



B. Tech Project Presentation

By

Abhinav Hinger 160101004

Apurva N. Saraogi 160101013

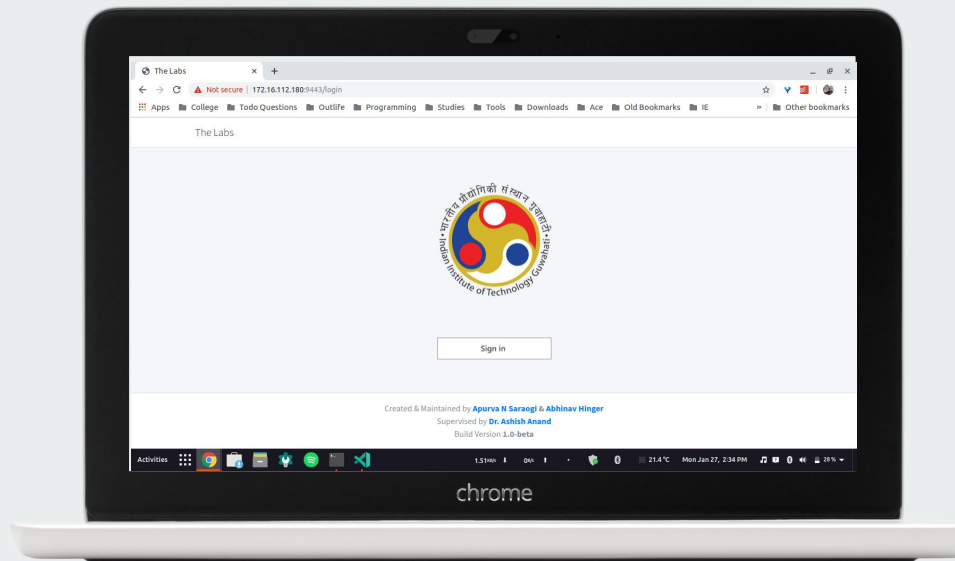
Under the guidance of

Prof. Ashish Anand

Department of Computer Science and Engineering, IIT Guwahati

The Labs

An unified coding and automatic
evaluation platform



Outline

Previous Methods of evaluation

Not just another 'portal'

The Labs use-case

What's in it for me?

The Platform

How does it work?

More importantly how do I use it?

Results from CS110

How did it work ?

Current Scenario

What are we up to now?



Previous Scenario

- Instructor creates a PDF file containing the questions. This is uploaded to a shared directory by TA from which students copy their assignment.
- Students solve this on their machine in an **offline environment**.
- At the end of the session, TAs manually evaluate each students' code



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you.

20% complete



For more information about this issue and possible fixes, visit <https://www.windows.com/stopcode>

If you call a support person, give them this info:

Stop code: CRITICAL_PROCESS_DIED

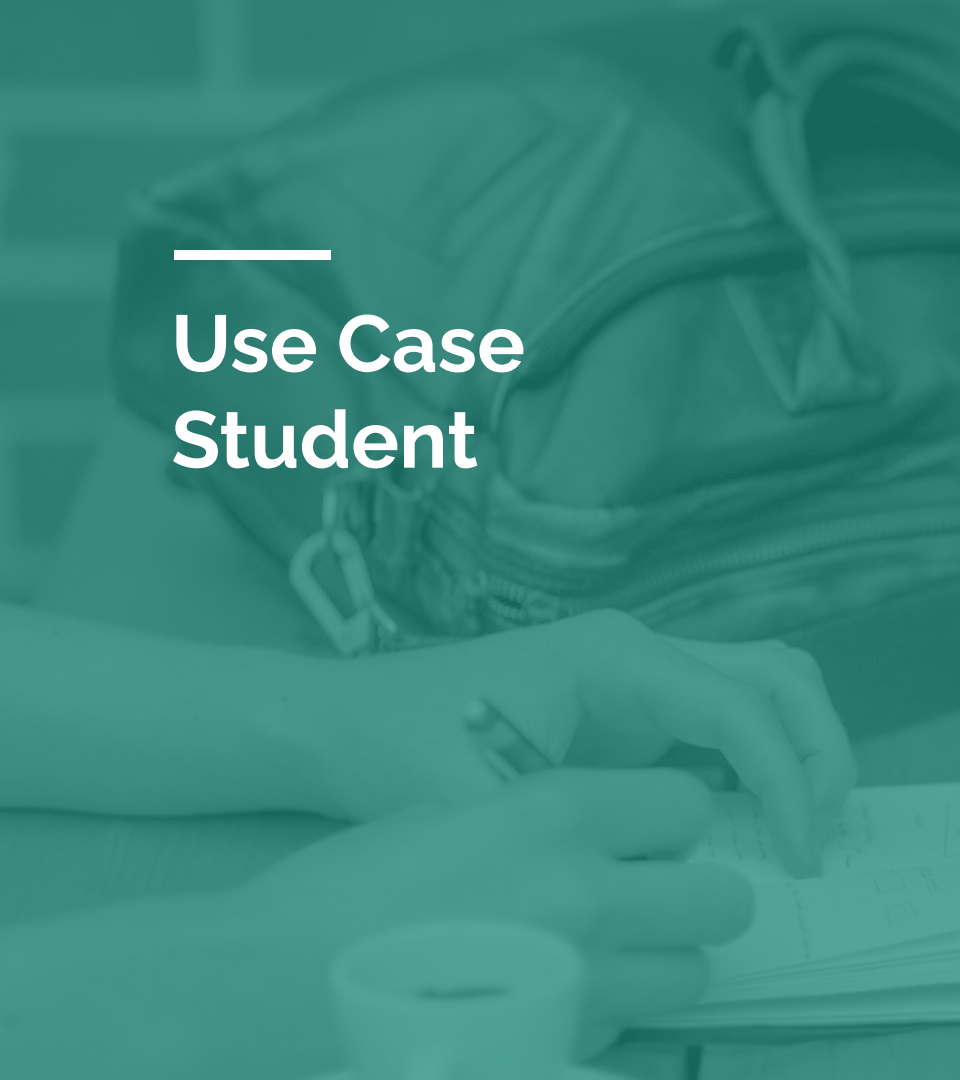


Problems

- System Crash
- Internet Issues
- No way to monitor progress of a student.
- No way to monitor unfair means
- Bias in evaluation
- Extra work for instructors

The Labs

An unified coding and automatic evaluation platform



Use Case Student

A platform for

- Writing and testing programs in a restricted environment as part of CS110.
- Preliminary knowledge of OS not required.
- Unbiased and Fair evaluations.
- Intuitive and Easy to understand



Use Case

Teaching Assistants

- No manual checking of each student's code
- More time for solving students' doubts and problems.
- Better tracking of students.
- Need to make questions once.



Use Case

Professors

- Systematic lab and question creation.
- Easy cloning of labs.
- Since Questions are added to the database as a separate entity, Professors can use them multiple times.
- Better control over students activities and progress.
- COVID19 --> Everything is online so no problems there.

How to use it?

