

# A Level Programming Project Report

Junrong Chen

September 22, 2021

# Contents

|          |  |          |
|----------|--|----------|
| <b>1</b> | <b>Analysis</b>  | <b>3</b> |
| 1.1      | Problem Identification . . . . .                       | 3        |
| 1.2      | Stakeholders . . . . .                                 | 3        |
| 1.2.1    | Computer Science teachers . . . . .                    | 3        |
| 1.2.2    | Computer Science students . . . . .                    | 4        |
| 1.3      | Why it is suited to a computational solution . . . . . | 4        |
| 1.4      | Solve by computational methods . . . . .               | 4        |
| 1.4.1    | Thinking abstractly . . . . .                          | 4        |
| 1.4.2    | Thinking ahead . . . . .                               | 5        |
| 1.4.3    | Thinking procedurally and decomposition . . . . .      | 5        |
| 1.4.4    | Thinking concurrently . . . . .                        | 6        |
| 1.5      | Interview . . . . .                                    | 6        |
| 1.5.1    | Design the interview . . . . .                         | 6        |
| 1.5.2    | Conduct the interview . . . . .                        | 7        |
| 1.6      | Research . . . . .                                     | 10       |
| 1.6.1    | LeetCode . . . . .                                     | 10       |
| 1.6.2    | Codeforces . . . . .                                   | 15       |
| 1.7      | Features . . . . .                                     | 17       |
| 1.8      | Limitations . . . . .                                  | 17       |
| 1.9      | Hardware and software requirements . . . . .           | 19       |

|                                 |           |
|---------------------------------|-----------|
| 1.10 Success criteria . . . . . | 20        |
| <b>2 Design</b>                 | <b>21</b> |
| 2.1 Decomposition . . . . .     | 21        |
| <b>3 Development</b>            | <b>22</b> |
| <b>4 Evaluation</b>             | <b>23</b> |

# Chapter 1

## Analysis

### 1.1 Problem Identification

A Level Computer Science students need to learn many algorithms and data structures during the course. In the final exam, they need to write pseudocode to solve computational questions. Many students find it is hard to achieve a high score on those questions due to the lack of efficient training. The general method used by students to learn and revise for Computer Science is to attempt and self-mark past paper questions. This works well for ordinary questions. However, for the algorithm questions, different students may produce completely different code solutions. This makes their self-marking very unreliable. It is also too much work for the teacher to mark their solutions one by one. So, in the end, students do not know whether they get things right, and teachers do not know how the students perform and how they can help, especially in this lockdown online learning era where no direct contact between teachers and students is possible.

Both the students and the teachers are looking for a more efficient method to learn and practice.

### 1.2 Stakeholders

There are two types of stakeholders, Computer Science teachers and Computer Science students.

#### 1.2.1 Computer Science teachers

Computer Science teachers find it is difficult to monitor their students' ability to design and implement algorithms, so they cannot provide efficient help to

their students. This software allows them to create coding questions and send them to the students. After the students hand their solutions back, the software will automatically mark their answers and provide detailed statistical data with simple visualisations. This helps the teachers saving a lot of time and allows them to help the students better.

The stakeholder is Mr. Grimwood, who is an experienced A Level Computer Science teacher who teaches a Year 12 CS group and a Year 13 CS group.

### **1.2.2 Computer Science students**

Computer Science students find that they tend to lose mark on the algorithms coding questions, so they want more practice. But unlike ordinary questions, they may take a completely different approach towards the questions comparing to the mark scheme, so they do not know whether they get it correct. Students may also think they have got things right, but actually, they have made some mistakes. The software provides a free practice space that automatically marks their solutions and points out their mistakes in real-time. So the students can learn and revise more efficiently.

The stakeholders are Timofei and PCloud, they are both Year 13 students studying A Level Computer Science.

## **1.3 Why it is suited to a computational solution**

The original problem, ‘understand and mark a student’s answer’ is a very difficult question for a computer to solve. But I transform the question into ‘compare the output of the students’ code with pre-generated test cases’, which makes the problem solvable using a computational method, since a computer is good at ‘executing a piece of code’ and ‘comparing two strings’. This approach solves the ‘marking’ question from another angle, and makes the question suited to a computational solution.

## **1.4 Solve by computational methods**

### **1.4.1 Thinking abstractly**

In reality, students use pens and paper to write their code solutions. This can be simplified into a code editor, and the students can use their keyboards to type in the code. In this way, no ‘text scanning’ or ‘handwriting recognition’ is needed which makes the design and programming much easier. The code editor will also provide a better user experience. Features such as syntax highlighting cannot exist on paper but are possible in a code editor.

