BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC PHENIKAA

Project Report

Topic:

“Introduction to the programming skill assessment system DMOJ”

Instructor : Ph.D. Vũ Đức Minh

Students	ID
1. **Nguyễn Gia Hào Kiệt ©**	20010872
2. **Hoàng Tuấn Anh**	20010839
3. **Nguyễn Nam Khánh**	20010869
4. **Nguyễn Văn Hưng**	20010864

HÀ NỘI, 12/2022

Table of Contents

I. General introduction about the system

What's DMOJ?

DMOJ (short for "Don't Memorize, Just Optimize") is a website that hosts programming contests and challenges. It was created in 2013 as a platform for students to practice their coding skills and prepare for programming competitions.

Users can browse a list of available problems, which are organized by difficulty and topic. Each problem includes a prompt that describes the task that the user must complete and provides input and output specifications. Users can write and submit code in a variety of programming languages, and the website will automatically test their submission against a set of test cases to determine if it is correct.

In addition to hosting programming challenges, DMOJ also has a number of other features. It has a leaderboard that ranks users based on their performance in past contests, and it includes a discussion forum where users can ask questions and share tips and strategies. The website also provides resources for educators, such as curriculum guides and lesson plans.

Overall, DMOJ is a useful resource for programmers of all skill levels to practice their coding skills and prepare for programming competitions. It is well-organized and user-friendly, making it easy for users to find and solve problems.

II. Structure of the DMOJ system

The architecture of DMOJ is designed to handle a large number of submissions and provide fast and accurate judgements. It consists of a number of components, including: Web server, Task queue, Judges, Database

1. **Web server**

   This component handles incoming HTTP requests from users and serves web pages to the user's browser. It is responsible for handling user authentication, displaying problem statements and submission forms, and displaying results and rankings.

2. **Task queue**

   This component is responsible for managing the submission queue and distributing submissions to the available judges. It ensures that submissions are judged in a fair and timely manner, and helps to balance the load on the judges.

3. **Judges**

   These are the components that actually run and test the submissions. They are responsible for compiling the code, running it with test inputs, and determining whether the output is correct. Judges can be run on a variety of platforms, including local machines, virtual machines, and cloud environments.

4. **Database**

   This component stores information about users, problems, submissions, and other data needed by the system. It is used to store and retrieve information needed by the other components of the system.

Overall, the architecture of DMOJ is designed to be scalable, reliable, and efficient, and to provide a high-quality platform for programming practice and competition.

III. The main functions and how it works of the DMOJ system

1. How DMOJ work?

DMOJ is a platform that provides online programming judges, similar to websites such as HackerRank and Codechef. It allows users to solve programming challenges, or "problems," and submit their solutions to be run against a set of test cases. The platform then provides feedback on the solution's correctness and efficiency.

Here's a general overview of how the DMOJ platform works:

1. Users register for an account and can browse through a list of available problems. Each problem has a description, input and output specifications, and a set of test cases.
2. Users can write and submit their solution to a problem using one of the supported programming languages.
3. When a solution is submitted, it is automatically compiled and run against the provided test cases. The platform checks the output of the solution against the expected output to determine if it is correct.
4. The platform provides feedback on the solution's correctness and efficiency, such as the time it took to run and the amount of memory it used. Users can view their submission history and compare their solutions to other users' solutions.

DMOJ also supports various features such as problem tagging, problem rating, and leaderboards, which allow users to more easily find and compare problems and solutions. It also has a community forum where users can discuss problems, ask for help, and share tips and resources.

2. The main functions of DMOJ system

On the DMOJ website, each programming problem has a description that outlines the task that the user is expected to solve. The description typically includes information such as:

- The input and output format of the solution
- Any constraints or limitations that apply to the solution (e.g., time or memory limits)

Example input and output to help the user understand the problem

For example, here is a sample problem description from DMOJ:

Problem:

Write a program that reads in a positive integer n and then outputs the sum of the squares of the first n positive integers.

Input:

The input will consist of a single integer n ($1 \leq n \leq 1000$).

Output:

The output should consist of a single integer, the sum of the squares of the first n positive integers.

Example Input:

Copy code

```
5
```

Example Output:

Copy code

```
55
```

Explanation:

The sum of the squares of the first 5 positive integers is 1^2 + 2^2 + 3^2 + 4^2 + 5^2 = 1 + 4 + 9 + 16 + 25 = 55.

