Online Programming Judge System (UOJ)

By

Tran Quang Trung

Dissertation submitted in partial fulfillment of
the requirements for the
Bachelor of Technology (Hons)
(Information Communication Technology)

JANUARY 2013

Universiti Teknologi PETRONAS
Bandar Seri Iskandar,
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

ONLINE PROGRAMMING JUDGE SYSTEM (UOJ)

By

Tran Quang Trung

A Project Dissertation submitted to the

Information Communication Technology Programme

Universiti Teknologi PETRONAS

in partial fulfillment of the requirements for the

BACHELOR OF TECHNOLOGY (Hons)

(INFORMATION COMMUNICATION TECHNOLOGY)

Approved by,

_____

(Ms. Amy Foong Oi Mean)

UNIVERSITI TEKNOLOGI PETRONAS

TRONOH, PERAK

JANUARY 2013

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

_____

TRAN QUANG TRUNG

ACKNOWLEDGEMENT

First, I would like to express my sincere thanks to my supervisor, Ms. Foong Oi Mean. The supervision and support that she truly help me in my Final Year Project (FYP) progress. Ms. Foong not only gave me significant guidance but also is the one who motivate me to implement it as a real system that people can use. This is the first time that I have done and published a big project like Final Year Project individually, the biggest achievement that I feel satisfied is the project has been implemented so that student can widely use by accessing the URL link above. Without motivation and support from Ms. Foong, my FYP would just stop at demo level; it cannot be completed as one of the little FYP produced by UTP student that is delivered successfully as a real product.

Second, I also would like to send my thanks to all technicians in High Performance Service Center (HPC) of UTP, who supported me for setting up and conduct server testing before deploy the system widely on internet. They also helped me a lot through analysis and error checking.

Last but not least, I would like to express my sincere thanks to Universiti Teknologi PETRONAS, especially Computer and Information Science Department, that has given me a chance to explore and apply my knowledge studied in a useful and interesting project, Final Year Project.

ABSTRACT

This research conducts a study to build an Online Programming Judge system with a mechanism to generate test cases automatically using Particle Swarm Optimization (PSO) algorithm. The system has the function to judge programming code by evaluating the output that the program produced. Based on the problem that it is time consuming for lecturers to manually compile, run and verify every student programs for judging. Moreover, they also need to define test cases for different programming exercises in order to judge student's code. The system is built on the purpose to assist lecturers in Universiti Teknologi PETRONAS in judging code submitted from students and generate test cases for every programming exercise automatically. It also helps UTP students practice and enhancing their programming skills. In this research, details of judging process are explored. Moreover, the mechanism of test cases generation using PSO algorithm is deeply analyzed. The study would focus on the primary structure of PSO and the proposed fitness function to calculate fitness value for each generated test case. There are comparisons between manual and automatic PSO test case generation results that would be conducted to evaluate the efficiency of the proposed method. Finally, conclusion of current results and recommendation for future development are also stated.

LIST OF FIGURES and TABLES

ABBREVIATIONS AND NOMENCLATURES

UTP	Universiti Teknologi PETRONAS
UOJ	UTP Online Programming Judge System
PSO	Particle Swarm Optimization algorithm
HPC	High Performance Service center

TABLE OF CONTENT

CHAPTER 1
INTRODUCTION

1.1 Background of Study

Programming is one of the major skills that IT students in every department of computer science of all universities in the world as a basic step to go higher for their career. Every computer science subject has certain relationships with programming. Needless to say, there are many computer science courses which require programming as the main techniques in order for student to capture the knowledge.

Students acquire programming skills through frequently study and practice. Normally, the lecturer prepares and gives programming assignments to help them apply the learned knowledge into practice. Through those coding activities, student can understand the theory clearly and get more experiences to implement what they have learnt in class into practical code. Most of the higher-level course, for example Operating System, Data Communication and Networking, would require that students have already got require level of programming. However, subjects that mainly focus on programming, such as Structured Programming, Algorithm and Data Structure usually required students to solve primary exercise that involves mathematical algorithm and programming techniques [**1**], those course always have programming assignments with difficult questions.

Normally, upon receiving code submitted from students, the lecturer would manually go through all the source files, for each of them, lecturer need to manually compile, run, key in the test case and verify the output. The student codes is judge based on the structure coding style whereby how efficient that they organize their program into related modules and the way in which variables are named. However, the major evaluation would fall into how much of correctness that output produced by student program match with output predefined by the lecturer.

By manually assessments, lecture can evaluate student ability through the way they organize their program but that would make a lots of difficulties if the number of students are high because it is time consuming for lecturer to give mark after go through all source codes. On the other hand, manual grading requires lecturer repeatedly compile & run, key in one test case,

verify output, stop the program, then recompile & run, key in another test case and so on. This process would consume many time and efforts, moreover the lecturer can only verify student codes with limited number of test case, therefore with program that require thousands of test to check, the testing requirements is not fully covered.

Automatic Online Judge system is built to assist to check the correctness of program code by automatically receives code submitted by programmer, subsequently, it would compile, run and repeatedly key in all predefined input into the running program and capture the output. The output produced by the program is compared with predefines output which are assumed to be correct to judge the precision of the program code.

Automatic Online Assessment System is used for different purposes. In order to assist computer programmers around the world to have the mechanism to practice and enhance their programming skills, many Automatic Online Judge System are built to serve for this purpose. In order to help students around the world to learn and upgrade their programming knowledge, and most important, to participate the annual ACM ICPC Programming contest, University of Valladolid provide UVA Online Judge System (**Figure 1**). The system allow user to read programming exercise then browse the source code and submit. Upon submission, user's code is compiled and run, the system automatically key in predefined input, and then compare output from users with the output predefined by administrators and generate status result. In order to get a correct result, user program outputs must match with predefined outputs from the system.

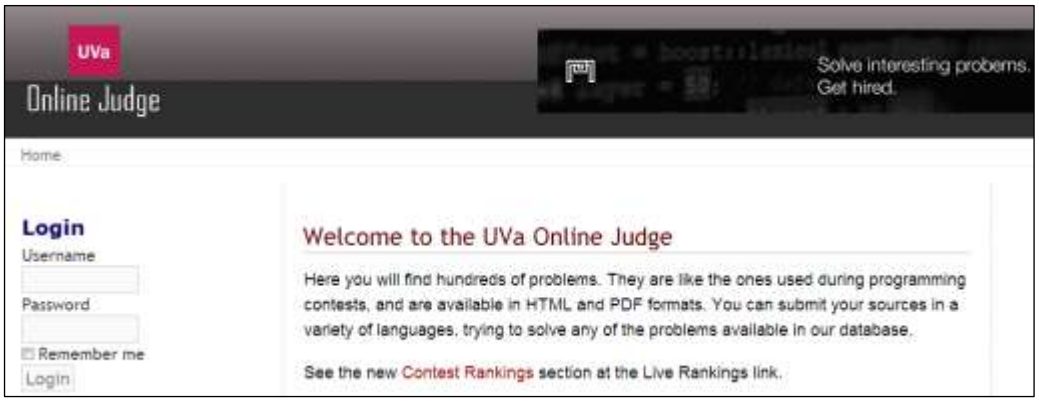

Figure 1: UVA Online Judge (http://uva.onlinejudge.org/index.php).

Automatic Online Judge System is not only used to provide training for IT student, it can also be adopted as recruitment system for many advanced IT company to identify potential programmers who have high level of programming skills that can become excellent employee for their organizations. Facebook Programming Challenge (**Figure 2**) is an automatic online judge mechanism that forces all developers who want to become Facebook employee must pass the test generated from the system before being contacted by a recruiter. The submitted code from user would be checked by two set of test case, the easy and difficult one. Depending on the percentage of precision between users' output with the correct output defined by Facebook and the coding style of programmer, Facebook staff would determine whether the programmer is qualified for the job or not.

Figure 2: Facebook Programming Challenge (https://facebook.interviewstreet.com/recruit/challenges).