The Labs

Not secure | 172.16.112.180:9443/login

Apps College Todo Questions Outlife Programming Studies Tools Downloads Ace Old Bookmarks IE Other bookmarks

The Labs

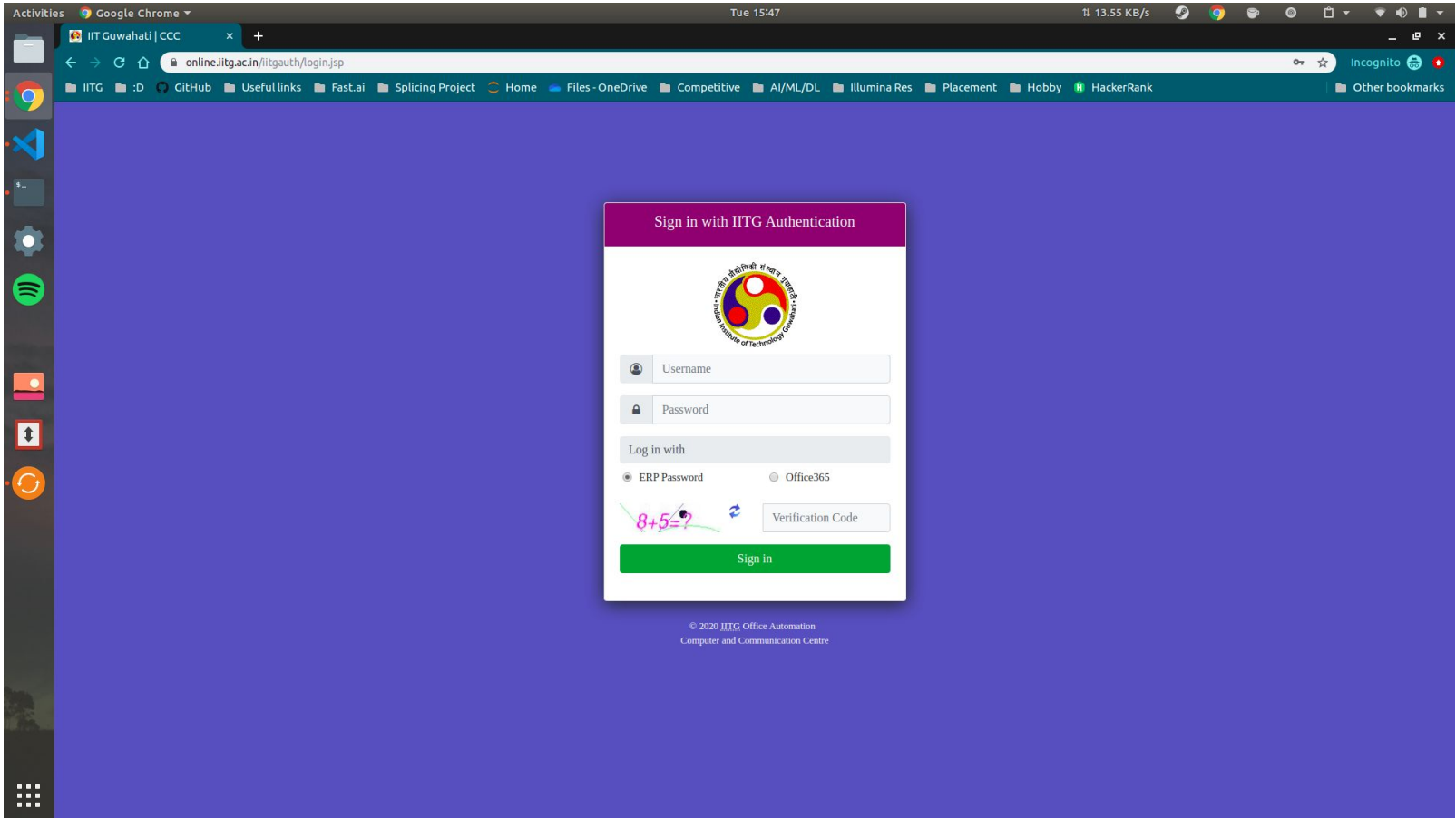


Sign in

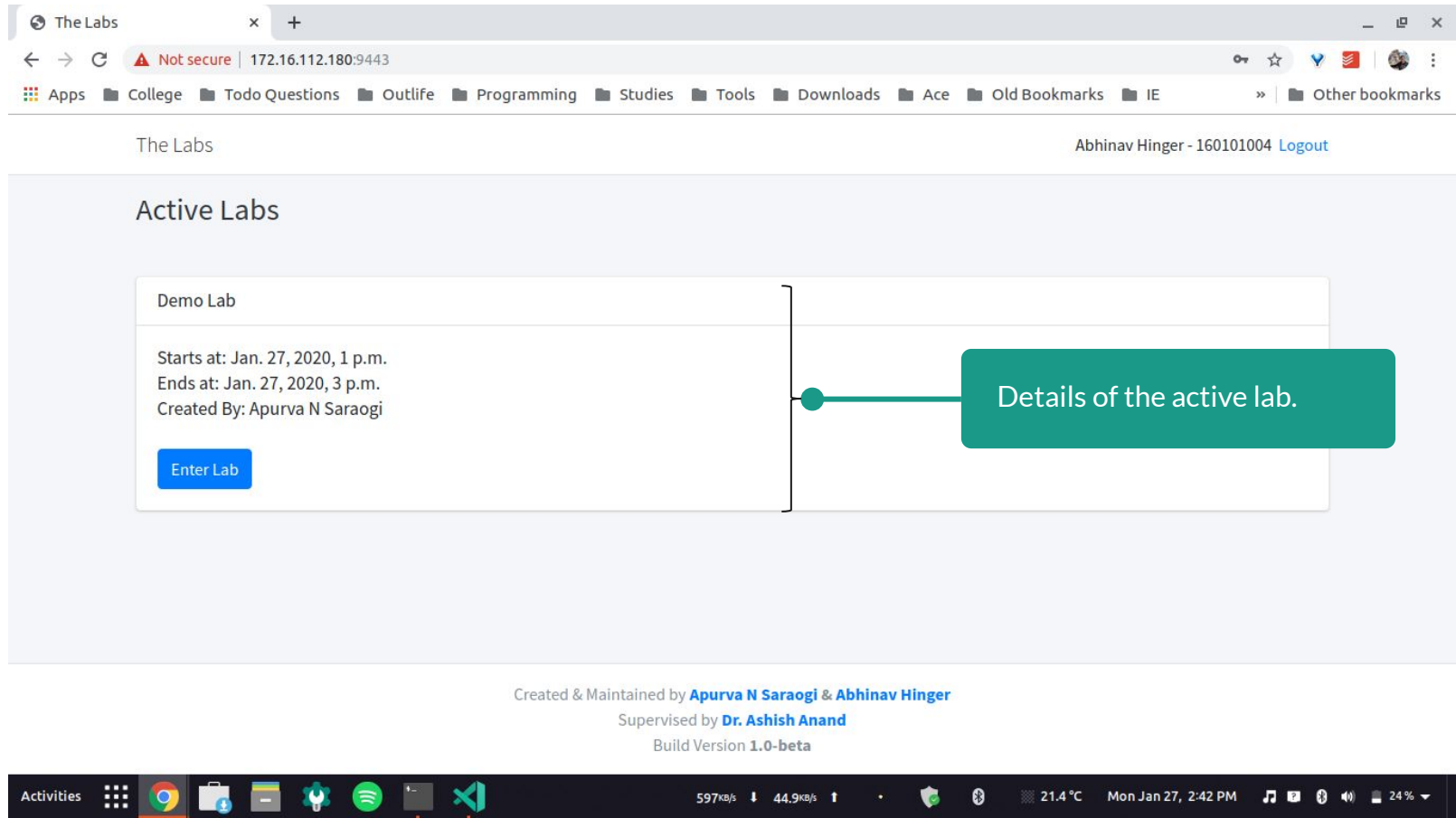
Created & Maintained by [Apurva N Saraogi](#) & [Abhinav Hinger](#)
Supervised by [Dr. Ashish Anand](#)
Build Version 1.0-beta

