competit... index.html logo.jpeg main.css package-lock.json package.json postcss.config.js problemset.html python-2.gif question1.html question2.html question3.html question4.html question5.html tailwind.config.js

```
1  {
2    "name": "codeforces",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": [
7      "start": "vite"
8    ],
9    "author": "",
10   "license": "ISC",
11   "devDependencies": [
12     "autoprefixer": "^10.4.13",
13     "postcss": "^8.4.21",
14     "tailwindcss": "^3.2.6",
15     "vite": "^4.1.1"
16   ]
17 }
18 }
```

Front-End

Snap shot of your project.

ALGO ARENA 

1

PROBLEMS

Welcome to ALGO ARENA

Welcome to Algo Arena, a platform for competitive programming and algorithm development, providing problems based on Data Structures and Algorithms (DSA).

How to Get Started

Competitive 1D Programming



Key feature: Online judge system that evaluates submitted code against test cases and provides instant feedback on correctness.

ALGO ARENA A

[HOME](#)[PROBLEMSET](#)[PROBLEMS](#)[SUBMIT](#)

Name	Code	Successful Submissions	Accuracy
Make it Divisible	MAKEDIV	4477	64.21
Indivisible Permutation	INDIPERM	3563	31.62
Minimum Number of Pizzas	MINPIZZAS	3367	53.44
Distinct Pair Sums	MANYSUMS	2971	53.55
Turn It	NFS	2685	49.1

© 2023 ALGO ARENA



Search



ALGO ARENA A

[PROBLEMS](#) [SUBMIT](#)

Make it Divisible

Given an integer N, help Ram in finding an N-digit odd positive integer odd positive integer X such that X is divisible by 3 but not by 9.

Note: There should not be any leading zeroes in X. In other words, 003 is not a valid 3-digit odd positive integer.

Input Format

- The first line of input contains a single integer T, denoting the number of testcases. The description of the T testcases follows.
- The first and only line of each test case contains a single integer N, denoting the number of digits in X.

Output Format

For each testcase, output a single line containing an N-digit odd positive integer X in decimal number system, such that X is divisible by 3 but not by 9.

Constraints

- 1 ≤ T ≤ 500
- 1 ≤ N ≤ 10⁴
- The sum of N over all test cases does not exceed 10⁵

Sample 1:

Input	Output
3	3
1	15
2	15
3	123



Algo Arena's Online Judge

Status: Correct Answer

Enter your code here:

```
t = int(input())
```

```
while t > 0:
```

```
n = int(input())
```

File n%2 000

```
for i in range(1,n+1,2)
    print(i+1,end=" ")
```

Enter Input:

Execution Time: 0.0010068416595458984 seconds
Memory Usage: 0.06695 MB

Output

四
卷之三

Testing

File Edit Selection View Go Run Terminal Help package.json - Frontend - Visual Studio Code

RUN AND ... No Cor ... package.json question1.html question2.html question3.html question4.html question5.html package.json

VARIABLES

```
package.json > {} scripts > start
1  {
2    "name": "codeforces",
3    "version": "1.0.0",
4    "description": "",
5    "main": "index.js",
6    "scripts": {
7      "start": "vite"
8    },
9    "author": "",
10   "license": "ISC",
11   "devDependencies": {
12     "autoprefixer": "^10.4.13",
13   }
14 }
```

WATCH PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JavaScript Debug Terminal

VITE v4.1.1 ready in 340 ms

- Local: <http://localhost:5173/>
- Network: use --host to expose
- press h to show help

CALL STACK

- Node.js Process: np... RUNNING
- vite.js [21548] RUNNING

LOADED SCRIPTS

BREAKPOINTS

- Caught Exceptions
- Uncaught Exceptions

Ln 7, Col 20 Spaces: 2 UTF-8 JSON Go Live

11:36 PM 14-02-2023



Result and Discussion

ALGO ARENA 

1

PROBLEMS

Welcome to ALGO ARENA

Welcome to Algo Arena, a platform for competitive programming and algorithm development, providing problems based on Data Structures and Algorithms (DSA).

How to Get Started

Competitive Programming



Key feature: Online judge system that evaluates submitted code against test cases and provides instant feedback on correctness.

ALGO ARENA A

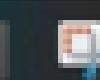
[HOME](#)[PROBLEMSET](#)[PROBLEMS](#)[SUBMIT](#)

Name	Code	Successful Submissions	Accuracy
Make it Divisible	MAKEDIV	4477	64.21
Indivisible Permutation	INDIPERM	3563	31.62
Minimum Number of Pizzas	MINPIZZAS	3367	53.44
Distinct Pair Sums	MANYSUMS	2971	53.55
Turn It	NFS	2685	49.1

© 2023 ALGO ARENA



Search

ENG
IN11:38 PM
14-02-2023

ALGO ARENA A

[PROBLEMS](#) [SUBMIT](#)

Make it Divisible

Given an integer N, help Ram in finding an N-digit odd positive integer odd positive integer X such that X is divisible by 3 but not by 9.

Note: There should not be any leading zeroes in X. In other words, 003 is not a valid 3-digit odd positive integer.

Input Format

- The first line of input contains a single integer T, denoting the number of testcases. The description of the T testcases follows.
- The first and only line of each test case contains a single integer N, denoting the number of digits in X.

Output Format

For each testcase, output a single line containing an N-digit odd positive integer X in decimal number system, such that X is divisible by 3 but not by 9.

Constraints

- 1 ≤ T ≤ 500
- 1 ≤ N ≤ 10⁴
- The sum of N over all test cases does not exceed 10⁵

Sample 1:

Input	Output
3	3
1	15
2	15
3	123



Algo Arena's Online Judge

Status: Correct Answer

Enter your code here:

```
t = int(input())
```

```
while t > 0:
```

```
n = int(input())
```

Page 2

```
for i in range(1,n+1,2)
    print(i+1,end=" ")
```

Enter Input:

Execution Time: 0.0010068416595458984 seconds
Memory Usage: 0.06695 MB

Output

四
卷之三

CONCLUSION

In conclusion, an online judge is an essential tool for competitive programming. It provides a platform for programmers to test their skills and knowledge, as well as a means of evaluation for their solutions. With its vast collection of problems and challenges, online judges have become a popular way of honing programming skills and keeping up with the latest developments in the field. However, it is important to note that online judges are just a tool, and success in competitive programming ultimately depends on the individual's dedication and hard work. With that said, we encourage all aspiring programmers to take advantage of online judges and continue to improve their skills through regular practice and participation in online programming competitions.



Thank
you!