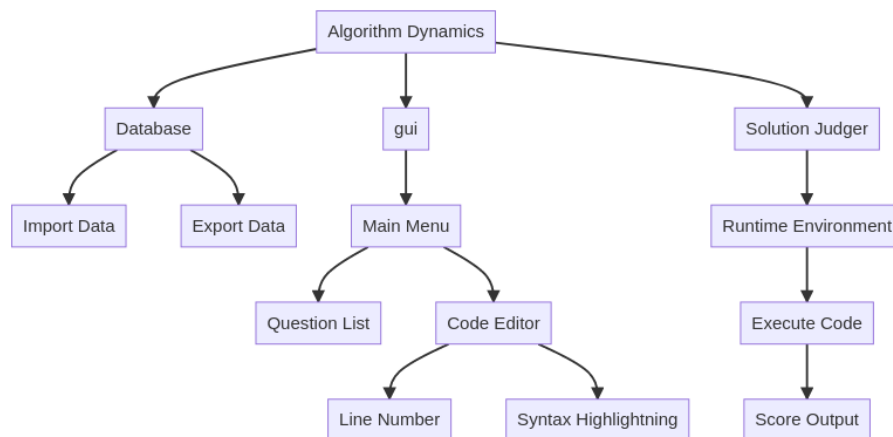
In reality, the students' answer is sent to a teacher to mark it against the mark scheme. The teacher needs to read the code line by line and check whether it is correct. This process is abstracted into a judger that marks the code against pre-generated test cases, which transforms a problem that originally cannot be solved by computational method into one which is very easy to be solved by a computer while saving time and costs. When creating a new question, instead of creating a mark scheme for marking, the teacher needs to provide test cases with the correct input and expected output. The judger will run the students' submissions with the input and check whether their output matches the expected one.

### 1.4.2 Thinking ahead

For teachers, the software requires them to enter questions and test cases. A question editor containing input boxes is needed for this purpose. For students, the software requires them to enter their code solutions. A code editor is needed for this purpose. A relational database is needed to store all the data. For all users, the software requires input data from mouse and keyboard to navigate between different windows and menus. Users will also need a monitor for the program to display all the information and outputs.

### 1.4.3 Thinking procedurally and decomposition

The program can be decomposed into several parts. Each part can be designed and maintained individually. Different components can interact with each other using custom APIs.



### 1.4.4 Thinking concurrently

When judging the students' solution, many test cases can be executed at the same time to reduce the judging time. The number of parallel judges need to be set carefully based on the user's hardware. Running too few test cases concurrently may result in a very long judging time while running too many test cases at the same time may use up computing resources and cause issues.

## 1.5 Interview

### 1.5.1 Design the interview

#### Interview for teachers

1. Do you find your students tend to lose marks on programming questions in exams?
2. Do you find marking the programming question takes a lot of time and effort?
3. Compare to the knowledge-based Computer System section, do you find it is more difficult to monitor students' skill level on the Algorithm and Programming section?
4. Have you ever heard about some online programming platforms?
5. Have you ever tried some of the online programming platforms?
6. If yes, what do you think about these platforms? Have you ever considered to use them for teaching and training?
7. Do you think a similar solution can help improve the efficiency of learning and training?
8. If no, do you think the idea of a software that can mark students' answers on programming questions and provide analysis data can help improve the efficiency of learning and training?
9. Do you have anything else to add?

Question 1 to 3 are a series of proof-of-concept questions, which I expect my stakeholders to answer 'Yes' to all of them. They confirm that the problem I am trying to solve actually exists and there is a need for such solution. Question 4 to 5 ask about the teachers' knowledge on existing solutions. Question 6 to 8 ask their experiences and opinions about these existing solutions, which gives me insights on the problems with existing solutions and how my solution can fit their need better.

## **Interview for students**

1. Do you find the programming questions difficult?
2. Do you find yourself lacking efficient practicing in algorithm designing and programming?
3. Have you ever heard about some online programming platforms?
4. Have you ever tried some of the online programming platforms?
5. If yes, what do you think about these platforms?
6. Do you think a similar solution can help you learning and practicing?
7. If no, do you think the idea of a software that provides coding questions and marks your answer instantly can help you learn and practice better?
8. Do you have anything else to add?