Activities 1.51KB/s 0B/s 21.4°C Mon Jan 27, 2:34 PM 28%

Authentication



Authentication



The Labs x +

Not secure | 172.16.112.180:9443/enter-lab/1

Apps College Todo Questions Outlife Programming Studies Tools Downloads Ace Old Bookmarks IE » Other bookmarks

The Labs Abhinav Hinger - 160101004 [Logout](#)

Enter the Test Key

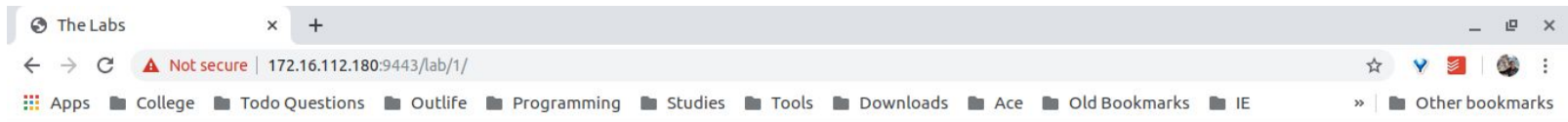
Password: Submit

Enter the lab password

Created & Maintained by [Apurva N Saraogi](#) & [Abhinav Hinger](#)
Supervised by [Dr. Ashish Anand](#)
Build Version 1.0-beta

Activities 17.6KB/s 28.7KB/s 21.4 °C Mon Jan 27, 2:43 PM 23 %

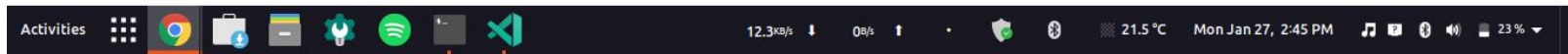
Entering the Lab



You are required to be in fullscreen mode to proceed.

Note: All tab changes and window changes will be monitored.

Go Fullscreen





Restricted Environment

Following tasks are restricted and are tracked for every student

- Switching to other Tabs. Time spent outside the focus is tracked.
- Switching to applications other than the browser
- Copy-Pasting code from other tabs
- Switching PC to another one than that assigned by the TA.



ALL

1

2

Questions

Important Instructions.

1. Greetings

50 Points

[Solve Question](#)

2. Addition

100 Points

[Solve Question](#)

[Finish Lab](#)

 Demo Lab

Time Left: 0:10:52 Abhinav Hinger - 160101004 [Logout](#)

Questions

?

#

ALL

1

2

1. Greeting

50 Points

Solve Question

2. Addition

100 Points

Solve Question

Finish Lab

Tutorial



ALL

1

2

Questions

1. Greetings

50 Points

[Solve Question](#)

List of all the questions

2. Addition

100 Points

[Solve Question](#)

[Finish Lab](#)



?

#

ALL

1

2

Questions

1. Greetings

50 Points

Solve Question

2. Addition

100 Points

Solve Question

Questions

Finish Lab



Demo Lab

Time Left

Time Left: 0:10:52 Abhinav Hinger - 160101004 [Logout](#)

Questions

1. Greetings

50 Points

[Solve Question](#)

2. Addition

100 Points

[Solve Question](#)

[Finish Lab](#)

?

#

ALL

1

2

 Demo Lab

Name & Roll Number

Time Left: 0:10: Abhinav Hinger - 160101004 [Logout](#)

Questions

?

#

ALL

1

2

1. Greetings

50 Points

Solve Question

2. Addition

100 Points

Solve Question

Finish Lab

 Demo Lab

Logout

Questions

?

#

ALL

1

2

1. Greetings

50 Points

Solve Question

2. Addition

100 Points

Solve Question

Finish Lab



ALL

1

2

Questions

1. Greetings

Name of the Question

50 Points

Solve Question

2. Addition

100 Points

Solve Question

Finish Lab



ALL

1

2

Questions

1. Greetings

Points for each question

50 Points

[Solve Question](#)

2. Addition

100 Points

[Solve Question](#)

[Finish Lab](#)



Demo Lab

Time Left: 0:10:52 Abhinav Hinger - 160101004 [Logout](#)

?

#

ALL

1

2

Questions

1. Greetings

Solve the Question

Solve Question

2. Addition

100 Points

Solve Question

Finish Lab



Demo Lab

Time Left: 0:10:52 Abhinav Hinger - 160101004 [Logout](#)

?

#

ALL

1

2

Questions

1. Greetings

50 Points

[Solve Question](#)

2. Addition

100 Points

[Solve Question](#)

[Finish Lab](#)

Finish Lab

AL2-W06-Clones [Go Home](#)

Time Left: 23:54:31 Apurva N Saraogi - 160101013 [Logout](#)

1 of 13



?

#

ALL

1

2

3

4

+

Tutorial

LAB #06

CS110

Department of Computer Science and Engineering
Indian Institute of Technology Guwahati
Semester II, Academic Year 2019 - 2020

Functions

A function is a group of statements that together perform a task. Every C program has at least one function, which is `main()`, and all the most trivial programs can define additional functions.

USES OF C FUNCTIONS:

- C functions are used to avoid rewriting same logic/code again and again in a program.
- There is no limit in calling C functions to make use of same functionality wherever required.
- We can call functions any number of times in a program and from any place in a program.
- A large C program can easily be tracked when it is divided into functions.
- The core concept of C functions are, re-usability, dividing a big task into small pieces to achieve the functionality and to improve understandability of very large C programs.

You can divide up your code into separate functions. How you divide up your code among different functions is



Question View

Clicking on a question opens the question view

It comes with question content followed by Code editor.

Students can write code and test against given test cases or their own custom input using Submit button.



ALL

1

2

Addition

Points: 100
Language: C++
Memory Limit: 64 MB
Time Limit: 2s

Basic Information

Abhinav is busy building the final release of this platform. His friend Apurva asks him to solve a quick question. Apurva gives two integers a & b Abhinav has to tell back the sum of the numbers. Can you help Abhinav in telling the answer?

Constraints

- $a, b < 10^5$

Input format

- First line contains space two separated numbers a & b

Output format

- One Integer giving the result

Example

Input

23 60

Output

83



ALL

1

2

Addition

Points: 100

Language: C++

Memory Limit: 64 MB

Time Limit: 2s

Abhinav is busy building the final release of this platform. His friend Apurva asks him to solve a quick question. Apurva gives two integers a & b Abhinav has to tell back the sum of the numbers. Can you help Abhinav in telling the answer?

