Introduction to Cafe-Grader

Preamble

This document is part of Algorithm Design (2110327) course material. Its intended readers are the students and teaching assistants of the class. Prerequisite knowledge for this series includes computer programming, discrete mathematics and data structure. This document is always a work in progress. The latest version of the document can be downloaded at http://www.nattee.net/teaching.

This document describes the use of automated quiz and assignment system newly implemented for the class. Tutorial of the system is also included.

This document was last edited on Monday, May 17, 2010

Introduction

The long-established wisdom is that students are required to do their assignment. The more they completed, the more they are proficient in the art. For Algorithm Design (2110327) of this academic year (2010), we put strong emphasize on this wisdom. We introduce a system called **cafe-grader** to help students, teaching assistants and lecturers incorporate assignment to the core of the class.

Cafe-grader is an automated program evaluation platform. The administrators of the platform, i.e., the lecturers and/or teaching assistant, will set up programming problems and put them in to the platform. The students then read the problems, solve them by writing C or C++ program and submit the programs to the system. After that, Cafe-grader will run the submitted programs several times with different test data which is also set-up by the lecturers. Cafe-grader will compare the results of the submitted programs with the expected outputs and give the score of evaluation accordingly.

With cafe-grader, assignment evaluation is streamlined. Lecturer/TA involvement in grading is reduced to simply design the problem and generate test data while grading will be done by cafe-grader will do the rest. As for the student, the score of the evaluation will be known almost instantly and they are allowed to correct their submission as long as they wish to do so. This eliminates turnaround time between submission and evaluation.

In Algorithm Design class, we will use Cafe-grader in two aspects. Firstly, it will serve as a programming assignment evaluation platform. Student will be assigned several homework during the course and they are expected to turn in their solution to Cafe-grader in due time. The time limit for the homework is about two weeks on average. Secondly, it will serve as a Quiz platform. Quizzes are scheduled approximately every two weeks. The problem of the quiz will not be made public until the start of the quiz. The students are expected to complete the problem in the given time, roughly about 90 minutes. Their answers must be submitted to cafe-grader as well.

Cafe-grader is free and open source software publishing under MIT license. It was designed and develop mainly by Dr. Jittat Fakcharoenphol and Pramook Khungurn from Kasetsart University and also few additional contributors. The web-based frontend uses Ruby on Rails framework while the backend is written in Ruby gluing the 3rd party sandbox system written in C.

Usage Overview

The students use cafe-grader in two aspects: 1) as homework platform and 2) as quiz platform. In both aspects, there are three common use cases listed as follows.

1) Task submission: This is the most important task. The users will submit their works to the system and wait for the scoring of their works.
2) Score report: Users can check whether their submission is correct, i.e., received full mark in the evaluation. Scoring is made almost instantly after the users submit their works.
3) Program Testing: Users can use testing system to test their works in the environment of the system.

These use cases require that the user must log-in to the system. The username and password will be given to the student few weeks before the start of the semester. These use cases is the same for both assignment and quiz aspect. The following sections describe how assignment (homework) and quiz are handled in the class.

Homework Procedure

Roughly every two weeks, homework will be posted in the website. Exact schedule can be found in the course syllabus available at http://www.nattee.net/teaching. The homework is given as a PDF document describing the problem that the students should write a program to solve as well as the format of input and output. Usually, the program is required to read input from the keyboard and display the output to the screen. (In java, this means that you should get your input from `System.in` and display output to `System.out`. In C and C++, this could be done by `scanf` and `printf`.)

When the students complete their programs, they should log in to Cafe-grader and submit their work to the system. The students must specify the problem that they are trying to solve. After submission, they should check their score. If there are some mistakes in the program, the system will report that something is wrong. The student is encouraged to fix their program and re-submit it. It is expected that the student should try their best to get the perfect score on every assignments.

Usually, all assignments are due within two weeks after it is assigned. The students can submit their work anytime and can submit as many times as they want until the last minute. After that, the system will refuse to accept any more submission.

Discussion is encouraged. The student should share their idea and help each other in solving the problem. There is only one exception. Do not copy the work of the other. Remember that all your submission history is store in the system and it is constantly revised by teaching assistants and lecturers. Cheating and copying means penalty.

Quiz Procedure

Similar to the homework procedure, the student are required to read the published problem and write a program to solve the problem and submit it to the system. The only different is time limit and proctoring system. Quiz will be done in the computer classroom with TA and lecturers standing as proctors. Discussion is prohibited and cheating is under zero-tolerance. The problem of the quiz will be available only when the quiz starts and submission is available in the classroom only. This means that there are about 90 minutes time slot for solving the quiz.

