E-Lab

Web based system for solving and automatic assessment of programming problems

Tomche Delev
tdelev@finki.ukim.mk
Dejan Gjorgjevikj
dejan.gjorgjevikj@finki.ukim.mk

Faculty of Computer Science and Engineering

September 2012

Overview

- What is E-Lab?

Overview

- What is E-Lab?
- Motivation

Overview

- What is E-Lab?
- Motivation
- The problems

Overview

- What is E-Lab?
- Motivation
- The problems
- The solution

Overview

- What is E-Lab?
- Motivation
- The problems
- The solution
- E-Lab philosophy

Overview

- What is E-Lab?
- Motivation
- The problems
- The solution
- E-Lab philosophy
- Architecture

Overview

- What is E-Lab?
- Motivation
- The problems
- The solution
- E-Lab philosophy
- Architecture
- Conclusion

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses
- Everything is done in a web browser

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses
- Everything is done in a web browser
- Using the power of version control systems

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses
- Everything is done in a web browser
- Using the power of version control systems
- Sandboxed execution

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses
- Everything is done in a web browser
- Using the power of version control systems
- Sandboxed execution
- Supporting different types of problems in several programming languages (C, C++, Java)

What is E-Lab?

- *E-Lab* is a system for solving and auto-grading programming problems from introduction programming courses
- Everything is done in a web browser
- Using the power of version control systems
- Sandboxed execution
- Supporting different types of problems in several programming languages (C, C++, Java)
- Designed for scalability and extension

Motivation

- Programming is one of the essential practical skills in computer science curriculums

Motivation

- Programming is one of the essential practical skills in computer science curriculums
- Great popularity among high school students resulting in large introduction classes with several hundreds of students enrolled

Motivation

- Programming is one of the essential practical skills in computer science curriculums
- Great popularity among high school students resulting in large introduction classes with several hundreds of students enrolled
- But programming is not easy!

Motivation

- Programming is one of the essential practical skills in computer science curriculums
- Great popularity among high school students resulting in large introduction classes with several hundreds of students enrolled
- But programming is not easy!
- Involves solving a lot of basic algorithmic examples (1/3 of the time)

The problems

- The context

The problems

- The context
  - X * 100 students

The problems

- The context
  - X * 100 students
  - Divided in groups up to 20

The problems

- The context
  - X * 100 students
  - Divided in groups up to 20
  - 3 - 6 problems a week

The problems

- The context
  - X * 100 students
  - Divided in groups up to 20
  - 3 - 6 problems a week
- The problems

The problems

- The context
  - X * 100 students
  - Divided in groups up to 20
  - 3 - 6 problems a week
- The problems
  - Each instructor should examine and assess up to 120 solutions in 15 minutes

The problems

- The context
  - X * 100 students
  - Divided in groups up to 20
  - 3 - 6 problems a week
- The problems
  - Each instructor should examine and assess up to 120 solutions in 15 minutes
  - Students' feedback

The problems

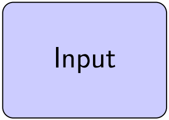
- The context
  - X * 100 students
  - Divided in groups up to 20
  - 3 - 6 problems a week
- The problems
  - Each instructor should examine and assess up to 120 solutions in 15 minutes
  - Students' feedback
  - Loosing focus from the actual programming

The solution

- Algorithmic type of problems

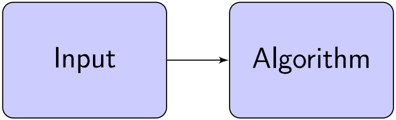
The solution

- Algorithmic type of problems


Input

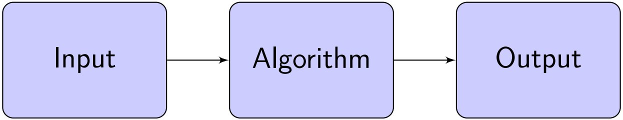
The solution

- Algorithmic type of problems



The solution

- Algorithmic type of problems



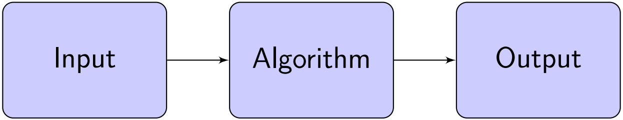
Input

Algorithm

Output

The solution

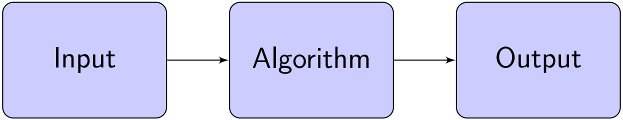
- Algorithmic type of problems



- Used in competitive programming systems

The solution

- Algorithmic type of problems



- Used in competitive programming systems
- Importance in programming to have working solution