Google, the company famous with technologies and innovation, has their own ways to recruit their employees. Instead of delivering problems exercise to potential candidate, they organize an annual programming competition called Google Code Jam (**Figure 3**) which comprises many rounds that candidates must compete to each other to get the value position. For each round, each candidate is provided a list from 4 to 5 exercises with a corresponding input file for each problem. Programmers then write coding, then use the input file to key in their running code, the output generated from their program are printed into a text file and submit back to Google Code Jam online judge system. The system will compare the output from candidate with the output predefined from the system to determine the winners. By doing this, there is an advantage is the

system can run faster because it does not compile and run user code on server but cannot detect cheat programmers who share correct output among each other's. To avoid this, Google ask candidate to submit their code along with the output, the code will be compared & verified if the output is totally correct or if they identify similar wrong outputs from two or more different candidates. The cheated candidates would be rejected and blacklisted permanently.

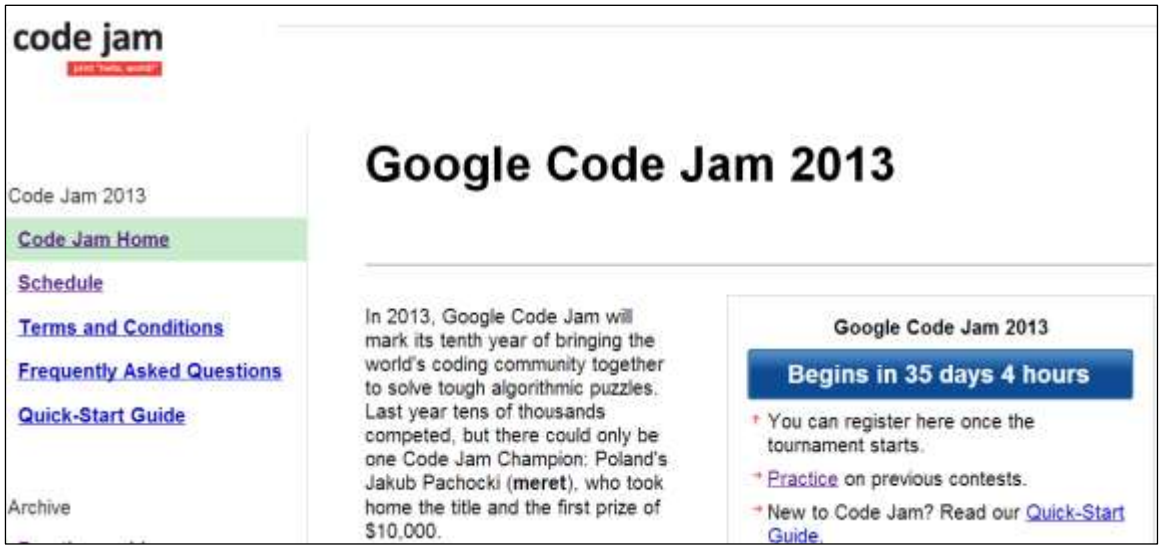

Figure 3: Google Code Jam (http://code.google.com/codejam).

1.2 Problem Statement

- Lecturer has to manually open the source code, compile, run, input test case and verify the output for every student's program for programming assignment.
- There is no mechanism to practice online programming within UTP with the problem created by lecturer.
- There is no mechanism to generate test case, lecturer must generate test case data manually, and thus the number generated test case is limited.
- There is no automatic online judge system for students to practice before participating Malaysian National ACM ICPC Programming Contest which is going to be organized at Universiti Teknologi PETRONAS in September 2013.

1.3 Objectives of study

- To propose a new enhanced methods for automatic test case generation base on PSO algorithm.

- To build an Online Programming Judge System which receives program written in C++ or Java submitted by student, then running, compare output from the program with the predefined output to judge the correctness of the code.
- To build a system, this will be implemented as a practicing resource for students to enhance their programming skills and participating Malaysian National ACM ICPC Programming Contest at UTP for the year 2013.

1.4 Scope of study

- The main purpose of this project is to produce system which is able to do judging of C++ or java program submitted by students. Due to resource limitations and time constraints, there are several restrictions that need to be taken into account. The project only focuses on programs written in two programming languages which are C++ or Java. This means program written in other language, for instance C#, Perl or VB cannot be accepted by the system.
- The target users of the research are students; it can only receive a single file program to be submitted. Therefore, a complex programming project which requires several file systems with many configurations cannot be adopted using the system.

1.5 Limitation of the Research

- Additionally, the program make assessment by key in a set of input into student's running program and compare the output with the predefined output test case, it cannot make assessment base on coding format, logical relationship among modules.
- It can only accept one single files, student cannot submit a complex project with multiple files.
- Assessment might be not fully covered because the number of test case is limited.
- It can only make assessment of simple program with the runtime not exceeding 20s and require a strong hardware server to run.

1.6 Feasibility of the project within scope and timeframe

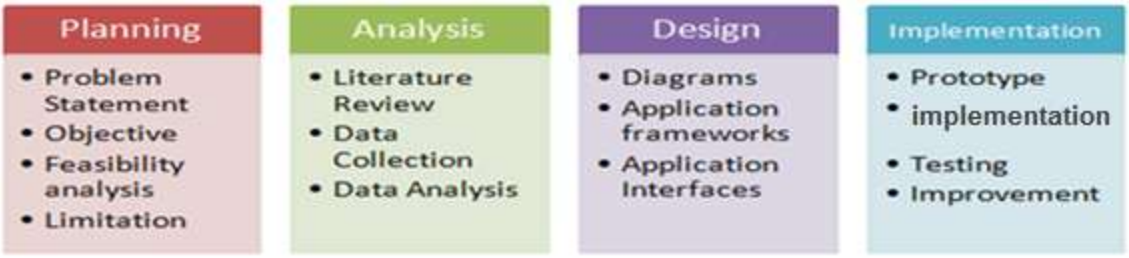

Figure 4: Expected deliverables

CHAPTER 2
LITERATURE REVIEW

2.1. Programming Judge System.

Gaining experience in the design and analysis of algorithms and data structures is one of the most important steps of the initial part of the learning process for students of computer science. The management of related courses can be easily automated using E-Learning platforms, such as Blackboard [**2**] or Moodle [**3**]. However, due to the very nature of programming assignments, for such courses one further step of automation is possible: the E-Learning platform may not only serve as an online document repository and communication medium, but may also perform tasks related to the automated assessment of user-submitted programs.

Historically, such an approach was first adopted for early programming con-tests, mainly in order to decrease the strain on the staff responsible for grading, and to ensure an entirely impartial and immediate judgment of submissions from all participants. This lead to an increase popularity of similar systems designed for training purposes, usually in the form of websites known as online judges, providing their services to potential contestants and other enthusiasts intent on improving their programming skills. But the application of online judges in formal E-Learning, especially in compulsory university tuition for computer science students, turns out to be a somewhat more complex process, which has only recently started gaining popularity [**4**, **5**].

Herein we conduct a study on 2 main judge systems which was developed to serve for different purposes. In the following sections, characteristic for each of three systems will be explored to analyze and evaluate the efficiency and backwards. Finally, the automatic test case generation based on particle swam optimization is analyzed in order to get the basic techniques for our research contribution proposed methods which is going to be implemented in our system.

2.1.1 PC^2-Kit.

The Auto Assessment System (**Figure 5**) developed by California State University, Sacramento used as the main judge system for ACM ICPC Programming Contest in all National, Regional and Final World Round. PC^2 kit is a dynamic, distributed real-time system designed to manage and control Programming Contests. It includes support for multi-site contests, heterogeneous platform operations including mixed Windows and Unix in a single contest, and dynamic real-time updates of contest status and standings to all sites.

PC2 operates using client-server architecture. Each site in a contest runs a single PC2 server, and also runs multiple PC2 clients which communicate with the site server. Logging into a client using one of several different types of PC2 accounts (Administrator, Team, Judge, or Scoreboard) enables that client to perform common contest operations associated with the account type, such as contest configuration and control (Administrator), submitting contestant programs (Team), judging submissions (Judge), and maintaining the current contest standings (Scoreboard).

Figure 5: The **PC^2** Programming Judge System.