Constraints

- $a, b < 10^5$

Input format

- First line contains space two separated numbers a & b

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

The Question

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

1

Code Editor

☐ Custom Input

Save

Submit

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

1

Supports key bindings of
Sublime Text Editor

☐ Custom Input

Save

Submit

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

1

☐ Custom Input

Submit

Save Submit

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

1

☐ Custom Input

Save

Save

Submit

Output format

- One Integer giving the result

Example

Input

23 60

Output

83

1

☐ Custom Input

Custom Input

Save

Submit

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y;
5     scanf("%d %d",&x,&y);
6     printf("%d\n",x);
7     printf("%d\n",x+y);
8     return 0;
9 }
```

☒ Custom Input

Saved successfully at 10:51:41 28-01-2020

Save

Submit

23 60

Custom Input

Compiler Output

23
83

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y;
5     scanf("%d %d",&x,&y);
6     printf("%d\n",x);
7     printf("%d\n",x+y);
8     return 0;
9 }
```

☒ Custom Input

Saved successfully at 10:51:41 28-01-2020

Save

Submit

23 60



Compiler Output

23
83

The output when the code is run.

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y;
5     scanf("%d %d",&x,&y);
6     printf("%d",x+y);
7     return 0;
8 }
```

☐ Custom Input

Saved successfully at 09:35:30 27-01-2020

Save

Submit

Compiler Output

```
main.cpp: In function 'int main()':
main.cpp:6:18: error: expected ';' before 'return'
   6 | printf("%d",x+y)
     |                  ^
     |                  ;
   7 | return 0;
     | ~~~~~
```

Compiler Output

Output

83

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y;
5     scanf("%d %d",&x,&y);
6     printf("%d",x+y);
7     return 0;
8 }
```

☐ Custom Input

Saved successfully at 09:34:53 27-01-2020

Save

Submit

Running testcases

✓ Testcase 1

✓ Testcase 4

✓ Testcase 2

✓ Testcase 5

Output of each test case.

Output format

- One Integer giving the result

Example

Input

23 60

Output

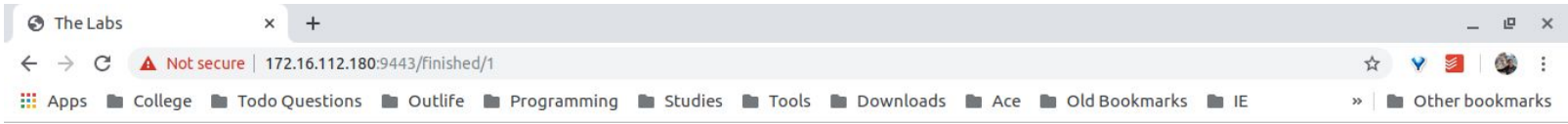
83

```
1 #include <stdio.h>
2
3 int main(){
4     int x,y;
5     scanf("%d %d",&x,&y);
6     printf("%d",x+y);
7     return 0;
8 }
```

☐ Custom Input

Saved successfully at 10:48:02 28-01-2020

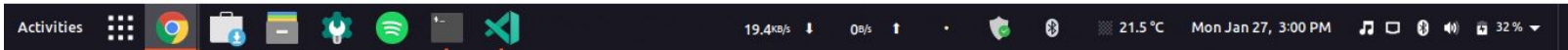
Saved Successfully.



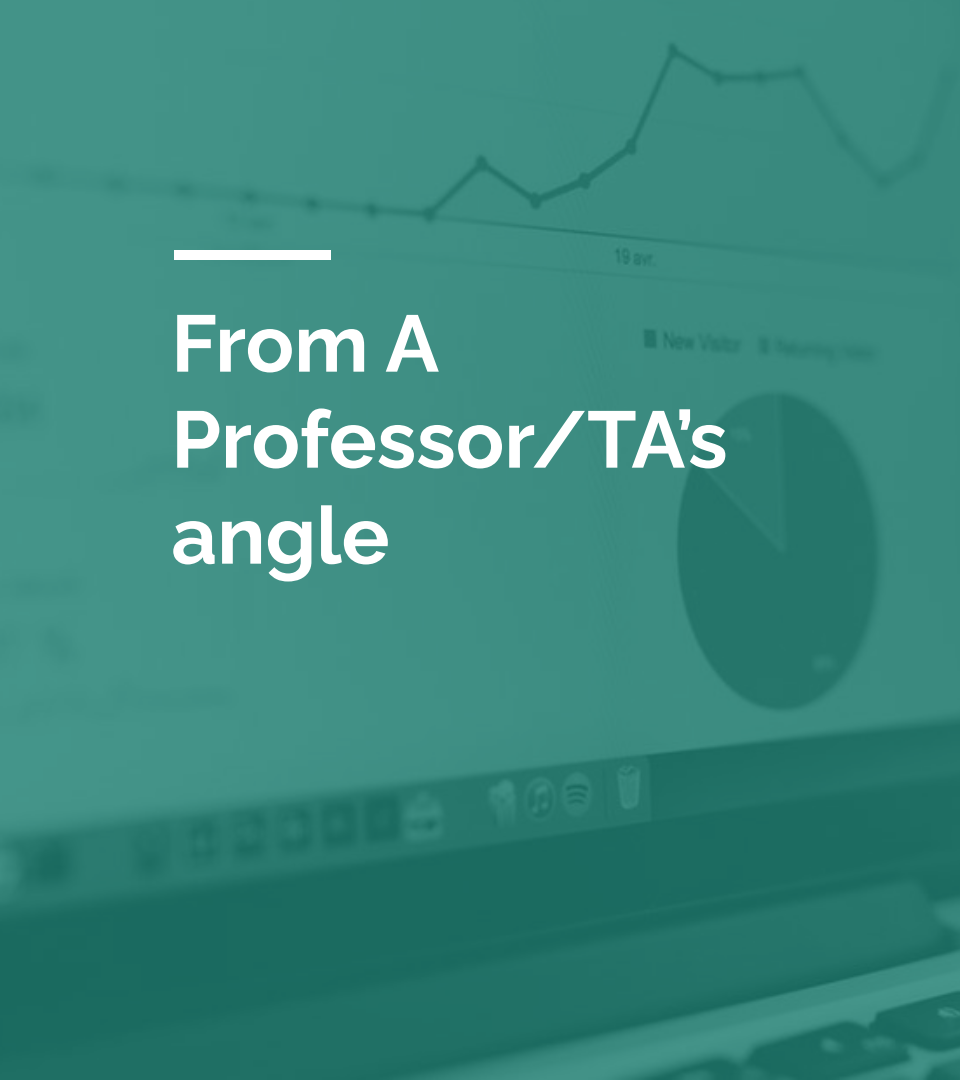
Lab session over
Demo Lab

Abhinav Hinger - (160101004)
[Logout](#)

Once Finished Show this to TA



Finished



From A Professor/TA's angle

In the previous sections we have seen features for students now in following sections we will see features for Professors/TAs

Active Labs

[Create Programming Lab](#)[Create MCQ Lab](#)[Add Accounts](#)[Users](#)[Create Various type of labs](#)

AL2-W06

Starts at: March 17, 2020, 2 p.m.

Ends at: March 17, 2020, 5 p.m.

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

ML2-W06

Starts at: March 17, 2020, 9 a.m.

Ends at: March 17, 2020, noon

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

AL1-W06

Starts at: March 16, 2020, 2 p.m.

Ends at: March 16, 2020, 5 p.m.

Active Labs

[Create Programming Lab](#)[Create MCQ Lab](#)[Add Accounts](#)[Users](#)[Question Bank](#)

AL2-W06

Starts at: March 17, 2020, 2 p.m.

Ends at: March 17, 2020, 5 p.m.