E-Lab Goals

- Better organization and implementation of programming exercises

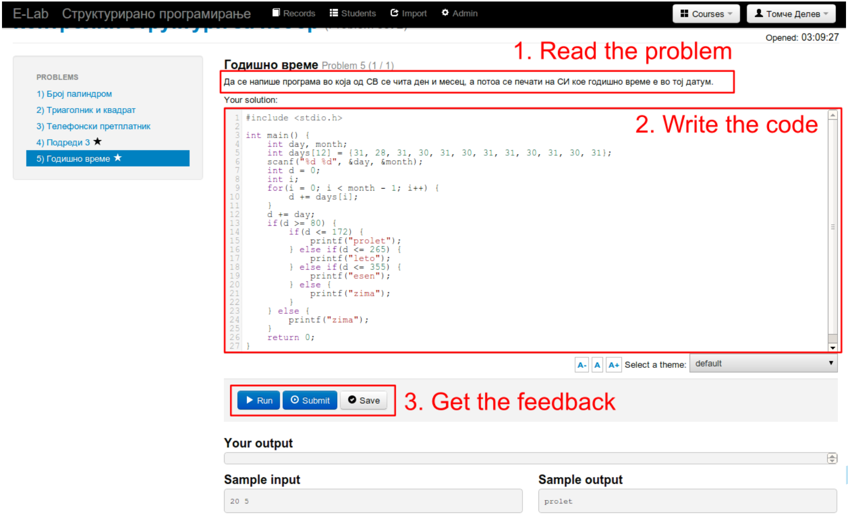
E-Lab Goals

- Better organization and implementation of programming exercises
- Motivate the students with continuous feedback

E-Lab Goals

- Better organization and implementation of programming exercises
- Motivate the students with continuous feedback
- Shift the role of of the instructors from teachers and graders to motivator

Integrated problem view



Opened: 03:09:27

1. Read the problem

Годишно време Problem 5 (1 / 1)

Да се напише програма во која од СВ се чита ден и месец, а потоа се печати на СИ кое годишно време е во тој датум.

Your solution:

2. Write the code

```c
#include <stdio.h>

int main() {
    int day, month;
    int days[12] = {31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31};
    scanf("%d %d", &day, &month);
    int d = 0;
    int i;
    for(i = 0; i < month - 1; i++) {
        d += days[i];
    }
    d += day;
    if(d >= 80) {
        if(d <= 172) {
            printf("prolet");
        } else if(d <= 265) {
            printf("leto");
        } else if(d <= 355) {
            printf("esen");
        } else {
            printf("zima");
        }
    } else {
        printf("zima");
    }
    return 0;
}
```

Select a theme: default

3. Get the feedback

Your output

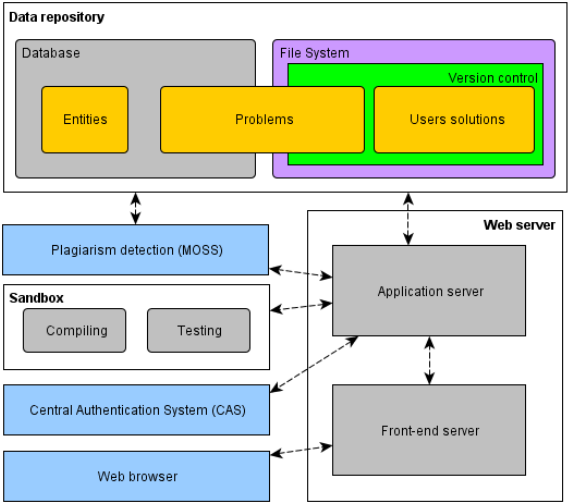
Sample input

```
20 5
```

Sample output

```
prolet
```

System architecture



Conclusion

- Directly address the main organizational problems

Conclusion

- Directly address the main organizational problems
- Simplifying and improving the process of creating and managing programming problems

Conclusion

- Directly address the main organizational problems
- Simplifying and improving the process of creating and managing programming problems
- Centralized repository (distributed)

Conclusion

- Directly address the main organizational problems
- Simplifying and improving the process of creating and managing programming problems
- Centralized repository (distributed)
- E-Lab is not the silver bullet

Questions

http://e-lab.finki.ukim.mk

Thank You

Questions?