Question 1 and 2 are similar proof-of-concept questions to confirm such problem exists. The following questions ask about students' knowledge on existing solutions. If they have used a existing product before, I ask whether they think it helps. Otherwise, I ask whether they think it will be useful.

### **1.5.2 Conduct the interview**

#### **Computer Science teacher - Mr Grimwood**

1. Do you find your students tend to lose marks on programming questions in exams?  
They do. Many of them don't understand the algorithms.
2. Do you find marking the programming questions takes a lot of time and effort?  
Yes. Because some students produce partially correct answers, so it takes a lot of time to identify the correct part and award them the corresponding mark. Some students may take completely different approaches which takes a lot of effort to understand and mark them.
3. Do you find it is more difficult to monitor students' skill level on the Algorithm and Programming section and more difficult to provide sufficient help?  
Yes.
4. Have you ever heard about some online programming platforms?  
I have. Emm... But I forget the names.
5. If yes, have you ever tried some of the online programming platforms?  
I have.



6. If yes, what do you think about these platforms?

I think the idea is quite interesting and I find them actually work quite well.

7. Have you ever considered to use them for teaching and practicing?

No. Because most of them require a paid subscription, and their content is more likely to be something like 'Learning Python' which is irrelevant to the A Level Computer Science content.

8. Do you think a similar solution can help improve the efficiency of learning and training?

Yes. The students can learn at their own pace and they can keep practicing by themselves.

9. Do you have anything else to add?

No.

Mr. Grimwood has several valuable points here. He points out that the 'partially correct' answers are the most difficult ones to mark. For my solution, if a student submits a 'partially correct' code answer, then its output will certainly not match the expected output. Which means my solution might not be able to tell the difference between a 'partially correct' answer and a 'incorrect' answer. This is a potential limitation I need to watch out. He also says the price is one of his concern. My solution will be a free and open-source software, which will meet his need perfectly. By adding the function to create custom questions and share them with others, users will be able to create and find A Level Computer Science content, or any content easier. It is also a good idea for me to create some A Level Computer Science content come with the software to make it easier to use.

### **Computer Science student - PCloud**

1. Do you find the programming questions difficult?

I find some of them quite complex and difficult, especially the graph algorithms such as Dijkstra.

2. Do you find yourself lacking efficient practicing in algorithm designing and programming?

Absolutely. Although I code a lot in my spare time, but normal projects are quite different from the exam questions. There are not many past papers and exam-style questions for practicing, so I usually don't feel confident of those questions.

3. Have you ever heard about some online programming platforms?

Yes. Such as AcWing, LeetCode and TopCoder.

4. Have you ever tried some of the online programming platforms?

Yes. I am an active user of AcWing.

5. If yes, what do you think about these platforms?

I really enjoy the experience. They can provide an instant feedback for my submissions. It provides a very strong positive feedback when I solve a new question. I find myself learning faster and more efficient with such platform.

6. Do you think a similar solution can help you learning and practicing?

Absolutely. The existing platforms do not provide A Level related content. So if a software solution can be altered for A Level Computer Science course, that will really help a lot.

7. (\*) How do you think it should be optimized for A Level CS content?

You can add past exam questions practicing. Add a timed practice mode will be helpful.

8. Do you have anything else to add?

No.

PCloud confirms that such solution will help him learning and practicing more efficiently. The instant feedback of whether he gets the question correct is very important to him. Instead of sending the user's submission to a remote server, my solution should judge the user's answer on their own computer. This can avoid the unstabilities caused by the remote server's availability and the network connection. He also gives me some good ideas about the content. I can add past exam questions for users to do timed practice, which enables users to practice algorithms and exam techniques at the same time.

(TODO): Conduct and analysis the interview with Timofei

### **Computer Science student - Timofei**

1. Do you find the programming questions difficult?

Yes. I geneually lose marks because of some careless syntax mistakes I made.

2. Do you find yourself lacking efficient practicing in algorithm designing and programming?

Yes. I find I cannot find many materials to practice.

3. Have you ever heard about some online programming platforms?

Codewar. Something like that.

4. Have you ever tried some of the online programming platforms?

Yes.

5. If yes, what do you think about these platforms?

I think they are quite helpful. But I find their marking is too specific, if I get a single character wrong in my output, it gets marked incorrect.

6. Do you think a similar solution can help you learning and practicing?

Yes.

7. Do you have anything else to add?

No.