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)

See lab details & edit it if needed

[Clone Lab](#)

ML2-W06

Starts at: March 17, 2020, 9 a.m.

Ends at: March 17, 2020, noon

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

AL1-W06

Starts at: March 16, 2020, 2 p.m.

Ends at: March 16, 2020, 5 p.m.

Active Labs

[Create Programming Lab](#)[Create MCQ Lab](#)[Add Accounts](#)[Users](#)[Question Bank](#)[Add/Remove Users and Modify Permissions](#)

AL2-W06

Starts at: March 17, 2020, 2 p.m.

Ends at: March 17, 2020, 5 p.m.

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

ML2-W06

Starts at: March 17, 2020, 9 a.m.

Ends at: March 17, 2020, noon

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

AL1-W06

Starts at: March 16, 2020, 2 p.m.

Ends at: March 16, 2020, 5 p.m.

Active Labs

[Create Programming Lab](#)[Create MCQ Lab](#)[Add Accounts](#)[Users](#)[Question Bank](#)

AL2-W06

Starts at: March 17, 2020, 2 p.m.

Ends at: March 17, 2020, 5 p.m.

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)

Allows to quickly duplicate a lab.

[Clone Lab](#)

ML2-W06

Starts at: March 17, 2020, 9 a.m.

Ends at: March 17, 2020, noon

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

AL1-W06

Starts at: March 16, 2020, 2 p.m.

Ends at: March 16, 2020, 5 p.m.

Active Labs

[Create Programming Lab](#)[Create MCQ Lab](#)[Add Accounts](#)[Users](#)[Question Bank](#)

Question Bank for MCQ Lab

AL2-W06

Starts at: March 17, 2020, 2 p.m.

Ends at: March 17, 2020, 5 p.m.

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

ML2-W06

Starts at: March 17, 2020, 9 a.m.

Ends at: March 17, 2020, noon

Created By: Gouri Sankar

[Enter Lab](#)[Lab Details](#)[Clone Lab](#)

AL1-W06

Starts at: March 16, 2020, 2 p.m.

Ends at: March 16, 2020, 5 p.m.



Adding Question & Test Case

**After entering a lab,
admin/staff will be displayed a
Plus button (+) on the sidebar**

Clicking on this will take you to
question creation form and
subsequently test case creation form



ALL

1

2

3

4

+

Edit Question

Name:

Stairs

Content:

Testcase Weightage

- Test Case 1: 25%
- Test Case 2: 25%
- Test Case 3: 25%
- Test Case 4: 25%

Problem Description

You are climbing a staircase and it takes **N** steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?

Points:

10

Memory limit:

64

Time limit:

1

Languages supported:

1

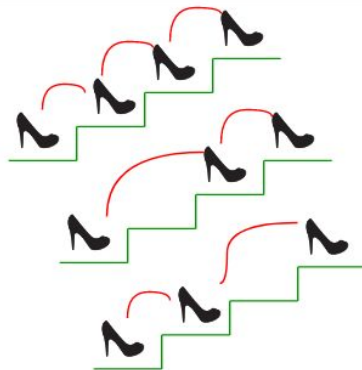
Base code:

Testcase Weightage

- Test Case 1: 25%
- Test Case 2: 25%
- Test Case 3: 25%
- Test Case 4: 25%

Problem Description

You are climbing a staircase and it takes **N** steps to reach to the top. Each time you can either climb 1 or 2 steps. In how many distinct ways can you climb to the top?



In the above figure value of **N** is 3. First step is not counted, as a person starts from this step. The number of ways to reach top is 3.

Testcase #1

Testcase #2

Points:

25

Stdin:

2



Stdout:

2

Testcase #3

Testcase #4

Submit

Upload Image



Viewing Lab Statistics

Staff/Admin can view the details of a lab in real time by clicking on Lab details button on a Lab card on the landing page.

Sessions

[CSV](#) [Copy](#) [Excel](#) [PDF](#)

Search:

Name ↕	Roll Number ↕	Started at ↕	Number of Times Switched ↕	Total Time Outside ↕	Q1 ↕	Q2 ↕	Q3 ↕	Q4 ↕	Q5 ↕	Number of Sessions ↕	Details ↕	Restart Session ↕
ABHISAR GAUTAM	190104001	Feb. 19, 2020, 9:03 a.m.	4	174.299	0	0	0	16.0	0	2	Here	Restart
ABHISHEK	190104002	Feb. 19, 2020, 9:04 a.m.	1	0.516	0	8.0	9.0	0.0	0	1	Here	Restart
ABHISHEK VERMA	190104004	Feb. 19, 2020, 9:11 a.m.	3	58.943	4.0	0	0	0	0	1	Here	Restart
ADARI YASHWANTH	190104005	Feb. 19, 2020, 9:09 a.m.	1	0.102	0	0	0	12.8	16.0	1	Here	Restart
ADITYA JOSHI	190104006	Feb. 19, 2020, 9:15 a.m.	2	0.373	3.0	0	0.0	0	0	1	Here	
ADITYA VYAS	190104007	Feb. 19, 2020, 9:10 a.m.	0	0.0	0.0	4.0	12.0	0	0	1	Here	Restart

[Edit Lab](#)
[Submissions](#)
[Download Code](#)

User details

Name	Abhisar Gautam
Roll Number	190104001
Started at	Feb. 19, 2020, 9:03 a.m.
Finished at	Feb. 19, 2020, 10:28 a.m.
Number of Times Switched	4
Total Time Outside	174.299

Sessions

172.16.5.81	Feb. 19, 2020, 3:33 a.m.
172.16.5.81	Feb. 19, 2020, 4:58 a.m.

Q4

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int n,k,b=0;
    int m[n],a[n];
    scanf("%d %d",&n,&k);
    for(int i=1;i<=n;i++)
    {scanf("%d\t",&m[i]);}