PC2 clients communicate only with the server (**Figure 6**) at their site, regardless of the number of sites in the contest. In a multi-site contest, site servers communicate not only with their own clients but also with other site servers, in order to keep track of global contest state. The following communication requirements must therefore be met in order to run a contest

using PC2: (1) a machine running a PC2 server must be able to communicate via TCP/IP with every machine running a PC2 client at its site; and (2) in a multi-site contest, every machine running a PC2 server must be able to communicate via TCP/IP with the machines running PC2 servers at every other site. In particular, there must not be any firewalls which prohibit these communication paths; the system will not operate if this communication is blocked. It is not necessary for client machines to be able to contact machines at other sites.

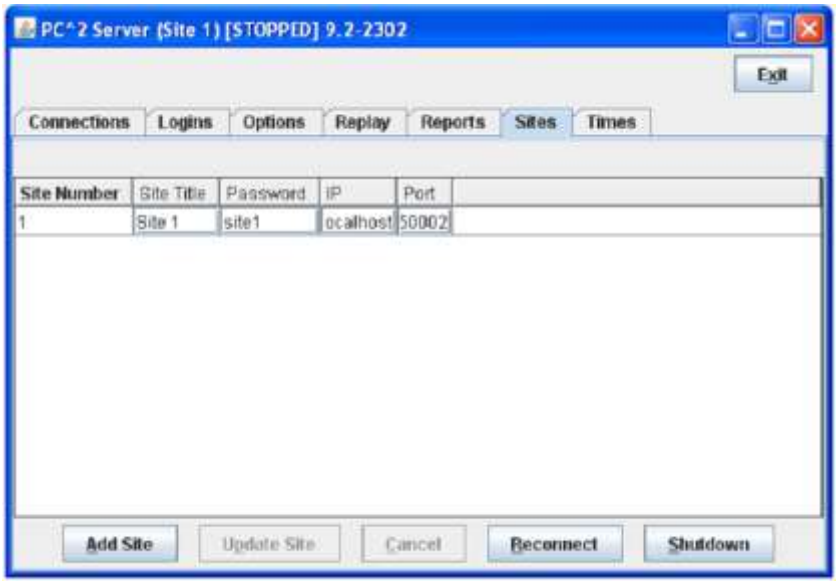

Figure 6: PC^2 Server Graphic User Interface (GUI).

Client users need to log in in order to use PC^2 kit. To login to PC^2, user clicks once on the Name box on the login screen, then enter assigned team ID, press the TAB key or click on the Password box, then enter assigned password. After entering team name and password, click on the Login button. Once the system accepts user login, contestant will be at the PC2 Main Menu screen (**Figure 7**). Note that the team ID and the team's site location are displayed in the title bar, and the amount of time remaining in the contest is displayed below the title bar.

- The **Submit Run** tab near the top of the screen displays the run submission screen.
- The **Problem** box will display a drop-down list of the contest problems; choose the problem for which contestant wish to submit a program (called a "run") to the Judges (in the example, a problem named "Hello" has been chosen).

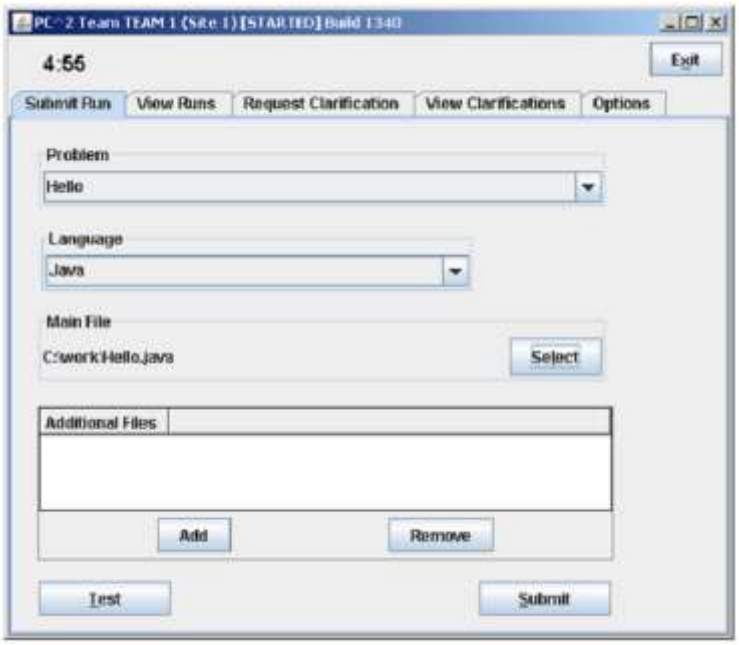

Figure 7: The PC2 Main Menu Screen.

- The Language box will display a list of the programming languages allowed in the contest; choose the language used by the program that user wish to submit to the Judges (in the example, "Java" has been chosen).
- To submit a program to the Judges, contestant must specify the name of the file containing their main program. Click on the Select button to invoke a "File Dialog" which lets contestant locate and select their main file. The dialog lets them automatically navigate to the correct path and file location (in the example, the main program file "C:\work\Hello.java" has been selected). Select only source code files for submission to the Judges.

Once contestant clicks the **Submit** button. The confirmation dialog shown on the left below will appear; verify the data and then click YES to confirm the submission and send the code to the Judges. When user submission has been received by the Judges they will get a "Run Received" confirmation message as shown on the right below (**Figure 8**).

Figure 8: Confirm Submission & Run Received notification.

The Judge's Response message identifies the submission and includes an indication of whether the submitted program correctly solved the problem or not. The "Status" field on the VIEW RUNS grid will also change to reflect the Judge's response. According to the below run (**Figure 9**), which has the "YES" response, it means that the user's program is run successfully and all the output produced from user program matched exactly with all output on the Judge's site. There are additional 4 possible Judge's Responses which are:

WRONG ANSWER : program was compiled & run successfully but produced incorrect output.

TIME LIMIT EXCEEDS: program was compiled & runs successfully but out of time limit.

RUN TIME ERROR: program was compiled successfully but fail to run due to wrong data structure definition.

COMPILED ERROR: program was not compiled successfully due to syntax error.

Figure 9: A "YES" Judge's Response.

PC2 kit is an efficient programming judge system but it also has some limitation. Firstly the system can only run in local network it must be set up with complex configuration. Moreover, it also consumes a lot of resource because the performance is high. Finally it does not have mechanism to generate test case automatically, therefore, the Judge Committee must manually define the test case.

2.1.2 Top Coder Programming Platform

Top coder programing system is an automatic online programming judge system created and operated by TopCoder (**Figure 10**), a company that administers contests in computer programming. TopCoder hosts fortnightly online competitive programming competitions — known as SRMs or "single round matches" — as well as weekly competitions in design and development. The work in design and development produces useful software which is licensed for profit by TopCoder. Competitors involved in the creation of these components are paid royalties based on these sales.

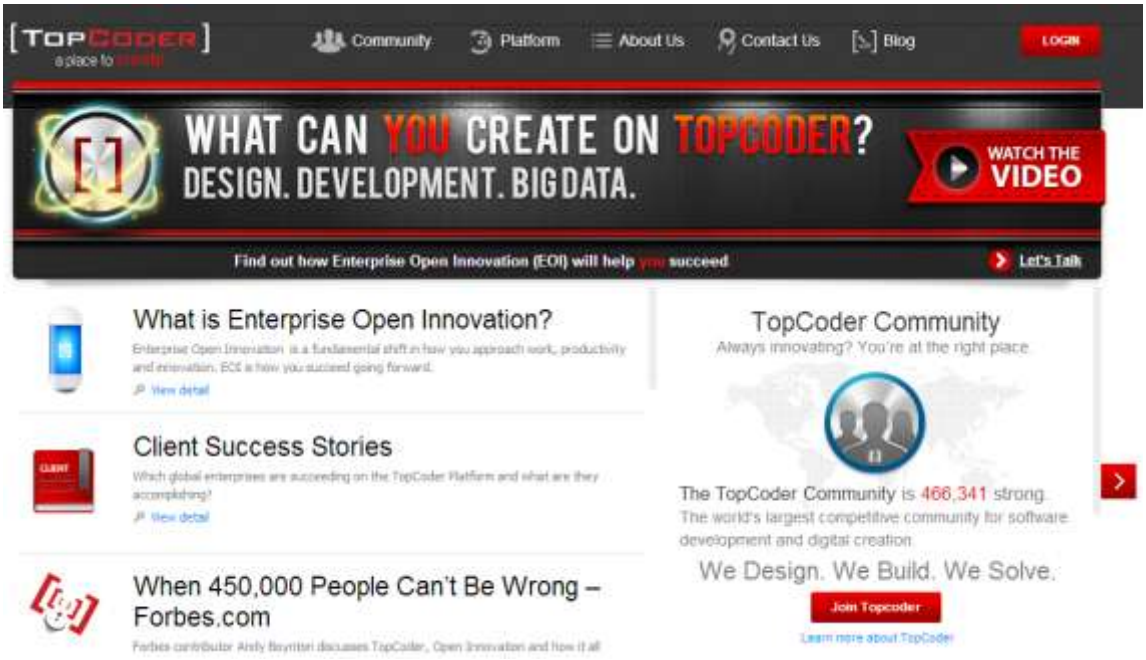

Figure 10: TopCoder Programming web page.