Timofei points out that the marking system in existing products is not very sensible. This may be a potential limitation of my solution as well. It is easy to directly compare the users' output and the expected output. But if they are different, it is difficult to figure out whether that difference is caused by a wrong code solution or just some formatting error. I can partially solve this by allowing the users to pre-test their code against examples before formal submissions, so they can check the output format.

## 1.6 Research

There are many coding training websites on the market, most of them share a similar idea, so I will investigate two of the most popular ones.

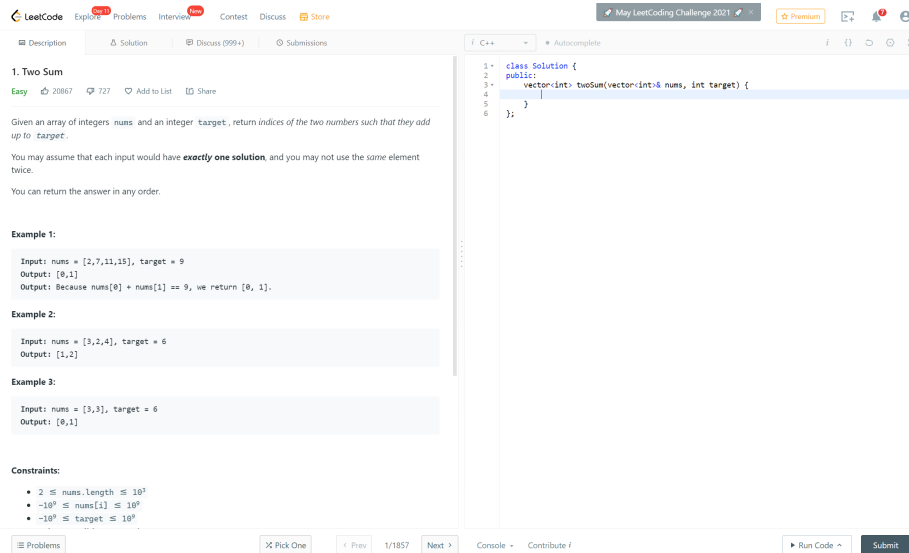
- LeetCode
- Codeforces

### 1.6.1 LeetCode

LeetCode is a platform for interview coding training, many large companies (Google, Facebook, ...) use it as a part of their interview.

LeetCode provides a database containing more than 1000 coding questions.

## Main coding layout



This is LeetCode's main coding area. The user's screen is split into two parts - the question section and the code editor for inputting answer. User can drag the splitter in the middle to adjust the size of each section.

The question section contains 4 tabs, 'Description' tab displays the content of the question. 'Solution' tab displays the solutions from the community. 'Discuss' tab displays the discussions in the community. 'Submission' tab lists user's previous submissions. Since I am not adding social functions in my solution, I will ignore the 'Solution' and 'Discuss' tab. Under the 'Description' tab, LeetCode provides the context of the question, followed by 3 examples, and constraints for this question. The examples allow the user to run and check their solution before formal submission for marking, this can help them avoid silly mistakes. My solution should also provide similar examples for each question. Under the 'Submission' tab, LeetCode records every history submission, so the user can revise old questions more efficiently. My solution should provide a similarly function as well.

The screenshot shows the LeetCode interface for the 'Two Sum' problem. The problem description on the left indicates a runtime of 8 ms and memory usage of 9 MB. The code editor on the right shows a C++ solution using a hash map. The submission table at the bottom left shows a list of submissions with their status, runtime, and memory usage. The 'Run Code' button is visible at the bottom right.

| Time Submitted   | Status       | Runtime | Memory | Language |
|------------------|--------------|---------|--------|----------|
| 05/11/2021 22:34 | Accepted     | 8 ms    | 9 MB   | cpp      |
| 02/06/2020 23:30 | Wrong Answer | N/A     | N/A    | cpp      |
| 01/31/2020 12:16 | Wrong Answer | N/A     | N/A    | cpp      |
| 01/31/2020 12:16 | Wrong Answer | N/A     | N/A    | cpp      |
| 01/31/2020 12:15 | Wrong Answer | N/A     | N/A    | cpp      |
| 01/31/2020 12:12 | Accepted     | 360 ms  | 9.2 MB | cpp      |
| 01/31/2020 12:12 | Wrong Answer | N/A     | N/A    | cpp      |

```

1  class Solution {
2  public:
3      vector<int> twoSum(vector<int>& nums, int target) {
4          vector<int> res;
5          unordered_map<int, int> hash;
6          for (int i = 0; i < nums.size(); i++)
7          {
8              int another = target - nums[i];
9              if (hash.count(another))
10             {
11                 res = vector<int>({hash[another], i});
12                 break;
13             }
14             hash[nums[i]] = i;
15         }
16         return res;
17     }
18 };

```

The code editor provides line number and syntax highlighting functions. User can change their programming language with a drop-down box. LeetCode supports all mainstream programming languages. My solution should be able to support multiple programming languages as well, which allows students with different background using it easily.