The students can re-submit their work as many times as they wish during these 90 minutes. Scoring is also made available immediately after submission as well.

Tutorial

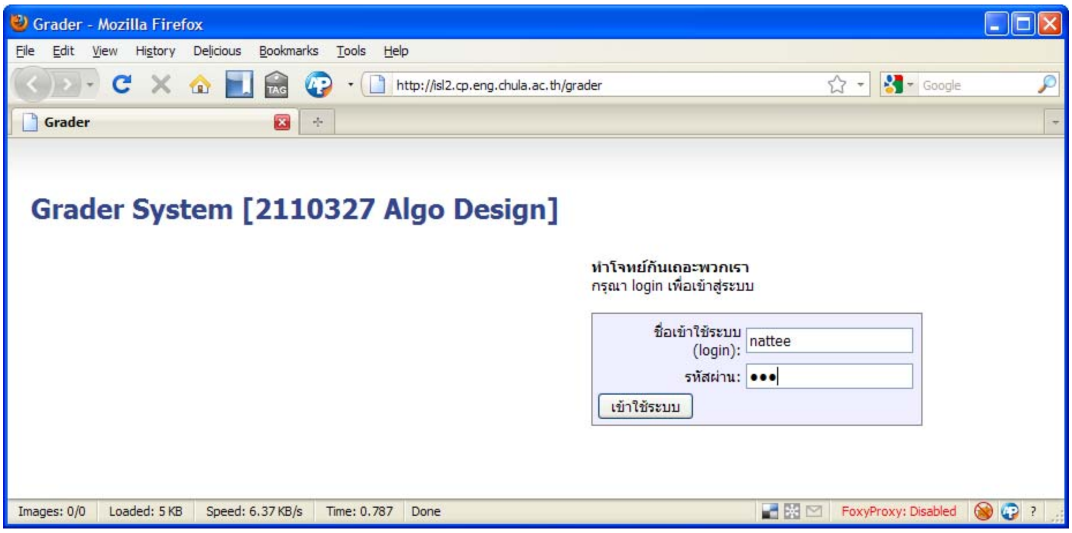
A step-by-step tutorial for homework procedure is presented in this section. In this tutorial, we use demo problems that already have solutions. Please see homework labeled `hw00a`, `hw00b` and `hw00c` in http://www.nattee.net/2110327/2553-a. Let us consider the first one (`hw00a`) for our tutorial.

Homework `hw00a` is a very simple one. It simply asks the reader to write a hello world application. The solution of the problem, which is a C++ source code, is also given in the same page. This source code is what the reader should submit to cafe-grader system.

We will not go into the detail of `hw00a`. The reader is referred to Lab document number 1 of the academic year 2009, available at http://www.nattee.net/2110327/2552-a, for a very brief introduction to C++ and the IDE and also the solution to `hw00a` and `hw00b`. We simply assume that the reader has finished the homework and has the source code for the problem already. The next section describes program submission and score checking. This covers use case 1) and 2).

Program Submission

1. The first step is to log in to the system. Point a web browser to http://isl2.cp.eng.chula.ac.th/grader as indicated in the figure. Enter username and password and then press login button.