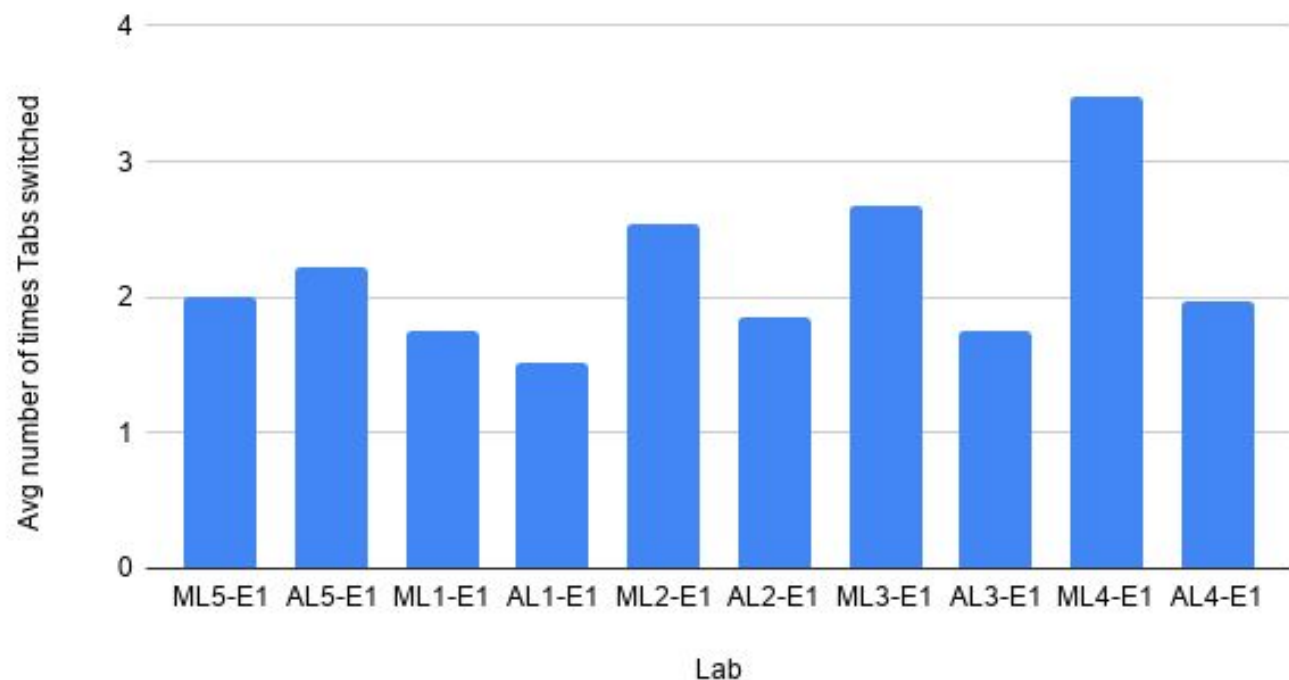
    for(int p=1;p<=n;p++)
```

Results from CS110

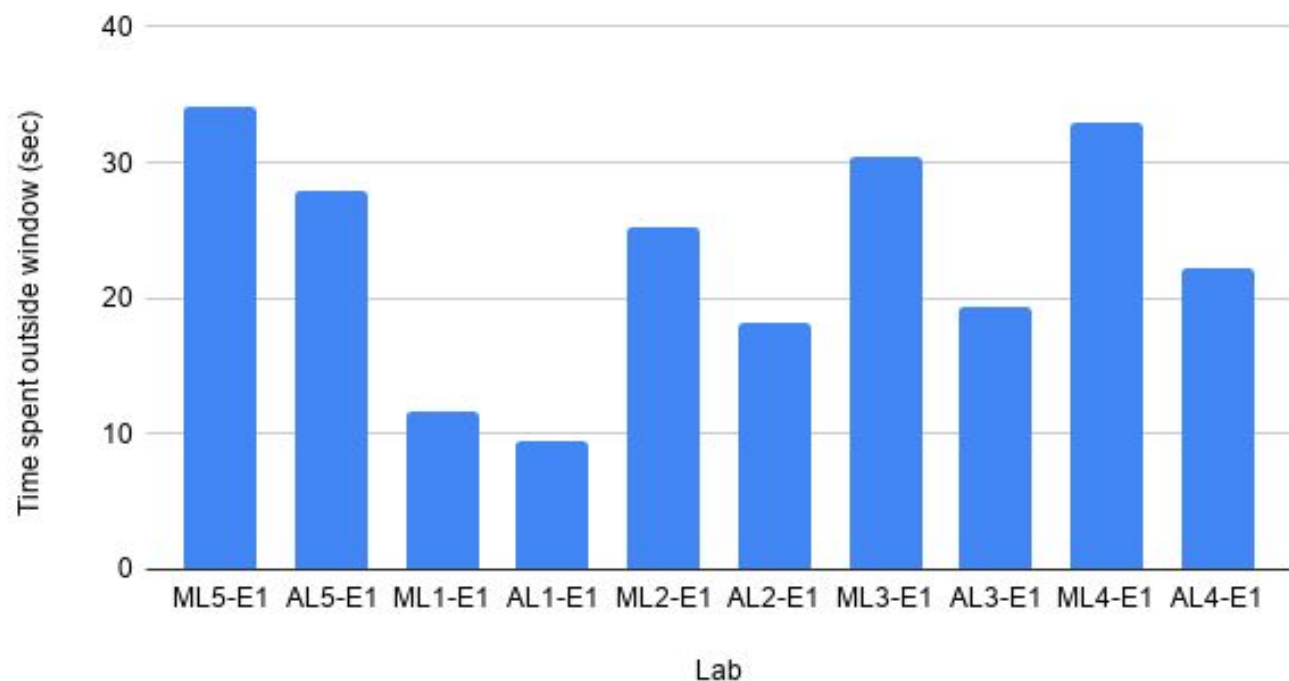
Overall

- Total users on the platform : 892
- Total number of labs created : 43
- Total number of questions created : 228
- Total submissions evaluated: 219934

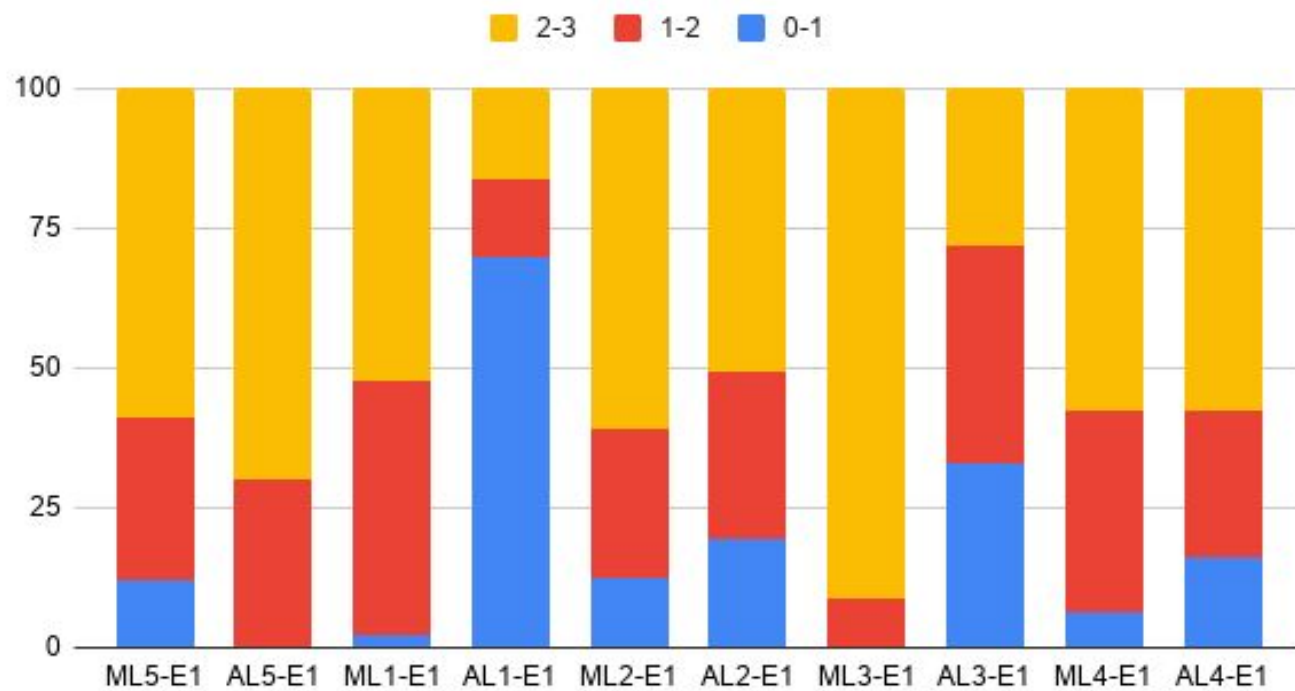
Avg number of times Tabs switched vs. Lab



Time spent outside window (sec) vs. Lab



Percentage of students finishing per hour



Compilation Error

7.9%

Wrong Answer

12.5%

Partially Correct

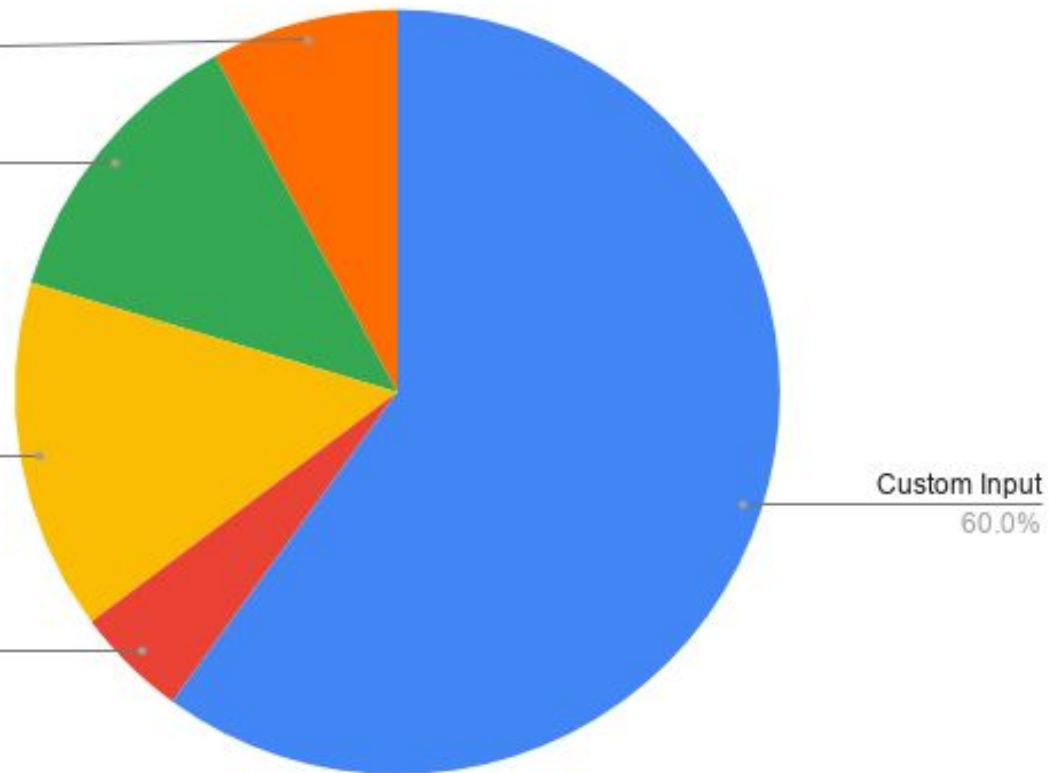
14.9%

Accepted

4.7%

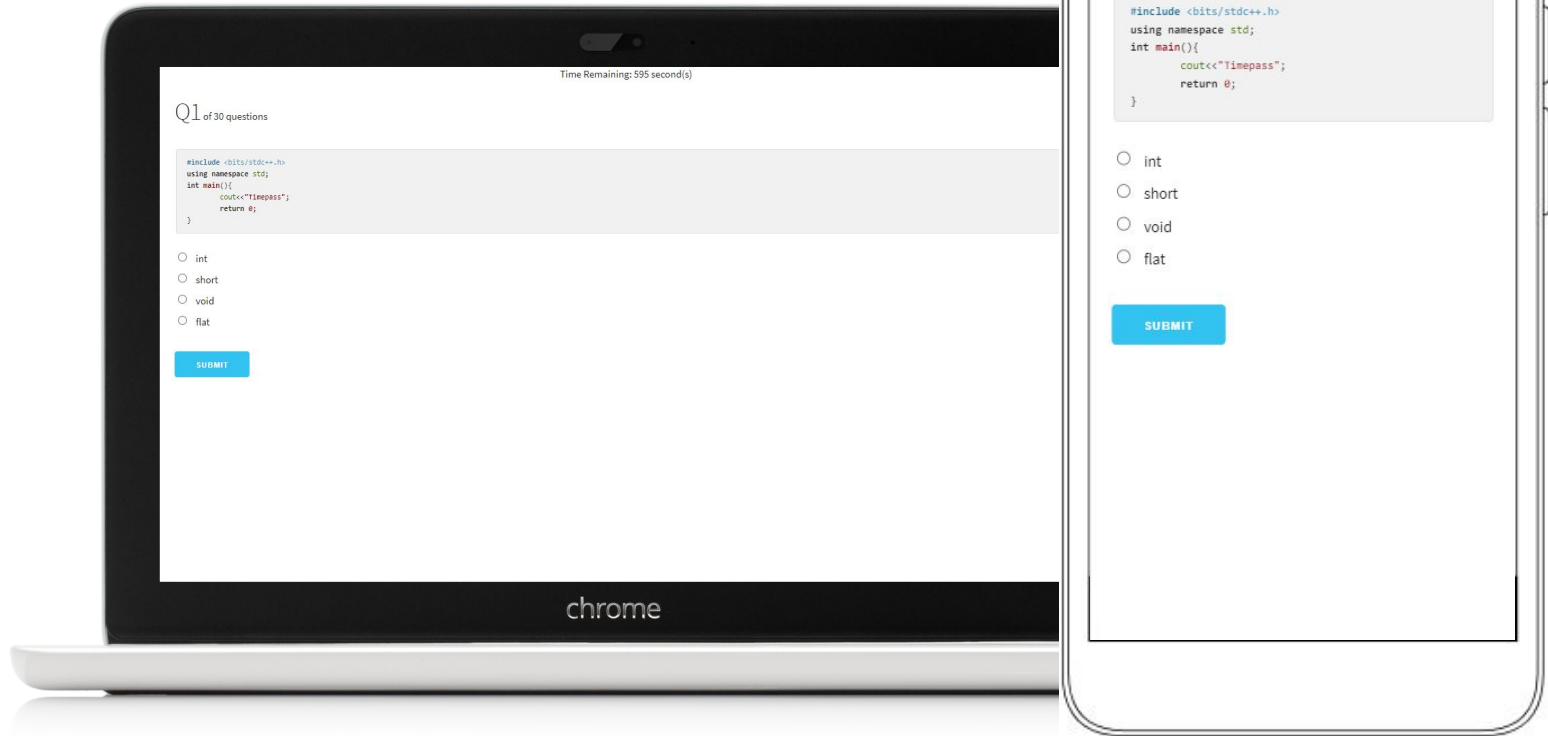
Custom Input

60.0%



Current Features under Development

Light weight
Low bandwidth Requirements
Works on both Mobile and Desktop



Structure

Pool

A set of questions, can be made on a given topic and difficulty.

Question

A question is an MCQ question with time allotted to the question.

Question Variant

A variant of a question is the same question but different data.

A pool will have let's say n questions and each question will have n_i variants.

A professor can allot m questions to be given to each student from this pool that will be selected randomly for each question. Also the options for MCQ will be shuffled for each question



MCQ Lab

- We present a unique way to take MCQ test. For each question an allotted time will be given. A student can not randomly go to a question. Due to all this the probability of cheating reduces as there is pressure on every one.
- Increasingly needed to take evaluations online which are both student friendly and instructor friendly.
- Professors can add questions to various pools into the question bank accessible from the landing page
- We will create a different set for each student by randomly picking question from each pool.
- Option of adding variants of a question so that all students get similar difficulty question



Plagiarism detection post-lab

- Plagiarism detection over all the code written in a particular lab. It will be beneficial for labs that go into more depth resulting in unique style of each student.

Eg. Database labs and data structure labs

- Easy to run plagiarism check accessible from lab details page itself.