On the button, the user can ‘Run Code’ to test their code against the examples before submission, and then click the ‘Submit’ button to submit their solution formally.

The split view design is clean and handy. The user can see the question and write their solution on the same page without switching between different windows. The design of examples and the ‘Run Code’ button is useful as well. I can refer to LeetCode’s coding layout when designing my solution’s interface.

## Question database

Array940

String466

Hash Table332

Dynamic Programming331

Math319

Depth-First Search230

Sorti

Expand

All Topics

Algorithms

Database

Shell

Concurrency

Lists

Difficulty

Status

Tags

Search questions

Pick One

| Status | Title   | Solution | Acceptance | Difficulty | Frequency |
|--------|---|----------|------------|------------|-----------|
|        | 1189. Maximum Number of Balloons                |          | 62.7%      | Easy       |           |
|        | 1. Two Sum                                      |          | 47.6%      | Easy       |           |
|        | 2. Add Two Numbers                              |          | 36.8%      | Medium     |           |
|        | 3. Longest Substring Without Repeating Chara... |          | 32.2%      | Medium     |           |
|        | 4. Median of Two Sorted Arrays                  |          | 32.7%      | Hard       |           |
|        | 5. Longest Palindromic Substring                |          | 31.2%      | Medium     |           |
|        | 6. ZigZag Conversion                            |          | 39.6%      | Medium     |           |
|        | 7. Reverse Integer                              |          | 26.1%      | Easy       |           |
|        | 8. String to Integer (atoi)                     |          | 16.0%      | Medium     |           |
|        | 9. Palindrome Number                            |          | 51.3%      | Easy       |           |
|        | 10. Regular Expression Matching                 |          | 27.9%      | Hard       |           |
|        | 11. Container With Most Water                   |          | 53.1%      | Medium     |           |
|        | 12. Integer to Roman                            |          | 58.1%      | Medium     |           |
|        | 13. Roman to Integer                            |          | 57.6%      | Easy       |           |
|        | 14. Longest Common Prefix                       |          | 37.7%      | Easy       |           |

Every question in LeetCode has many different attributes (Lists, Difficulty, Status, Tags, Title, Acceptance), so it is very easy for a user to find a suitable question to practice. My solution can organize the question database in a similar way and provide corresponding query interface for a better user experience. The 'Pick One' button on the top right is a very handy feature as well. User can simply click that button to start working on a quick random question. The idea of 'a list of question' is great. Users can organize a series of questions to practice and share with each other.

## Pricing

The screenshot displays the LeetCode pricing page. At the top, there are two subscription cards: a 'Monthly' card with an orange header and a 'Yearly' card with a dark blue header. The Monthly card shows a price of \$35/month, down from \$39/month, and includes a 'Subscribe' button. The Yearly card shows a price of \$159/yr, down from \$299/yr, and also includes a 'Subscribe' button. Below the cards, there is a section titled 'Video Solutions' with a 'NEW' tag, followed by a list of premium features: 'Access to Premium Content', 'Select Questions by Company', 'Autocomplete', 'Debugger', 'Lightning Judge', 'Sort Questions by Prevalence', 'Interview Simulations', and 'Unlimited Playgrounds'. Each feature is accompanied by a brief description of its benefits.

**Monthly** Subscription **\$35<sub>mo</sub>**  
Down from \$39/month. Our monthly plan grants access to all premium features, the best plan for short-term subscribers. (prices are marked in USD)

**Yearly** Subscription **\$159<sub>yr</sub>**  
Our most popular plan previously sold for \$299 and is now only \$159/month. This plan saves you over 60% in comparison to the monthly plan. (prices are marked in USD)

**Video Solutions** NEW  
Unlock elaborate premium video solutions like [this](#). Each video includes a detailed conceptual overview and code walkthrough that will efficiently guide you through the problem.