In DMOJ (Don Mills Online Judge), the program description is the written description of a programming problem that is provided to users when they select a problem to solve.

The program description typically includes the following information:

1. The problem statement: This explains the task that the user is expected to accomplish with their program. The problem statement may include examples of input and output to help the user understand the requirements of the problem.

2. Input specification: This describes the format of the input that the program should expect. This may include the number and types of input values, as well as any constraints on those values.

3. Output specification: This describes the format of the output that the program should produce. This may include the number and types of output values, as well as any constraints on those values.

4. Example input and output: This provides examples of input and corresponding output that the program should produce. These examples are used to test the user's solution and ensure that it is correct.

3. The workflow of the DMOJ

The workflow of the DMOJ website can be described as follows:

1. A user accesses the website using a web browser and sends a request to the server via the internet.

2. The NGINX web server receives the request and forwards it to the Django web application on the server.

3. The Django web application processes the request and returns data to the NGINX web server.

4. The NGINX web server sends the returned data to the user.

5. During the request processing, if necessary, the Django web application can call asynchronous tasks using the Celery tool to process different tasks. For example, if a user submits a source code assignment for grading, the web application can send this task to Celery to process while continuing to handle other requests.

6. When the asynchronous tasks are completed, the results can be stored in the database and can be accessed by the user or other web applications.

7. This process will repeat until the user closes the browser or leaves the website.

IV. Setting up the DMOJ system

1. Overview of building a DMOJ system

Web DMOJs can be built using a variety of tools. Some of the tools that can be used to create this website include:

1. Programming languages: Web DMOJs can be built in many different programming languages, such as Python, Java, Ruby or C++.

2. Framework: You can use frameworks like Django, Rails or Spring to build DMOJ websites.

3. Data storage engine: Web DMOJ can use a database like MySQL, PostgreSQL or MongoDB to store data.

4. Source code management tools: Tools like Git or Mercurial can be used to manage and track changes in the source code of a website.

5. Grading Tools: Web DMOJs can use tools like Judge0 or PolyJudge to grade programming assignments and answer questions.

6. User Interface Tools: Tools like Bootstrap or Materialize can be used to create the user interface for a DMOJ website.

2. Setting Overview

To deploy Web DMoj, you will need to follow these steps:

1. Install the required dependencies: Web DMoj requires Python 3.5 or higher, and uses the Django web framework. You will need to install these dependencies on your server.

2. Clone the Web DMoj repository: Web DMoj is open source and available on GitHub. You can clone the repository using Git by running the following command:

Copy code:

```
git clone https://github.com/DMOJ/web.git
```

3. Install the required Python packages: Navigate to the directory where you cloned the Web DMoj repository and run the following command to install the required Python packages:

Copy code:

```
pip install -r requirements.txt
```

4. Create a configuration file: Web DMoj uses a configuration file to store your site-specific settings. You can create a configuration file by copying the example configuration file and modifying it to match your settings:

Copy code

```
cp site_config/example.py site_config/local.py
```

5. Run the Django migrations: Web DMoj uses a database to store data such as user accounts and problem information. You will need

to create the required database tables by running the Django migrations:
Copy code:

```
python manage.py migrate
```

6. Collect the static files: Web DMoj uses static files such as CSS and JavaScript for the frontend. You will need to collect these static files by running the following command:
Copy code:

```
python manage.py collectstatic
```