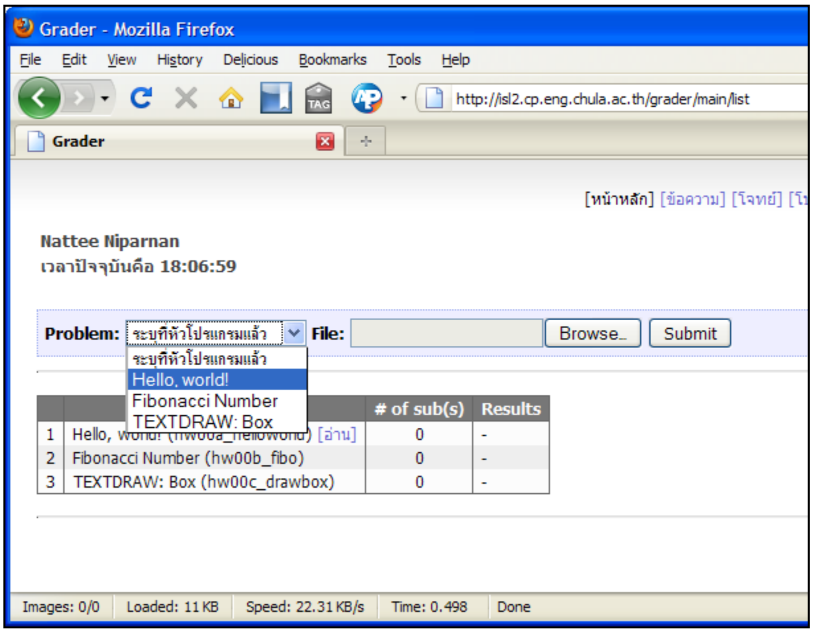

Grader - Mozilla Firefox
File Edit View History Delicious Bookmarks Tools Help
http://isl2.cp.eng.chula.ac.th/grader
Google
Grader
Images: 0/0 Loaded: 5 KB Speed: 6.37 KB/s Time: 0.787 Done
FoxyProxy: Disabled

Grader System [2110327 Algo Design]

ทำโจทย์กันเถอะพวกเรา
กรุณา login เพื่อเข้าสู่ระบบ

ชื่อเข้าใช้ระบบ (login): nattee
รหัสผ่าน: •••
เข้าใช้ระบบ

2. The main screen will appear. This screen is used to submit a program as well as to check the score on each problem. We will submit a source code for homework `hw00a`. The name of the homework is “Hello, world!” so we choose “Hello, world!” from the problem dropdown list.


Grader - Mozilla Firefox
File Edit View History Delicious Bookmarks Tools Help
http://isl2.cp.eng.chula.ac.th/grader/main/list
Grader
Images: 0/0 | Loaded: 11 KB | Speed: 22.31 KB/s | Time: 0.498 | Done

[หน้าหลัก] [ข้อความ] [โจทย์] [โ

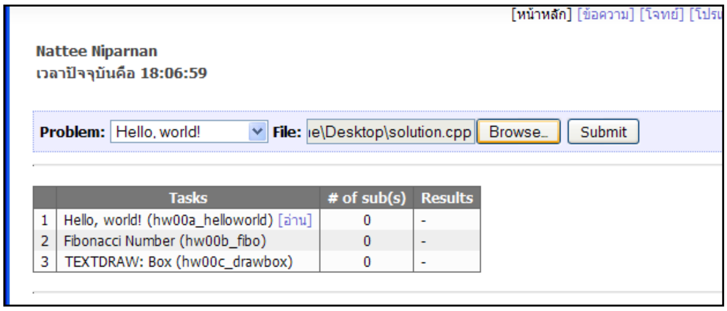
Nattee Niparnan
เวลาปัจจุบันคือ 18:06:59

Problem: ระบุที่หัวโปรแกรมแล้ว File: Browse... Submit

ระบุที่หัวโปรแกรมแล้ว
Hello, world!
Fibonacci Number
TEXTDRAW: Box

		# of sub(s)	Results
1	Hello, world! (hw00a_helloworld) [อ่าน]	0	-
2	Fibonacci Number (hw00b_fibo)	0	-
3	TEXTDRAW: Box (hw00c_drawbox)	0	-

3. Click browse and select file containing the solution to the problem and then click submit.

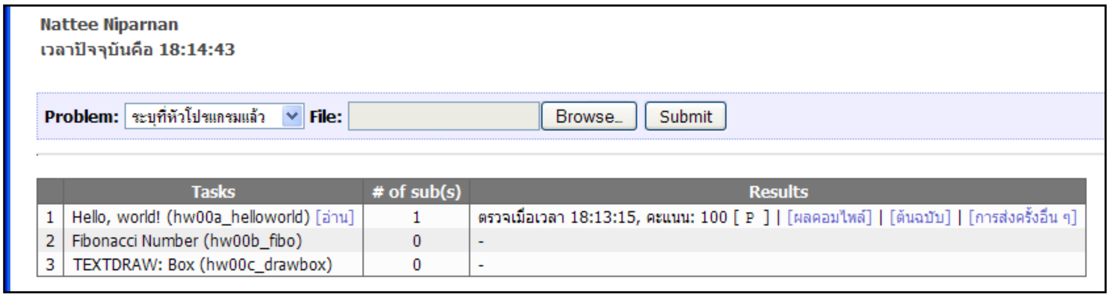

[หน้าหลัก] [ข้อความ] [โจทย์] [โปร

Nattee Niparnan
เวลาปัจจุบันคือ 18:06:59

Problem: Hello, world! File: ie\Desktop\solution.cpp Browse... Submit

	Tasks	# of sub(s)	Results
1	Hello, world! (hw00a_helloworld) [อ่าน]	0	-
2	Fibonacci Number (hw00b_fibo)	0	-
3	TEXTDRAW: Box (hw00c_drawbox)	0	-

4. The screen will display submission time and the number of submission will be increased by one.
5. Wait a few seconds. The system usually takes around 10 seconds to evaluate each submission. This means that if there are 100 concurrent submissions, the last one will be evaluated around 15 minutes later. In most case, half a minute is suffice. Click refresh on the browser. The main screen will show the evaluation result.