**Access to Premium Content**  
Gain exclusive access to our latest and ever-growing collection of premium content, such as questions, Explore cards, and premium solutions, where detailed explanations are written by our team of algorithm and data structure experts.

**Select Questions by Company**  
Target your studying more accurately towards your dream job. Find out which companies asked which questions, we have nearly 200 questions from Google alone.

**Autocomplete**  
Not interested in memorization? With premium access, you receive intelligent code completion inside the LeetCode code editor based on language and an analysis of your source code.

**Debugger**  
Tired of `System.out.println(val)`? Set breakpoints and debug your code interactively line by line right inside our code editor.

**Lightning Judge**  
Tired of waiting? Premium users get priority judging using an exclusive queue, resulting in a 3X shorter wait time, up to 10X during peak hours.

**Sort Questions by Prevalence**  
Find out which questions turn up most frequently in interviews so that you know where to focus your personal studying. Invaluable data collected from thousands of samples.

**Interview Simulations**  
Mock assessments provide you with a way to test your abilities in a timed setting, just like a coding challenge or on-site interview. You choose the company and we will select an appropriate question from our constantly growing database.

**Unlimited Playgrounds**  
Premium users can create an unlimited number of Playgrounds, up from 10! You also get the ability to organize your Playgrounds in folders.

The basic functions of LeetCode is free to use for all users and it charges a fee for premium subscriptions. The premium subscription provides a larger question database, better code editor, faster judger and more.

## Analysis

LeetCode is a fully web-based solution, which means it works on any devices. However, it also means you will not be able to use it without an stable Internet connection. I decide to make my solution a desktop application, since most students practice coding with a computer. It also save me a lot of cost from running and maintaining a server. LeetCode runs a large community for users to discuss questions with each other. I am not adding such function to my solution. Teachers and students can use existing platforms they have been familiar with, it is unnecessary for me to develop a new platform and for the users to migrate from mature solutions. LeetCode has an easy-to-use graphical interface, which is important so new users can get their hands on very easily.

LeetCode does not support custom questions or any functions for educators. It is mainly designed for self-learners. My solution is designed for school use, so

it must support functions like custom questions, custom assignments, statistics data visualisations. LeetCode charges a subscription fee for essential functions. My solution will be free and open-source so everyone can benefit from it.

## 1.6.2 Codeforces

Codeforces is a competitive coding platform, it is mainly used by people to held coding competition.

### Main question layout

**A. Fox And Snake**

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

Fox Ciel starts to learn programming. The first task is drawing a fox! However, that turns out to be too hard for a beginner, so she decides to draw a snake instead.

A snake is a pattern on a  $n$  by  $m$  table. Denote  $c$ -th cell of  $r$ -th row as  $(r, c)$ . The tail of the snake is located at  $(1, 1)$ , then it's body extends to  $(1, m)$ , then goes down 2 rows to  $(3, m)$ , then goes left to  $(3, 1)$  and so on.

Your task is to draw this snake for Fox Ciel: the empty cells should be represented as dot characters  $(\cdot)$  and the snake cells should be filled with number signs  $(*)$ .

Consider sample tests in order to understand the snake pattern.

**Input**  
The only line contains two integers:  $n$  and  $m$  ( $3 \leq n, m \leq 50$ ).  
 $n$  is an odd number.

**Output**  
Output  $n$  lines. Each line should contain a string consisting of  $m$  characters. Do not output spaces.

**Examples**

**input**

```
3 3
```

**output**

```
***
..#
***
```

**input**

```
3 4
```

**output**

```
****
...#
****
```

**input**

```
5 3
```

**output**

```
***
..#
***
#..
***
```

**input**

```
9 9
```

**output**

```
*****
.....#
*****
.....#
*****
.....#
*****
.....#
*****
```

**Codeforces Round #290 (Div. 2)**

Finished

Practice

★

**Virtual participation**

Virtual contest is a way to take part in past contest, as close as possible to participation on time. It is supported only ICPC mode for virtual contests. If you've seen these problems, a virtual contest is not for you - solve these problems in the archive. If you just want to solve some problem from a contest, a virtual contest is not for you - solve this problem in the archive. Never use someone else's code, read the tutorials or communicate with other person during a virtual contest.

[Start virtual contest](#)

**Practice**

You are registered for practice. You can solve problems unofficially. Results can be found in the contest status and in the bottom of standings.

**Clone Contest to Mashup**

You can clone this contest to a mashup.