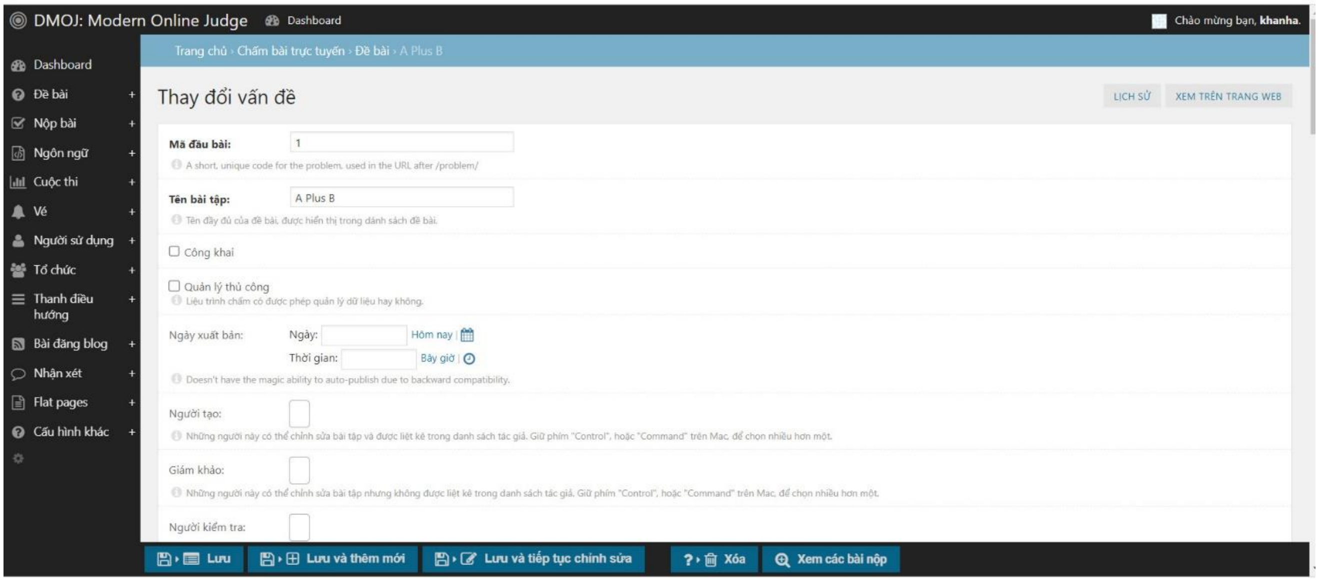
7. Run the development server: You can test your installation by running the development server with the following command:
Copy code:

```
python manage.py runserver
```

This will start the development server on your local machine. You can access the site by visiting http://127.0.0.1:8000 in your web browser.

All step to Setting and Install DMOJ:
https://docs.dmoj.ca/#/site/installation

V. Test



Thay đổi vấn đề

Mã đầu bài: 1

A short, unique code for the problem, used in the URL after /problem/

Tên bài tập: A Plus B

Tên đầy đủ của đề bài, được hiển thị trong danh sách đề bài.

□ Công khai

□ Quản lý thủ công

Liệu trình chấm có được phép quản lý dữ liệu hay không.

Ngày xuất bản: Ngày: Hôm nay | Thời gian: Bây giờ |

Doesn't have the magic ability to auto-publish due to backward compatibility.

Người tạo:

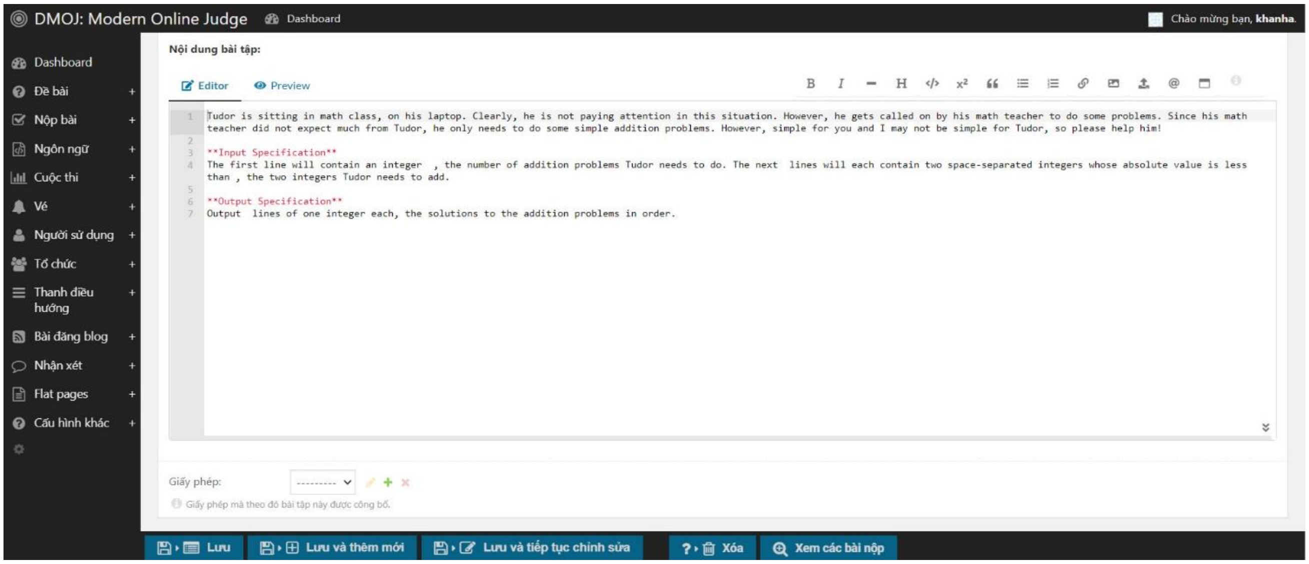
Những người này có thể chỉnh sửa bài tập và được liệt kê trong danh sách tác giả. Giữ phím "Control", hoặc "Command" trên Mac, để chọn nhiều hơn một.