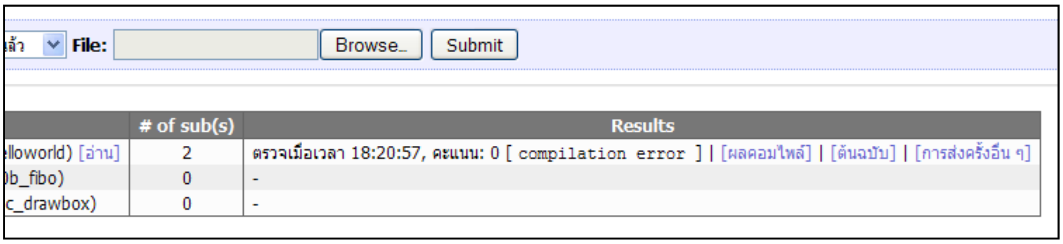

Nattee Niparnan
เวลาปัจจุบันคือ 18:14:43

Problem: ระบุที่หัวโปรแกรมแล้ว File: Browse... Submit

	Tasks	# of sub(s)	Results
1	Hello, world! (hw00a_helloworld) [อ่าน]	1	ตรวจเมื่อเวลา 18:13:15, คะแนน: 100 [P] | [ผลคอมไพล์] | [ต้นฉบับ] | [การส่งครั้งอื่น ๆ]
2	Fibonacci Number (hw00b_fibo)	0	-
3	TEXTDRAW: Box (hw00c_drawbox)	0	-

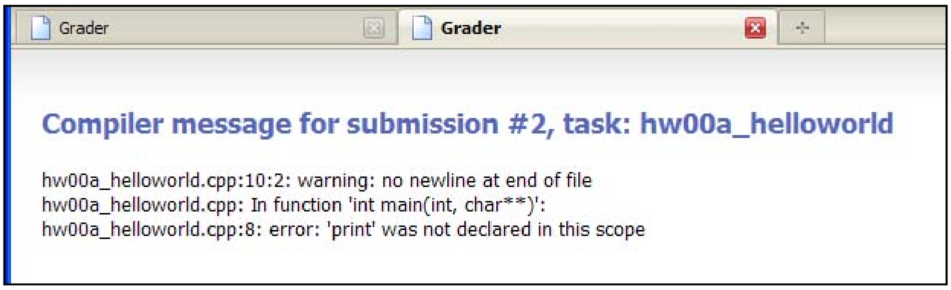
6. The result is given as a score and a result of each test data. The full score for each problem will always be 100. If the score of 100 is shown, it means that the submission is correct. After the score, result of each test data is shown. The string "`[ P ]`" means that the submission "pass" one test data.
7. Let us assume that the submitted program contains error, i.e., it could not be compiled by the grader. For this particular homework, let us edit the "`printf`" function to simply

“print” (with no f at the end). This will cause compilation error. The result will be listed as “[compilation error]” indicating that cafe-grader cannot compile the submitted program.



	# of sub(s)	Results
lloworld) [อ่าน]	2	ตรวจเมื่อเวลา 18:20:57, คะแนน: 0 [compilation error] | [ผลคอมไพล์] | [ต้นฉบับ] | [การส่งครั้งอื่น ๆ]
)b_fibo)	0	-
c_drawbox)	0	-

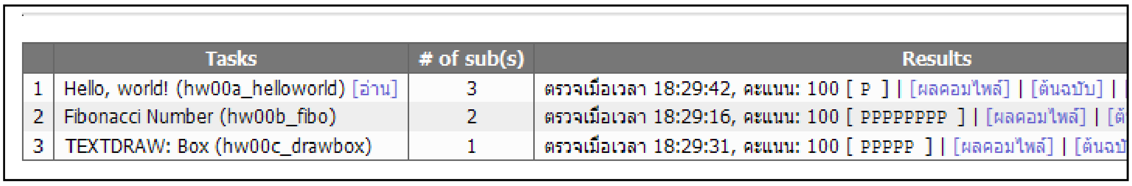
8. Clicking on “compilation result” (ผลคอมไพล์) will display the compilation error message. The reader can use this to fix the source code.