[Clone Contest](#)

**Submit?**

Language: GNU G++17 7.3.0

Choose file: Choose File No file chosen

Be careful: there is 50 points penalty for submission which fails the pretests or resubmission (except failure on the first test, denial of judgement or similar verdicts). "Passed pretests" submission verdict doesn't guarantee that the solution is absolutely correct and it will pass system tests.

[Submit](#)

**Last submissions**

| Submission | Time              | Verdict                |
|------------|-------------------|------------------------|
| 66675479   | Dec/12/2019 14:07 | Accepted               |
| 66675379   | Dec/12/2019 14:05 | Wrong answer on test 1 |
| 66675290   | Dec/12/2019 14:03 | Wrong answer on test 3 |

**Problem tags**

Implementation 800

No tag edit access

The questions and the examples take up nearly all the spaces on the question page. There is no online editor or online runtime environment provided. Users are expected to write and test their code in their own IDEs and only submit the solution for judging. Custom IDEs may be more powerful than a builtin one. My solution will provide an editor, it is much more convenient to use. Even if a user decide to use his own environment, he can paste his code into the editor for submission. It sets time and memory limit for users' submissions, if a piece of code takes to long to run, or takes up too much memory while running, it will be terminated and marked wrong.



## Submission

PROBLEMS SUBMIT CODE MY SUBMISSIONS STATUS HACKS ROOM STANDINGS CUSTOM INVOCATION

General

| #        | Author               | Problem   | Lang         | Verdict  | Time  | Memory | Sent                   | Judged                 |   |         |
|----------|----------------------|-----------|--------------|----------|-------|--------|------------------------|------------------------|---|---------|
| 66675479 | Practice:<br>PCloud_ | 510A - 15 | GNU<br>C++17 | Accepted | 31 ms | 8 KB   | 2019-12-12<br>14:07:49 | 2019-12-12<br>14:07:49 | ★ | Compare |

→ Source

Copy

```
#include<iostream>
using namespace std;
int main()
{
    int m = 0;
    int n = 0;
    cin >> n >> m;
    for(int i = 0; i < n; i++)
    {
        if(i%2==0)
        {
            for(int j = 0; j < m; j++)
            {
                cout<<"#";
            }
            cout<<endl;
        }
        else if(i%4==1)
        {
            for(int j = 0; j < m-1; j++)
            {
                cout<<".";
            }
            cout<<"#<endl;
        }
        else if(i%4==3)
        {
            cout<<"#";
            for(int j = 0; j < m-1; j++)
            {
                cout<<".";
            }
            cout<<endl;
        }
    }
}
```

When the user submits the code, the code enters a queue waiting for judging, then the user can look up their result. Users can check their source code, performance stats, and more important, when they have not passed all test cases, they are able to see what they have got wrong. The judgement protocol provide detailed information about each test case, so users can debug easily.

### —Judgement Protocol

Test: #1, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

aaabb

Output

6

Answer

6

Checker Log

ok 1 number(s): "6"

Test: #2, time: 15 ms., memory: 4 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

usac0

Output

1

Answer

1

Checker Log

ok 1 number(s): "1"

Test: #3, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 0, verdict: OK

Input

l0l

Output

0

Answer

2

Checker Log

ok 1 number(s): "2"

Test: #4, time: 15 ms., memory: 0 KB, exit code: 0, checker exit code: 1, verdict: WRONG\_ANSWER

Input

qdpinbncrfwxpdbfgozvvcemjructoadewegtvtbvbfmrpgyeaxgddrwnlqnygmhmhrhaizpyxvgaFlrsvzhzhrouvprixkfza

Output

-763363328

Answer

37

Checker Log

wrong answer 1st numbers differ - expected: '37', found: '-763363328'

## Analysis

Codeforces is optimized for coding competition, so it has a light weight and complex interface for better performance. It is completely free to use. Users can create their own questions but it is very complicate to do so. My solution needs to enable users without experiences to create questions easily. There is no function for education - there is no way for a teacher to 'create a class' and monitor his students. Codeforces is an online platform, so it also work across all devices and require an Internet connection. Users have to use their own IDEs to write and debug their code. Codeforces sends all submissions to a central 'judging queue' for marking. My solution will mark all submissions in local, which makes judging a lot faster and save me from running a server. By limiting time and space allowance, Codeforces effectly prevents malicious code from running.

## 1.7 Features

(TODO): Propose a solution to the problem by describing each element of the product in detail. You can have mock ups of the graphics / screen designs from a drawing application at this stage.