From the contest hosted by TopCoder, statistics, data about excellent programmers are recorded. Those information are provided to potential employers who are seeking for developers that have advanced knowledge in algorithm design.

When participate into TopCoder, user need to download and install TopCoder UML tool (**Figure 11**) which is an java based client log in interface that allow user to enter their own user id and password. Once login, the system is connect to top coder server to provide user numbers of competition round which have programming exercise for them to do. User would finish their code and submit to server through the system. User's code program is subsequently compiled, run and judged. Result is sent back to user with similar judging procedure to PC^2 Kit above.

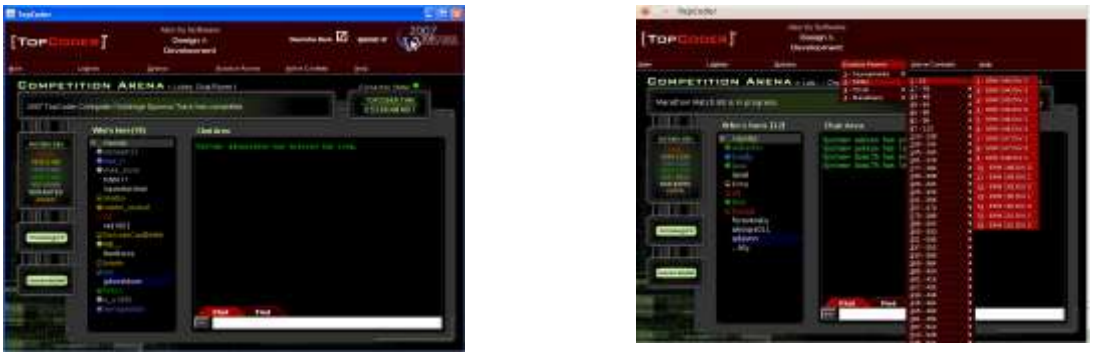

Figure 11: TopCoder User Interface UML programming tool.

TopCoder is efficient programming judge system in terms of qualifying excellent programmers who possess advanced programming skills. However it is not suitable for beginners who have basic knowledge of coding and is seeking programming mechanism to enhance their skills. Therefore it cannot fit for the purpose to teach and motivate knowledge of programming for majority of university students. Finally, similarly to PC^2 Kit system, it does not have mechanism to generate test case automatically.

2.2 Automatic test case generation using particle swarm optimization (PSO).

In this part we are going to explore an existing research conducted about Particle swarm optimization (PSO) [**8**] which is a promising new optimization technique, which models the set of potential problem solutions as a swarm of particles moving about in a virtual search space.

PSO model starts with the random initialization of a population of particles in the search space [**10**]. It converges to the global best solution by simply adjusting the trajectory of each individual particle towards its own best location and towards the best particle of the entire swarm at each generation.

Particle Swarm Optimization is a classic method which has been applied into different fields of computer science [**11**]. The algorithm contains two main processes: Initialization and Iteration. In the initialization, attributes of a swarm that contains many particles is initialized. During Iteration steps, those attributes would be updated based on movements of each particle. An Fitness function would be used to evaluated the best position of each particles to determine the best position of the swarm [**12**].

Let a swarm $X=\{x_1, x_2 \ldots x_n\}$ that have n particles. This swarm is moving in a space S with D dimension. Each particle i (i=1, 2… N) Has an initial position x_i with a velocity v_i. Let pb_i is the best position of particle i in dimension D. Goal of the algorithm is to find the best position of the swarm by evaluating the individual position of each particle. During Initialization process, each particle is assigned initial position and velocity, and then then the fitness function is used to evaluate the initial best position for each particle and best position of the swarm [**11**]. The algorithm of PSO is described in pseudo as follows:

Particle Swarm Optimization [**8**] – Pseudo Code:

For each particle in dimension D **do**:

Initialize each particle's position.

Initialize each particle's velocity.

Initialize the particle's initial best known position.

If fitness of swarm best position less than fitness particle's best position **do**:

Update the swarm's best position.

End for

Repeat:

For each particle i = 1... S **do**:

For each dimension d = 1... N **do**:

Update particle's velocity

Update the particle's position

If fitness of particle's best position less than fitness particle's position **do**:

Update the particle's best position.

If fitness of swarm best position less than fitness particle's best position **do**:

Update the swarm's best position.

End for

End for

Until max iteration**.**

Now **g** holds the best found solution.

For different research objectives, there are different definitions of the fitness function [**12**]. The overall purpose is to evaluate the best position of the whole swarm based to best positions of individual particle in all dimensions.

Let consider a test case which contains many variables as a swarm. Each variable is a particle that has its own type, limitation and attributes. By applying, PSO we can evaluate the best position of the swarm to pick up best swarms that has highest fitness value among those potential swarms. By doing this we can generate and pick up the best test cases to meet our primary objectives: To generate test case automatically and pick up the best candidate that has the highest fitness value based on PSO. The details implementation of algorithm and fitness function would be described in next chapter.

CHAPTER 3
METHODOLOGY

On this chapter, a formalized approach is proposed to evaluating usability and correctness PSO algorithm of the online programming judge system. The Online Programming Judge system (UOJ) has been developed since my internship (October 2011 to May 2012). I developed the system because I want to learn more about the mechanism of evaluating user's program by monitoring the output from their running codes and compare with the standard output. However, after finish the internship, there is a requirement to have Online Judge mechanism to support the CIS department in organizing the Malaysian National ACM ICPC Programming Contest 2013. That's why I proposed my UOJ and gets approval from CIS Head Department. Prior to implementation, I need to conduct testing properly and ensure that the system is running perfectly. Therefore this chapter is described the methodology that I applied to implement UOJ system. On the other hand, I also explained the method to implement automatic test case generation algorithm using PSO. All would be in order as follows:

The chapter is divided into 4 sections and a summary. Section 3.1 introduces RAD approach, the research methodology that is used as main approach to deliver the system. Next, section 3.2 presents all project activities that need to be performed to implement the system that followed the proposed methodology. This section presents whole system structure including different modules, database architecture, and system flow in which the whole process is conducted. On the other hand, the automatic test case generation algorithm, an enhancement of the existing PSO would be proposed to implement as the research contribution in our project. Finally, section 3.4 describes all tools and software involves that are required to develop the project.

3.1 Prototyping based methodology

Rapid Application Development methodology (RAD), a software development methodology that produces rapid product prototyping with minimal planning. The process involves iteration development and enhancement until the desired result is reached. Using RAD, speed of development can be increased since early part can be delivered to user. Moreover, with the involvement of from the early stages, final product quality would be improved.

Prototyping based methodology, one type of software development methodology which follows RAD concept that allow developers concurrently analysis, design and developing portions of the system to demonstrate and testing before the final version is completed. With the use of prototyping based methods, users are invited to qualify the product quality since the beginning and the whole development cycle. Based on user‘s feedback and appreciation after every iteration, the enhanced prototype is re-analyzed, re-designed and re-development. Iteration development steps is continued to be taken until the final satisfaction version is completed.