Giám khảo:

Những người này có thể chỉnh sửa bài tập nhưng không được liệt kê trong danh sách tác giả. Giữ phím "Control", hoặc "Command" trên Mac, để chọn nhiều hơn một.

Người kiểm tra:

Picture 1: Create assignments



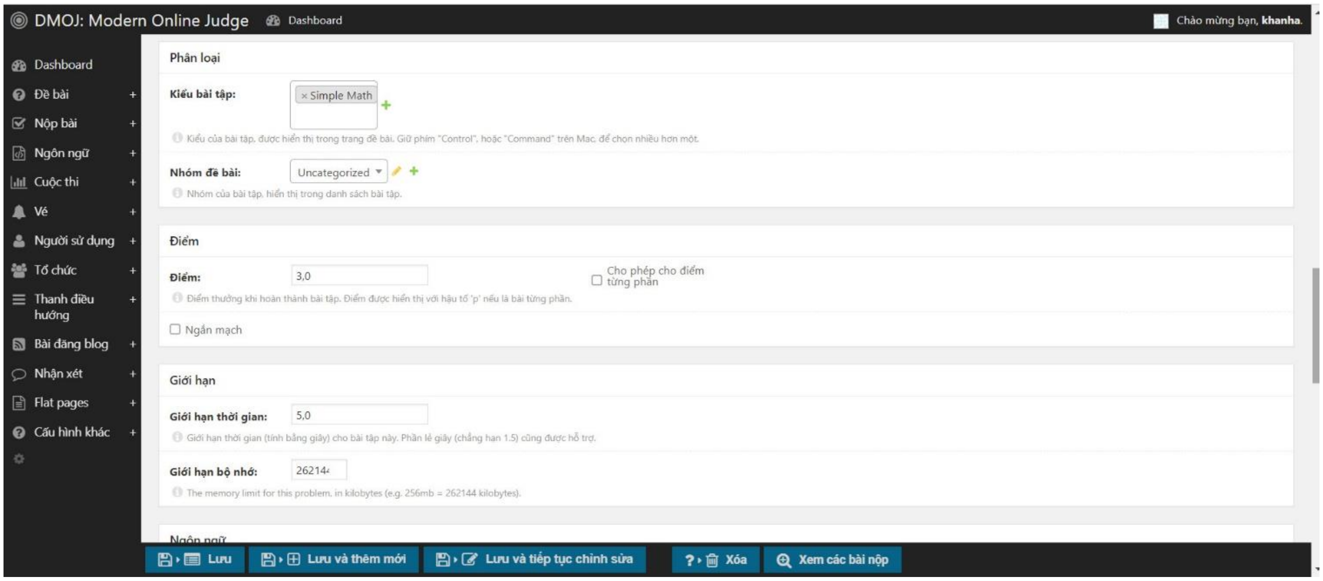
Nội dung bài tập:

```
Tudor is sitting in math class, on his laptop. Clearly, he is not paying attention in this situation. However, he gets called on by his math teacher to do some problems. Since his math teacher did not expect much from Tudor, he only needs to do some simple addition problems. However, simple for you and I may not be simple for Tudor, so please help him!

**Input Specification**
The first line will contain an integer  , the number of addition problems Tudor needs to do. The next  lines will each contain two space-separated integers whose absolute value is less than , the two integers Tudor needs to add.

**Output Specification**
Output  lines of one integer each, the solutions to the addition problems in order.
```

Giấy phép: ---------

Giấy phép mà theo đó bài tập này được công bố.

Picture 2: Write assignment request



Phân loại

Kiểu bài tập: × Simple Math

Kiểu của bài tập, được hiển thị trong trang đề bài. Giữ phím "Control", hoặc "Command" trên Mac, để chọn nhiều hơn một.