Compiler message for submission #2, task: hw00a_helloworld

```
hw00a_helloworld.cpp:10:2: warning: no newline at end of file
hw00a_helloworld.cpp: In function 'int main(int, char**)':
hw00a_helloworld.cpp:8: error: 'print' was not declared in this scope
```

9. The reader should fix the source code back to the original version and then resubmit it to the system. Also, let us submit all other solutions for hw00b and hw00c as well, the solution can also be found at http://www.nattee.net/2110327/2553-a. After submission, the main screen should be as follow.



	Tasks	# of sub(s)	Results
1	Hello, world! (hw00a_helloworld) [อ่าน]	3	ตรวจเมื่อเวลา 18:29:42, คะแนน: 100 [P] | [ผลคอมไพล์] | [ต้นฉบับ] |
2	Fibonacci Number (hw00b_fibo)	2	ตรวจเมื่อเวลา 18:29:16, คะแนน: 100 [PPPPPPPP] | [ผลคอมไพล์] | [ต้
3	TEXTDRAW: Box (hw00c_drawbox)	1	ตรวจเมื่อเวลา 18:29:31, คะแนน: 100 [PPPPP] | [ผลคอมไพล์] | [ต้นฉบั

It can be seen that the test results are different amongst the three problems. The number of letter ‘P’ indicates the number of test data. In the third problem (hw00c), there are five test cases and thus “[PPPPP]” indicates that this submission passes all test cases. This results in full 100 marks for the problem.

There are others indication of result which are summarized in this table

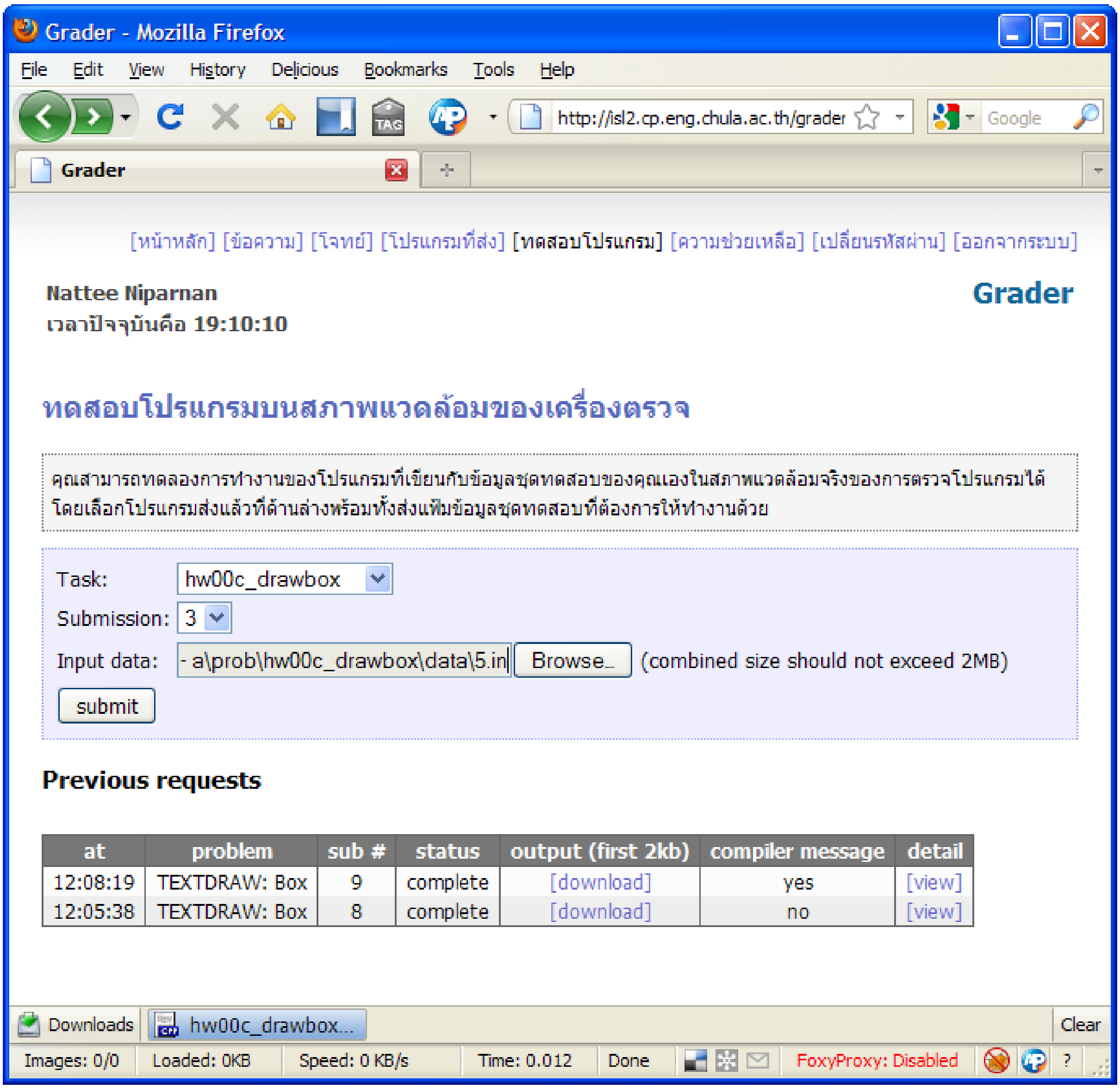
P	The program passes the test case
x	The program crashed
-	The program produced wrong output
T	The program used more time than allowed.