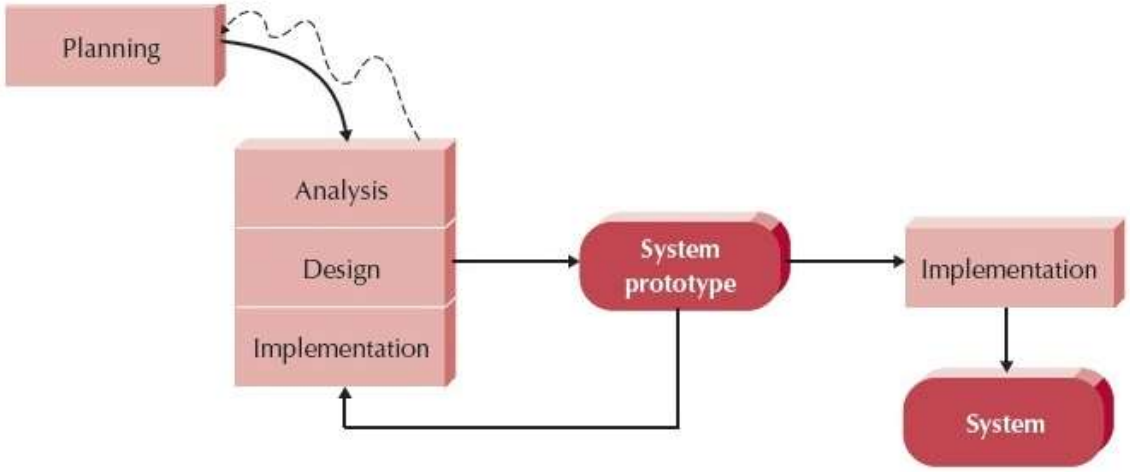

Figure 12: The prototyping-based methodology [**9**]

During the planning process, developer collects and defines essential requirements as much as possible. Those features much be understood in details by conducting interview with numbers of user to get all aspect covering the existing system. The purpose of those actions is to understand problem statements and to recognize the scope and system limitation before starting to save time, cost and efficient achievement.

Upon planning, system architecture, database application is analyzed, explored and compare with the reference to existing research topic. On the other hand, user requirement must be clearly defined to summarize clarified data which is significant important during design and development phase. On the Design phase, developer is required to provide frame work structure to deploy applications. This is achieved by designing systematic diagram (Activity Diagram, Use Case Diagram, and Class Diagram). At the end of the stages, proposed user interface is generated and provided back to user to get evaluation feedback and necessary recommendations.

Implementation is the final phase of the process, in this stage, the implementation and actual installation is performed to deliver the final product to user. Developers totally involve with technical activities to move the project from development status into production status. During this final step, documentation regarding user manual, system architecture are also produced to provide usage direction to users and for future updated and maintenances.

3.2 Project Activities

This sub-chapter describes all activities that need to be carried out in sequence to collect essential data for the project. The goal objective is to evaluate the effectiveness of the Online Programming Judge System (UOJ). History of code submission is conducted to get user manipulation information regarding which type of data set that is most relevant to be gathered and analyzed. The term "data set" here refers to type of programming exercise that are most relevant for UTP IT students. Due to wide range of data set, we only focus on those that are most relevant to user concerns.

3.2.1 Data Collection

Data are categorized into two main types: primary and secondary. Both are lowest unit of information from which measurement and analysis can be done. Primary data is a term for data gathered based on user objective and collected directly from a source of information. Primary data can be considered as raw material which cannot be available in any existing research. It also cannot be manipulated because the data itself does not generate any meaning. Primary data are gathered to serve for purpose of researchers.

Secondary data, on the other hand is collected by other parties except users. It has been existed and are not generated from investigators but gathered by other researchers. Secondary data usually contained in public media like newspaper, magazines, or generated from common source including surveys. Data are collected through qualitative methodologies or qualitative researches. By applying those two methodologies, we can take advantage of getting relevant information in the most convenient ways. On the other hand, this type of data collection is economic in term of time and cost. Nevertheless, because this type of data comes from existing research, it only serves and fulfills the research requirements and does not meet our own needs. Therefore, we need to analyze and extract essential features of those data to carry out our own requirements. For this project two strategies are used to collect primary and secondary are as followed:

1. Server Set Up:

   As mentioned earlier, the system has been developed since in May-2012, two semesters before I take Final Year Project. But at that time, it just a stand – alone java application that can only run offline to test separate user program written in C++ and Java language. Since I started my Final Year Project, I need to convert it into an online web based system that allows user access from the internet. It takes me two semesters to learn and do enhancement, it also take another semester to get approval and deploy the system into internet. The processes involve many phases. However, to fulfill the Data Gathering and analysis purposes, I only describe the activities that I take up for testing and collect user submissions.

   Upon getting approval from Head Department of Computer & Information Science, Universiti Teknologi PETRONAS, I have the right to access the university resources. I have set up two servers for testing and gathering data from user. One server is running on High Performance Service (HPC) center and one server running on Academic Block 2. Two servers run the same system application. The HPC server allows users to access, view exercises and submit codes while the one on Block 2 only user for automatic test case generation testing.

The HPC server use test cases produce by ACM –ICPC Malaysia committee test case while the Bock 2 server use test cases generated by the automatic PSO system. All user submissions submitted into HPC server is re-submitted again and compare the result.

2. Primary Data:

   Primary data are collected through user submission codes to fully assess user interaction with the system. To collect primary data, I have sent email to announce about the system to all ICT / BIS student in CIS department to access, view exercise and perform coding to solve them. I also send email request to many excellent programmers, the ones I have met during National ACM ICPC Malaysia 2012 to request them to solve difficult problems. The purpose is I want to have as much primary submission codes from different category of users as possible. Those raw data is valuable for later analysis.

3. Secondary Data:

   In order to set up the system, I have request material, problem description, exercises from many sources. It includes input, output text file in order to set up exercise. Those are secondary data are used to judge user codes by comparing output that their program produced with the standard output defined by authorized professional.

3.2.2 Data Analysis

Data collected for our project is quantitative type which has simple statistics and tabulation of results. To support analysis, statistic of user submission codes, running time, programming language used are recorded and analyzed. The process is conducted as followed. Details of data gathering and analysis, is presented on **Chapter 4: Result and Discussion**.

3.3 System Architecture

Figure 13: System Architecture (http://hpc.utp.edu.my/UOJ).

According to **Figure 13** above, user need to access the system at address http://hpc.utp.edu.my/UOJ to view exercises. Exercise description is retrieved from database and asked user to solve. The system perform judging by receive codes submitted by user through online web browser. Then, it perform compiling, running, key in input test cases from "Input.txt" file into the running program, it would capture the produced output and compare to standard output defined in "Output.txt" file. The user program is considered correct if it match all output in the output file. Finally, result status is sent back to user. There I four possible of status result:

- **Accepted**: Program run successfully and generates correct output.
- **Wrong Answer**: Program run successfully but generates some invalid output.
- **Time Limit Exceeded**: Program run successfully but exceed limit time.
- **Compile Error**: Program was not compiled, syntax error occurs.

Below are Key Milestones and Grant Chart which includes all activities to be carried out for current Final Year Project II.

3.6.1 Key Milestone

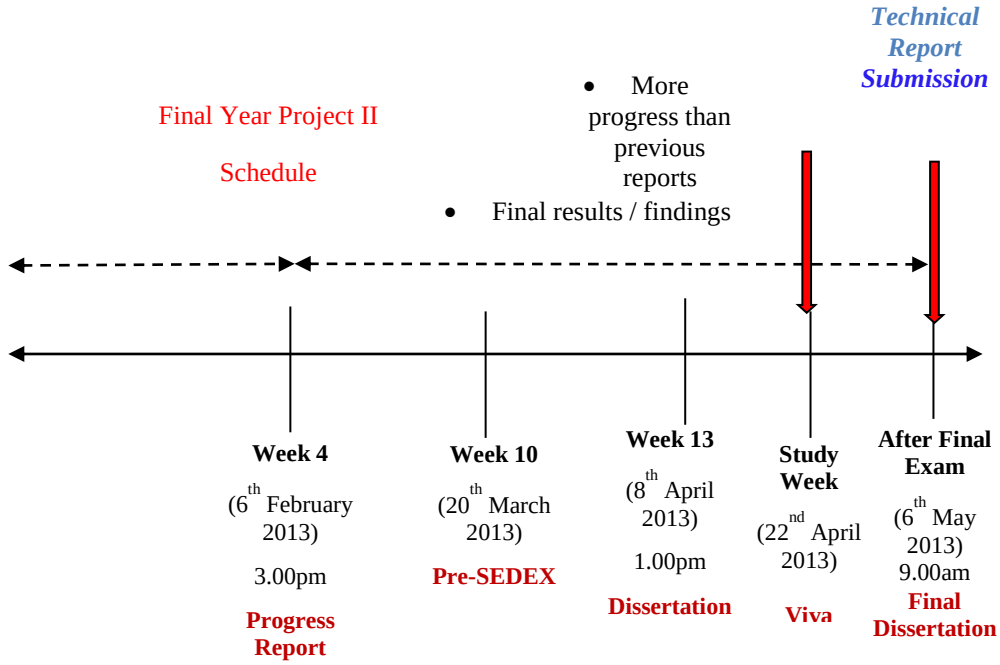

Figure 14: Key milestone.