- Provide visualisations for similarity report so that it is easier identify similar source code

File 1	File 2	Lines Matched
submission/190103088.4.c (96%)	submission/190104045.4.c (96%)	37
submission/190103101.4.c (68%)	submission/190103104.4.c (38%)	19
submission/190103090.4.c (53%)	submission/190104058.4.c (48%)	24
submission/190104019.1.c (98%)	submission/190104033.1.c (98%)	20
submission/190104013.1.c (98%)	submission/190104033.1.c (98%)	22
submission/190104013.1.c (98%)	submission/190104019.1.c (98%)	22
submission/190103100.4.c (87%)	submission/190104035.4.c (24%)	18
submission/190103091.1.c (98%)	submission/190103098.1.c (98%)	24
submission/160106021.4.c (72%)	submission/190104001.4.c (70%)	17
submission/190104035.4.c (21%)	submission/190104058.4.c (36%)	17
submission/190104008.1.c (93%)	submission/190104042.1.c (93%)	26
submission/190103098.1.c (83%)	submission/190104033.1.c (74%)	19
submission/190103098.1.c (83%)	submission/190104019.1.c (74%)	19
submission/190103098.1.c (83%)	submission/190104013.1.c (74%)	20
submission/190103091.1.c (83%)	submission/190104033.1.c (74%)	17
submission/190103091.1.c (83%)	submission/190104019.1.c (74%)	17
submission/190103091.1.c (83%)	submission/190104013.1.c (74%)	20
submission/190103098.1.c (79%)	submission/190104006.1.c (71%)	18
submission/190103091.1.c (79%)	submission/190104006.1.c (71%)	12
submission/160106021.4.c (63%)	submission/190103100.4.c (67%)	10
submission/190104020.1.c (69%)	submission/190104039.1.c (77%)	18
submission/190103100.4.c (63%)	submission/190104017.4.c (59%)	11
submission/190103100.4.c (63%)	submission/190104001.4.c (57%)	9
submission/190104002.3.c (56%)	submission/190104009.5.c (56%)	19
submission/190104001.4.c (52%)	submission/190104017.4.c (54%)	8
submission/160106021.4.c (54%)	submission/190104017.4.c (54%)	8
submission/190104021.1.c (67%)	submission/190104027.1.c (59%)	14

All the following slides are an example from an actual Lab session. The Plagiarism Report is made using MOSS.

Can be really helpful in labs that require deeper understanding of concepts

PS: It even detects variable renaming

submission/190103088.4.c (96%)		submission/190104045.4.c (96%)	
4-40		4-33	

submission/190103088.4.c