## 1.8 Limitations

The software will be written in C# instead of web-based which means extra software needs to be downloaded by the user. I plan to use .Net 5 runtime and WinUI 3 library for my solution, so only the Windows 10 1809 or newer Windows operating systems will be supported. This should not cause much compatibility issues since most school computers are running the required version of operating system. Downloading an extra software is inconvenient and may violate the IT security policy of some schools.

The judger can only accept code submission in limited programming languages and the user may requie to configure their own runtime environment. Creating a compiler for 'OCR Pseudocode Programming Language' is too complex for this project. I will attempt to allow the user to add their own programming language and write a documentation for them to make the process easier.

Unlike LeetCode, there is no Discussion pages for users to discuss questions because it is a desktop program instead of a web one. But this is not a big problem, students and teachers should use an existing product such as Microsoft Teams which has very good support in sharing code snippets. It is unnecessary to rebuild the wheel.

Distribute the questions and assignments is still something inconvenient. Currently, distributing questions and assignments requires the teacher to first ex-

port the questions and assignments, then send them to the students through email or file-sharing platforms. When the students finish working, they need to send their result back through email or other apps. I have attempted to integrate the file-sharing function with the existing platform - the Microsoft Teams Assignment function. ~~But very unfortunately, the Graph API required for this operation is still in beta version, which means it can only be tested in the development environment and cannot be used in production. So for now, the users still have to use this inconvenient way to share questions and submissions. But in the future, the integration with some existing platforms may improve the experience.~~ (Update: the Microsoft Teams API is out of beta, now it is possible to integrate with it)

There are no good ways to maintain and distribute a large question database. Computer Science teachers are required to maintain a database for their own students. But this is a difficult work. Creating good test cases is much time consuming than writing a mark scheme, it is very likely for a wrong solution to pass the judging if the test cases are not good enough. It relies on the teacher who creates the questions to consider everything clearly to minimize its impact.

The judger can only simply compare the students' output with the expected output, if there is a format error such as trailing space and extra newline in their output, which will not be considered as a mistake in a real exam, will be marked as a wrong answer by the judger. So students may need to spend extra time debugging their output format. It cannot judge "partially correct" answer as well. It does not care which line did the student get correct or wrong, if the final output doesn't match, the submission will be marked wrong.

## 1.9 Hardware and software requirements

| Hardware and software requirements  | Justification  |
|---|--|
| Standard mouse, keyboard and monitor.   | Standard I/O devices are required for the user to interact with the software. Users need a mouse to navigate around different menus and pages, they need to use a keyboard to input their code solutions and use a monitor to get the output from the software.  |
| Operating system: Windows 10 (1809 or later), Windows 11.                                       | The software is designed with the WinUI 3 library and .Net 5 runtime, which require such operating system to run.  |
| x86 64-bit CPU (Intel / AMD architecture) with 2 or more cores and 1 GHz or higher clock speed. | A modern CPU is required for the software. 1 core will be used to run the main program and at least 1 spare core is required for the judger to judge the submitted code. A clock speed higher than 1 GHz is required to ensure the software is running smoothly. |
| 1GB free memory or more.  | Around 512MB RAM is required to run the software, and another 512MB RAM is required for the judger to judge the submissions.   |
| 256MB free disk space or more.  | 256MB free disk space is required to store and run the program itself, user may need extra disk space to store extra cache data and the database.  |
| A modern dedicate or integrate graphics card.   | The software has very little graphical demand, if the user's graphics card can run their operation system, it should be able to handle software as well.   |

## 1.10 Success criteria

| Criteria   | Justification  |
|--|--|
| Provide a menu and a homepage filled with buttons link to different functions of the solution.                         | To make the program easy to use and allow the user finding the function they want to use quickly |
| Provide an interface for the problem database.   | Users can search for questions easily through the interface                                      |
| Provide an interface for the user to add or export new questions.  | Users can easily add new questions, edit existing questions or share them with others.           |
| Provide a training interface where the question and the code editor is displayed.                                      | This allows the user to practice.  |
| Provide a setting menu for the user to adjust their preferred programming languages, colour theme, and other settings. | Users can have a customized experience with different setting options.                           |

## Chapter 2

# Design

### 2.1 Decomposition

(TODO): Add design chapter

## Chapter 3

# Development

(TODO): Add development chapter

## Chapter 4

# Evaluation

(TODO): Add evaluation chapter