3.7.2 Grand Chart

Table 3 is a Gantt chart showing the estimated time for every task to be completed to develop & deploy UOJ during Final Year Project II. (http://hpc.utp.edu.my/UOJ)

Table 1: Gantt chart

Delivery Activities	Week													
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14
Analyze existing system structure	■	●												
Proposed enhancement modifications.		■	■	■	■	■								
Methodology Analysis & System Design			■	■	■	■								
Development & Functional Testing					■	■								
Server testing & Deployment (trial)						●								
Server deployment & implementation							■	■	■	■	■	■		
Create data exercise & collect user submission codes.											■	■	●	
Delivery.											■	■	●	
Support Plan & Documentation														●

■ **Process Activity**

● **Milestone**

3.7 Tool and required software.

To develop the system, we need specific development tools that is able to create, debug, maintain, and installation support. Those are relatively software programs which are combined together to accomplish a task or a physical object. As our system contains numbers of back end processing, it is essential to be developed using java platform because java is one of the most popular technology that receives strong supports and contributions from developers worldwide. On the other hand, it must be widely tested and used by number of users. It should be implemented as online application. Combine those two requirements, we used java - servlet web based technology to develop and deploy the system. Below are descriptions for all required tools and software.

3.7.1 Eclipse Integrated Development Environment (IDE)

Eclipse is an open source, integrated development environment (IDE) software application that provides comprehensive facilities to developers for system development. Eclipse IDE is chosen compare to other IDES like NetBeans or Microsoft Visual Studio because it provides most needed features to build Java and Java Web Application System which are required for our project. With the use of Eclipse, many Java and web development tools can be integrated like Java-SE 6, Java EE 7, or graphical HTML, JSP, management and database support are all available. Moreover, web server resource can also be connected, therefore, all feature we expect of an IDE are there (Code, compilation, server, integrating with build system). Those are all reason why I chose Eclipse IDE compare to others.

3.7.2 Java SE Development Kit (JDK).

Java is a free, popular programming language originally developed by Sun Microsystem and now has been owned by Oracles. It is a general-purpose, concurrent, class-based, object-oriented language that can quickly build stand alone as well as web application system. Java programs are portable which means it can be run on different kinds of computer platform without rewrite the code. The requirement for the code to be compiling and run is the java runtime environment must be available. For our

requirement, our web application system would be build using Sun JDK 7 - the latest update version to obtain the highest facilities.

3.7.3 Java Web Application Plugin for Eclipse

In order to develop java web system, we need to have Java Web Application Plugin to be installed in Eclipse. This step can be done by downloading from eclipse main page via eclipse downloading page. Once the software is downloaded, it is integrated inside Eclipse and be ready for web development. Bellows are services component contains tools for developing and interacting with Java Web service. It consists of: [23]

- Extensible Web services wizards for creating Web service and Web services client wizards for consuming Web service.
- Web services Ant tasks for creating and consuming Web services.
- Wizard extensions for the Apache Axis v1.4 and Apache Axis2 Web service runtimes.
- Web services preferences pages.
- Web services frameworks such as the creation framework and finder framework.
- Web Services Explorer, a Web application that let to discover and publish to UDDI, and invoke a WSDL/WSIL via native XML.
- WSDL model.
- WSDL Editor.
- WSDL and WS-I validator.

3.7.4 Apache Tomcat Server

Apache Tomcat Server is an open source web server and servlet container developed by the Apache Software Foundation. It is used implements the Java Servlet and the Java Server Pages (JSP). Both are server host scripts written in java programming language and run under Tomcat. It provides specifications from Oracle Corporation, and provides a HTTP web server environment for Java code to run. It includes tools for configuration and management, but can also be configured by editing XML configuration files. In our project, we install Tomcat inside Eclipse as a server to host java web applications.

3.7.5 MySQL

MySQL is the most popular open source database that is widely used by developers worldwide. It is applicable for both fast growing web property, technology ISV and large enterprise. It also provide administrator interface called MySQL Administrator and interactive components called MySQL Query browser that allow user and admin to write, execute, backup and load database. It helps developers to deliver high performance, scalable database applications with an efficient cost.

CHAPTER 4
RESULT AND DISCUSSION

4.1 Data Gathering

As mentioned in chapter 3 above, data collection is conducted by setting up two servers, one in HPC service center and one in Block 2, UTP. The application system running on HPC is accessed via http://hpc.utp.edu.my/UOJ. Programmers access this website, browse exercise, perform coding and submit their codes. The system would perform judging procedure by compiling, keying test case, compare output and retrieving result. Data regarding user codes, number of submission and type of programming language are recorded to be analyzed and explored.

To perform testing automatic test case generation, the server in Block 2, UTP is set up and running the same application like HPC. The only different is it use test case produced by PSO algorithm to judge user's code. To verify the correctness of the automatic PSO application, the code submitted in HPC is collected and re submitted again in Block 2 server. Result is recorded and analyzed.

4.2 Finding

During the testing process, useful information is recorded regarding:

- Total number of user who registered an account and performs coding activities.
- Total number of submission submitted by users.
- Total of correct answer over total number of submission.
- Numbers of programming language that user choose to solve for all exercises.
- Comparison between two judging system: The HPC server running test case generated by ACM ICPC Professionals and Block 2 Server using judging test case generated automatically by PSO algorithm.
- Evaluation of system usability and efficiency before official deployment.
- Evaluation from UTP professional whether the system is good enough to be implemented as practice resource for National ACM ICPC Malaysia 2013 or not.

4.3 Data Analysis

Figure 15 show that there are 72 account registered. The number of users who is student is 65, the rest is non-student.

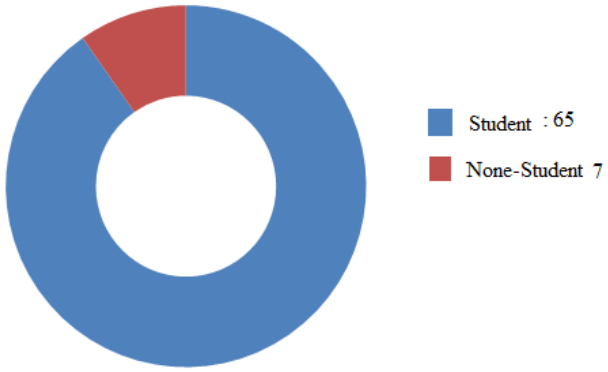

Figure 15: Number of registered users.

Upon implementation, there are 274 submissions from 72 users, with the total number of 174 submissions written in C++ while the number of submission written in Java is 104. Based on those statistics, I obtain that, C++ is the most popular programming languages which account more than 63% over the total submissions. Those data is described in below details:

Total number of submissions: **274**

Average runtime of C++ programs: 6.58 (s)

Average runtime of Java programs: 8.22 (s)

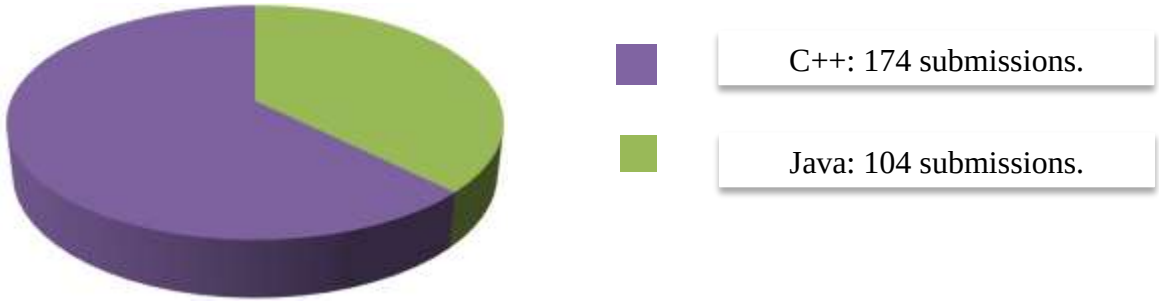

Figure 16: Submission data statistics.