Thus, a test result of [PP-Px] means that the program passes three out of five test cases, failed one and crashed on the last test case. This will result in 60/100 score since each P equals to 20 marks in this problem.

Test Environment

Cafe-grader runs under Linux. In most case, it behaves exactly the same as in windows. Sometime, it is not the same. This is especially true when the problem limits the memory usable by the program. The students can test their program under the grading environment by using the “test interface” (ทดสอบโปรแกรม) screen. This is use case 3) described above.

In this screen, a student can select the submitted program by choosing the problem and submission number. A student must also provide input to the program. The system will run the submitted program with the provided input and then give the result back to the student. The student can check the output produced by the submitted program under grader environment.



[หน้าหลัก] [ข้อความ] [โจทย์] [โปรแกรมที่ส่ง] [ทดสอบโปรแกรม] [ความช่วยเหลือ] [เปลี่ยนรหัสผ่าน] [ออกจากระบบ]

Nattee Niparnan
เวลาปัจจุบันคือ 19:10:10

Grader

ทดสอบโปรแกรมบนสภาพแวดล้อมของเครื่องตรวจ

คุณสามารถทดลองการทำงานของโปรแกรมที่เขียนกับข้อมูลชุดทดสอบของคุณเองในสภาพแวดล้อมจริงของการตรวจโปรแกรมได้ โดยเลือกโปรแกรมส่งแล้วที่ด้านล่างพร้อมทั้งส่งแฟ้มข้อมูลชุดทดสอบที่ต้องการให้ทำงานด้วย

Task: hw00c_drawbox
Submission: 3
Input data: - a\prob\hw00c_drawbox\data\5.in Browse... (combined size should not exceed 2MB)
submit

Previous requests

at	problem	sub #	status	output (first 2kb)	compiler message	detail
12:08:19	TEXTDRAW: Box	9	complete	[download]	yes	[view]
12:05:38	TEXTDRAW: Box	8	complete	[download]	no	[view]

Useful tips

These are common tips in writing a program for submission to cafe-grader.

- In most cases, the program to be submitted to cafe-grader is a console application that reads data from standard input (keyboard) and print result to standard output (screen). Unless indicate otherwise in the problem, the student should assume this.
- Our recommended development environment is dev-c++. In dev-c++, it is default by the IDE that the new project ends with `system(“PAUSE”);` function call. This is good for debugging the system. However, this is not valid in cafe-grader system. If the student includes this line in the submitted program, the test result will be either 'x' or 'T', depends on the situation.
- It is possible to indicate the problem in the source code. This can be done by putting these lines at the beginning of the source code.

  ```
  /*
  LANG: C++
  TASK: hw00c_helloworld
  */
  ```

  The TASK line indicates the problem of the source code. With these line, it unnecessary to indicate the problem in problem dropdown list when submitting the program.
- There are several other functions of cafe-grader. Its usage should be straightforward enough to be understood by simply navigating around and trying each screen. Simply be noted that what the student need to do is to submit a solution to every problem in the main screen and score full 100 marks on every of them.

Conclusion

This document provides tutorial for cafe-grader system as well as its use cases. By following this document, the reader should be familiar with the system. Overview of the quiz and homework procedure is also described. With this system, the students and the lecturers could have a streamlined approach for assignment and its evaluation.