```
#include <stdio.h>
#include <stdlib.h>
#define EOL '\n'

int main(){
    int i,k,r,l=0,h=0,p,j;

    int n;
    scanf("%d %d",&n,&k);
    if(k<0)
    {
        p = (-1)*k;
        for(j=0;p;j++){
            r = p%10;
            l = l*10+r;
            p=p/10;
        }

        int b[n];

        for(i=0;i<n;i++){
            scanf("%d",&b[i]);

            if(k>=0)
            {
                b[i]=b[i]+k;
                if(b[i]%7==0)
                    h++;
            }
            if (k<0)
            {
                b[i]=b[i]+1;
                if(b[i]%7==0)
                    h++;
            }
        }
    }
}
```

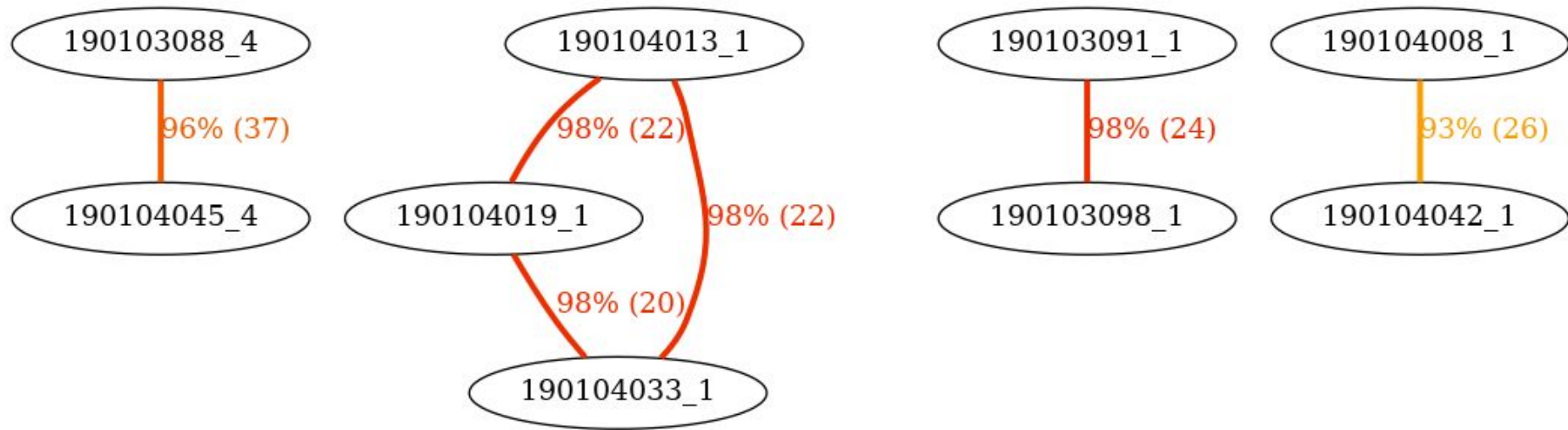
submission/190104045.4.c

```
#include <stdio.h>
#include <stdlib.h>
#define EOL '\n'

int main(){
    int i,k,r,l=0,h=0,p,j;
    int n;
    scanf("%d %d",&n,&k);
    if(k<0)
    {
        p=(-1)*k;
        for(j=0;p;j++){
            r=p%10;
            l=l*10+r;
            p=p/10;
        }

        int b[n];
        for(i=0;i<n;i++){
            scanf("%d",&b[i]);
            if(k>=0)
            {
                b[i]=b[i]+k;
                if(b[i]%7==0)
                    h++;
            }
            if(k<0)
            {
                b[i]=b[i]+1;
                if(b[i]%7==0)
                    h++;
            }
        }
        printf("%d",h);
        return 0;
    }
}
```

Made using Mossum: Moss Visualizer
Format: Percent (Lines Matched)





Other works

- Using for more labs, instructors can track individual student progress and help them improve
- Adding support for question tagging, using this tagging automatically generate new labs for the professor.
- Dashboard for students so that they can also find ways to improve their code quality and performance.
- Lightweight OS containing the platform so that code can be run within user's machine removing the need for processing on a central server.

Questions?

Thank You

"Programming isn't about what you know; it's about what you can figure out."