For every submission, the time calculated for compile & running program is recorded. According to **Figure 16,** I obtain that average runtime of program written in C++ is faster than program written in Java.

Upon collecting submission codes from HPC server, I collect those resources and re-submit again in Block 2 server. The purpose is to evaluate the efficiency of automatic test case generation using PSO algorithm by comparing result between HPC server and Block 2 server.

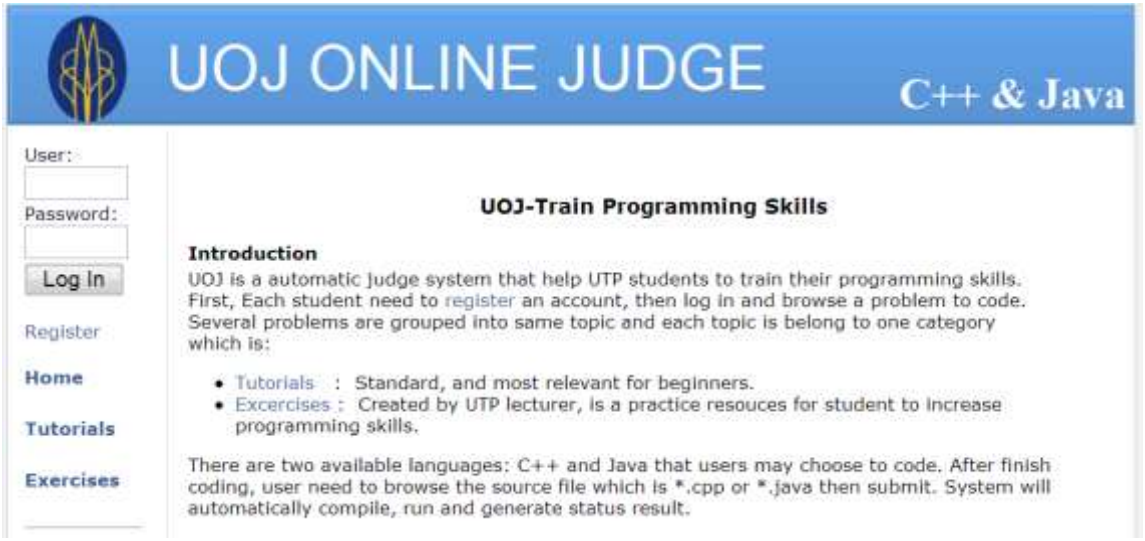

Figure 18: Main User Interface (http://hpc.utp.edu.my/UOJ).

The main interface provides essential instructions for beginner of how to use the system. By accessing main page and follow these directions, user can learn and get into easily. In order to use the system, user need to register an account then log in and browses through problems of coding. Several problems are grouped into several topics and each topic belongs to a category, the categories are:

- Tutorials : Standard, and most relevant for beginners.
- Excercises: Created by UTP lecturers, resources for the practices of students to improve programming skills.

There are two available languages: C++ and Java that users may choose to code in. Once done with coding, user need browse through the source file which is *.cpp or *.java then submit it. System will automatically compile, run and generate status result.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

As a whole, this project is form with the purpose to produce an Online Programming Judge System with built-in Automatic Test Case Generation function using PSO. The system was successfully developed and implemented as an automatic judging system for UTP students to practice and enhances their programming skills. User can access the system via browsing web-browser using this URL: http://hpc.utp.edu.my/UOJ.

The system would help UTP students and other programmers in Malaysia to improve their programming skills. It also assists UTP lecturers to deploy and share programming exercises to their students. It also assists them in judging student program more efficient. The system will be used as an official judging system for the coming national ACM ICPC Programming Contest in 2013.

5.2 Recommendation for future development

Future works suggest proposing a new Hybrid PSO method in order to evaluate fitness value of test cases generated in high complex programming algorithms. This is due to the increasing complexity of programming problems that would be evolved in future. Moreover, this new method would help to evaluate test cases that cover in more scenarios based on the nature of the programming requirement.

Finally, the system can only judge programs written in C++ or Java language. I suggest an enhancement which integrates updated programming language judging method so that it can accept and judge programs written in other programming languages, for instance: C#, Python, Perl... Those are becoming more significant in IT industry.

REFERENCES

[**1**] Andy Kurnia, Andrew Lima, Brenda Cheang. Online Judge, Computers & Education 36 (2011) 299–315, department of Computer Science, National University of Singapore.

[**2**] Blackboard: http://www.blackboard.com/

[**3**] Moodle Project: http://moodle.org/

[**4**] Cheang, B., Kurnia, A., Lim, A.: Online judge. Computers and Education 36, 299–315 (2001)

[**5**] Cheang, B., Kurnia, A., Lim, A., Oon, W.-C.: On automated grading of programming assignments in an academic institution. Computers and Education 41, 121–131 (2003).

[**6**] PC^2 Kit Programming Judge System, California State University, Sacramento's: http://pc2.ecs.csus.edu/

[**7**] TopCoder Programming page: http://www.topcoder.com/

[**8**] Geetha Devasena M. S, Valarmathi M. L, Structural Test Suite Generation using Adaptive Particle Swarm Optimization [APSO]. European Journal of Scientific Research ISSN 1450-216X Vol.75 No.3 (2012), pp. 361-369.

[**9**] Elfira Nureza Ardina. Introduction To Object-Oriented System Analysis And Design With The UML, Version 2.0.
(http://www.ekohandoyo.blog.undip.ac.id/category/apbo/page/3/).

[**10**] Katerina Georgouli1, Pedro Guerreiro. Incorporating an Automatic Judge into Blended Learning Programming Activities. Advances in Web-Based Learning – ICWL 2010 Lecture Notes in Computer Science Volume 6483, 2010, pp 81-90.

[11] Andreas Windisch, Stefan Wappler, Joachim Wegener. Applying Particle Swarm Optimization to Software Testing. Proceedings of the 9th annual conference on Genetic and evolutionary computation. Pages 1121-1128 ACM New York, NY, USA ©2007

[12] José Luis Fernández Alemán, Automated Assessment in a Programming Tools Course. Dept. of Inf. & Syst., Univ. of Murcia, Murcia, Spain. Volume: 54 , Issue: 4 Page(s): 576- 581. Nov-2011.

APPENDICES

APPENDIX A: National ACM ICPC Programming Contest (2011-2012) Questions.

A-2011: SUM OF ODD SEQUENCES

Given an odd integer N, calculate the sum of all the odd integers between 1 and N inclusive.

Input

First line of the input contains T, the number of test cases. Each test case contains a single integer N. N is between 1 and 100.

Output

For each test case output the value 1+3+….+N.

Sample Input	Sample Output
4 1 3 5 15	1 4 9 64

B-2011: SPIRAL

Consider all positive integers written in the following manner (you can imagine an infinite spiral).

21 22 23 24 25 26

20 7 8 9 10 ...

19 6 1 2 11 ...

18 5 4 3 12 ...

17 16 15 14 13 ...

Your task is to determine the position (row,column) of a given number N, assuming that the center (number 1) has position (0,0). Rows are numbered from top to bottom, columns are numbered from left to right (for example, number 3 is at (1,1). Your program should output a string containing the position of N in the form (R,C) where R is the row and C is the column. R and C must not contain any leading zeroes.

Input

The first line of the input gives an integer T, which is the number of test cases. Each test case contains an integer N

($1 \leq N < 2^{31}$).

Output

For each test case, output the position as described above. See sample output for further clarification.