Nhóm đề bài: Uncategorized

Nhóm của bài tập, hiển thị trong danh sách bài tập.

Điểm

Điểm: 3,0 □ Cho phép cho điểm từng phần

Điểm thưởng khi hoàn thành bài tập. Điểm được hiển thị với hậu tố 'p' nếu là bài từng phần.

□ Ngắn mạch

Giới hạn

Giới hạn thời gian: 5,0

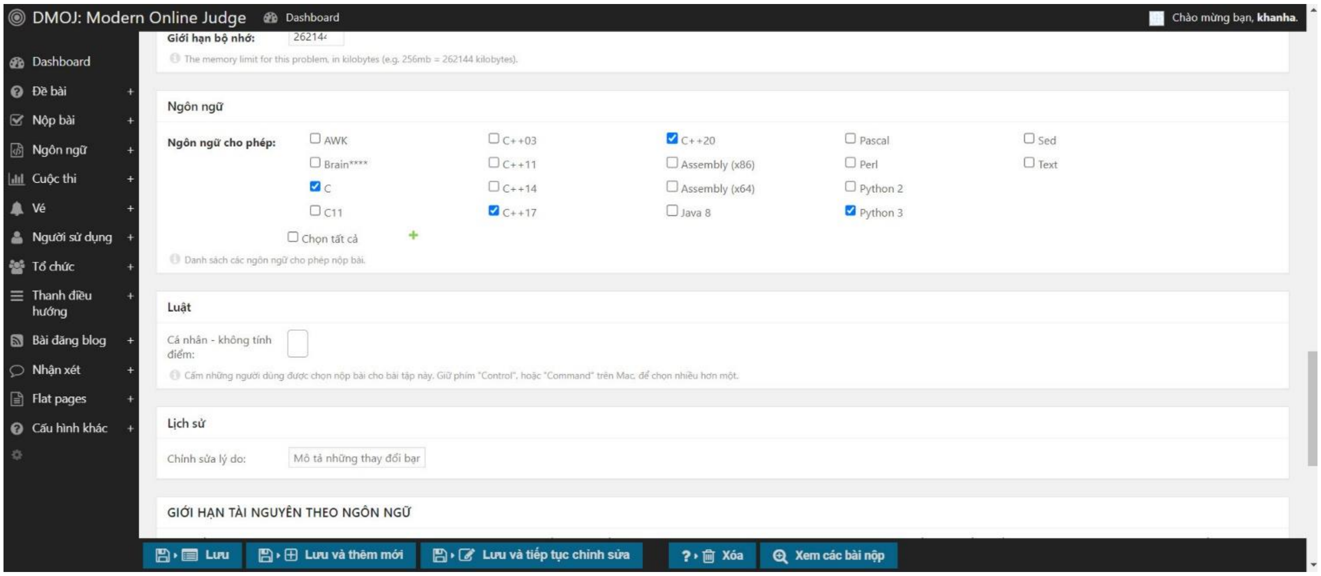
Giới hạn thời gian (tính bằng giây) cho bài tập này. Phần lẻ giây (chẳng hạn 1.5) cũng được hỗ trợ.

Giới hạn bộ nhớ: 26214

The memory limit for this problem, in kilobytes (e.g. 256mb = 262144 kilobytes).

Ngôn ngữ

Picture 3: Classification and grades



Giới hạn bộ nhớ: 26214

The memory limit for this problem, in kilobytes (e.g. 256mb = 262144 kilobytes).

Ngôn ngữ

Ngôn ngữ cho phép:

□ AWK
□ Brain****
☑ C
□ C11
□ C++03
□ C++11
□ C++14
☑ C++17
☑ C++20
□ Assembly (x86)
□ Assembly (x64)
□ Java 8
□ Pascal
□ Perl
□ Python 2
☑ Python 3
□ Sed
□ Text

□ Chọn tất cả

Danh sách các ngôn ngữ cho phép nộp bài.