Sample Input	Sample Output
7	(0,1)
2	(1,1)
3	(-1,-1)
7	(2,-2)
17	(-2,1)
24	(-14,3)
830	(-437,221)

765409	

C-2011: SURFACE AREA

The solid in the picture below is made up of 1x1x1 cubes in a 3D grid. In this problem, we'll limit ourselves to solids that are made up of columns rooted on the ground (a column consists of one or several 1x1x1 cubes stacked on top of each other). Such solids can be described as a matrix of digits, where each digit corresponds to the height of a column in the 2D grid that makes up the ground. A zero means there is no column at all in that position.

The volume of such a solid is simple enough to calculate, but what we're interested here in the total surface area including the floor (that is, the number of 1x1 "squares" non - hidden on the outer surface). You are given the information of the solid as a matrix. Your task is to compute the surface area of the given solid. You can assume that the solid is always connected, i.e the columns will be attached to each other in the four cardinal directions.

Input

First line of the input contains T, the number of test cases. Each test case starts with a line containing R and C denoting the number of rows and columns of the solid. Each of the next R lines contains C digits. Each digit are between 0 to 9 inclusive. R and C will be between 1 and 50 inclusive.

Output

For each test case, output the total surface area of the given solid, including the floor area.

Sample Input	Sample Output
4 1 2 11 3 4 4231 2101 0001 3 3 111 101 111 1 1 5	10 54 32 22

G-2011: BI-COLORING

Given a graph determine how many ways you can color the graph with at most two colors. There cannot be an edge containing two vertices of the same color.

Input

First line of the input contains T the number of test cases. Each test case starts with a line containing two integers V($1 \le V \le 30$) and E($0 \le E \le 1000$). Each of the next E line contains two integers a, b ($0 \le a \le V-1$, $0 \le b \le V-1$) denoting that there is a bidirectional edge between a and b. There will not be any self-loop or duplicate edges. The last line of the input will be a blank.

Output

For each test case, output the number of ways you can color the graph with only two colors. If you cannot color the graph with at most two colors output -1.

Sample Input	Sample Output
2 5 5 0 1 0 4 1 2 2 3 3 0 3 3 0 1 1 2 2 0	2 -1

J-2011: QURBAN 2011

Qurban is an Islamic practice that involves the sacrifice of certain livestock. Qurban is done only on the 10-13 Zulhijjah, which is the last month in the Islamic calendar. Livestock to be slaughtered (and sacrificed) are limited to camels, cattle, buffaloes, sheep or goats. The meat from each slaughtered camel, cattle, or buffalo can be divided into 7 equal parts to be shared by 7 persons. Since goats and sheep are small in size compared to the other livestock, therefore one goat or sheep is equivalent to one part of the previous three previous livestock. A person can register to sacrifice more than one part. In order to get the correct number of livestock to be slaughtered, one has to ensure that all 7 parts is accounted for in order to slaughter one buffalo. If not, there will be a cost liability to the person in charged. In a situation when a person registers for more than 7 parts, for example 8 parts of a buffalo, the remainder of one part has to be combined with parts sacrificed by other persons.

Normally, the buffaloes and sheep are supplied by farmer, and the location for slaughtering is done at different places depending on the customers. There is only one truck that can transport either the buffaloes or the sheep. The truck can transport a maximum of 3 buffaloes or 6 sheep at one time. Given a list of orders for qurban, calculate the minimum number of trip to transport the Buffaloes and sheep to be slaughtered.

Input

The input consists of a few lines of data. The first line is an integer T, which represents the number of test data. It is followed by T lines of data. Each of these lines begins with an integer m, which represents the number of order, followed by a single space and a set of orders. The orders are in the following format: two characters where the first is either character 'b' (represents parts of a buffalo) or 's' (represents a sheep). This is followed by an integer that represents the number of orders for the respective parts of cow or sheep. Each order is separated by a single space.

Output

The output is the minimum number of trip to transport the buffaloes and sheep to be slaughtered.

Sample Input	Sample Output
3 4 b3 s1 b9 b4 2 s3 b7 5 b1 b1 b2 b2 b1	2 2 1

B-2012: Longest Balanced Sub-Sequence

We call a sequence of numbers balanced, if the amount of positive values in the sequence equals the amount of negative values.

Input

The first line of the input contains an integer T, $1 \leq T \leq 15$, indicating the number of test cases. For each test case, two lines appear. The first line contains an integer N, $0 \leq N \leq 100000$. The second line contains N non-zero 32-bit signed integers.

Output

For each test case, the output contains a line in the format Case #x: M, where x is the case number (starting from 1) and M is the length of the Longest Balanced Sub-Sequence (LBSS) of consecutive elements.

Sample Input	Sample Output
2 8 -1 -5 1 -7 8 -6 -9 -2 17 5 -2 1 3 7 9 -9 -1 6 -7 -1 2 8 3 1 -2 -1	Case #1: 4 Case #2: 12

Explanation

For the first test case, the LBSS is sub-sequence (-5 1 -7 8) or (1 -7 8 -6); both with two positive and two negative values.

For the second test case, the LBSS is sub-sequence (9 -9 -1 6 -7 -1 2 8 3 1 -2 -1) with six positive and six negative values.

C-2012: CHARITY BOOTH RENTAL

The Kajang Municipal Council is organizing a fundraising event to raise money for charity. The council plans to provide booth spaces for rent to government agencies (GA), private companies (PC) and the general public (GP) who wish to participate in that event. The organizer has decided to allocate booth spaces as follows:

0 < booth spaces for GP < booth spaces for GA < booth spaces for PC

As an example, for 10 booth spaces, the following allocations are possible

GP =1, GA =2, PC =7

GP =1, GA =3, PC =6

GP =1, GA =4, PC =5

GP =2, GA =3, PC =5

Write a program to calculate the number of possible allocations given the number of booth spaces.

Input

Each case is the number of booth spaces T where 6 <= T <= 1000000. The input is terminated with a 0.

Output

For each test case, the output contains a line in the format Case #x: M, where x is the case number (starting from 1) and M is the number of booth spaces and the answer with a colon (':') separating them. The output for each case must be printed on a separate line.

Sample Input	Sample Output
8 10 0	Case #1: 8:2 Case #2: 10:4

E-2012: FIND THE MARBLES

A group of friends are playing "Find the Marbles" game. A handful of marbles are thrown on the ground. The position of the marbles is represented by integer coordinates in a two-dimensional space. The players can only walk on a straight line to find the marbles. The person who can find the highest number of marbles on a straight line wins the game. You are to help them find the number of marbles that the winner found.

Input

The first line of the input is a single positive integer T indicating the number of test cases. Each case starts with the number of marbles N, where 1 < N < 100. On each of the following N lines there are a pair of integers separated by a blank that represents the coordinate in a two-dimensional space. No pair will occur twice.

Output

For each test case, the output contains a line in the format Case #x: M, where x is the case number (starting from 1) and M is an integer representing the largest number of marbles that lie on one line.

Sample Input	Sample Output
2 5 1 1 2 2 3 3 5 10 6 11 3 4 5 6 7 8 8	Case #1: 3 Case #2: 2

For question **E & F**, user can refer to question paper by accessing this URL:
http://fskm.uitm.edu.my/acmicpc/alkhawarizmi/

APPENDIX B: Solution for B-2012: Longest Balanced Sub-Sequence

Solution (C++ code):

```
#include<iostream>
#include<vector>
#include<map>
using namespace std;
int main(){
   int T, N, max;
   cin>>T;
   for(int i=0;i<T;i++){
         cin>>N;
         max=0;
         vector<int> v(N);
         vector<int> sumleft(N);
         map<int, int> m;
         int sum=0;
         for(int ii=0;ii<N;ii++){
            cin>>v[ii];
            if(v[ii]>0) v[ii]=1;
            else v[ii]=-1;
```

```
            sum+=v[ii];
            sumleft[ii]=sum;
            if(sumleft[ii]==0) max=ii+1;
        }
        for(int ii=0;ii<N;ii++){
            if(m[sumleft[ii]]==0) m[sumleft[ii]]=ii+1;
            else if(max<ii+1-m[sumleft[ii]]) max=ii+1-m[sumleft[ii]];
        }
        cout<<"Case #"<<i+1<<": "<<max<<endl;
    }
    return 0;
}
```