Luật

Cá nhân - không tính điểm:

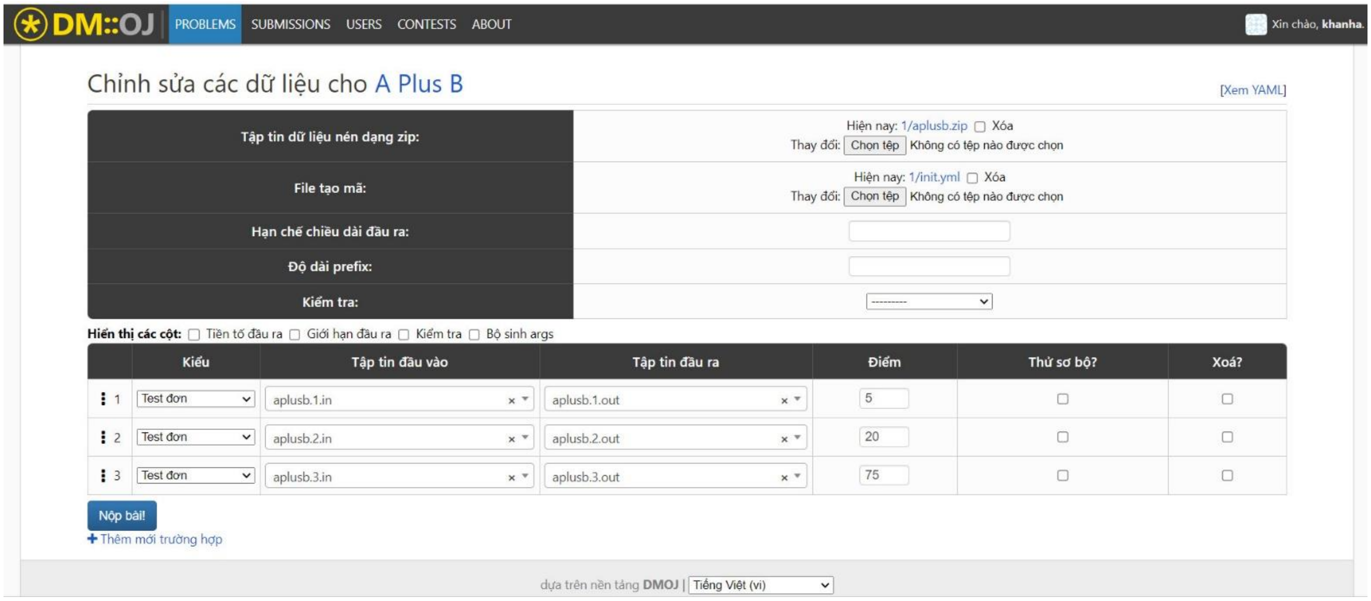
Cấm những người dùng được chọn nộp bài cho bài tập này. Giữ phím "Control", hoặc "Command" trên Mac, để chọn nhiều hơn một.

Lịch sử

Chỉnh sửa lý do: Mô tả những thay đổi bạ

GIỚI HẠN TÀI NGUYÊN THEO NGÔN NGỮ

Picture 4: Select the programming language



Chỉnh sửa các dữ liệu cho A Plus B

Tập tin dữ liệu nén dạng zip:	Hiện nay: 1/aplusb.zip □ Xóa Thay đổi: Chọn tệp Không có tệp nào được chọn
File tạo mã:	Hiện nay: 1/init.yml □ Xóa Thay đổi: Chọn tệp Không có tệp nào được chọn
Hạn chế chiều dài đầu ra:	
Độ dài prefix:	
Kiểm tra:	---------

Hiển thị các cột: □ Tiền tố đầu ra □ Giới hạn đầu ra □ Kiểm tra □ Bộ sinh args

	Kiểu	Tập tin đầu vào	Tập tin đầu ra	Điểm	Thử sơ bộ?	Xoá?
1	Test đơn	aplusb.1.in	aplusb.1.out	5	□	□
2	Test đơn	aplusb.2.in	aplusb.2.out	20	□	□
3	Test đơn	aplusb.3.in	aplusb.3.out	75	□	□

Picture 5: Download input and output test for grading

VI. Conclusion

In conclusion, DMOJ is a valuable resource for individuals interested in computer programming and competitive programming. It offers a wide range of programming challenges and resources for users to improve their skills and test their knowledge. Additionally, the platform's robust judging system and active community make it a reliable and enjoyable platform for both novice and experienced programmers.

Overall, DMOJ is a valuable tool for anyone looking to improve their programming abilities and participate in online coding competitions.

VII. References

1. Stack Overflow

https://stackoverflow.com/

2. **DMOJ Docs**
   https://docs.dmoj.ca/#/
3. **HackerRank**
   https://www.hackerrank.com/
4. **Codechef**
   https://www.codechef.com/