



Faculty of Computers and Information

Programming Competitions Training Guide



CPPHamza team 2007 Winner



S2++; team 2009 Winner

By [Mostafa Saad](#)

WORK SMART NOT HARD.
OPTIMIZE YOUR METHODOLOGIES FOR
UNDERSTANDING, THINKING, CODING AND DEBUGGING.

14/3/2013

Table of content

| | |
|---------------------------------------|-----------|
| TABLE OF CONTENT | 2 |
| INTRODUCTION | 3 |
| POPULAR ABBREVIATIONS | 4 |
| PROBLEM EXAMPLE | 4 |
| UVA & LIVE ARCHIVE & SPOJ | 4 |
| <i>Let's solve a problem.....</i> | 5 |
| TOPCODER | 12 |
| <i>Plugins.....</i> | 25 |
| HOW THE JUDGE WORKS..... | 26 |
| WATCH OUT | 28 |
| <i>Blank lines</i> | 28 |
| <i>Extra Spaces</i> | 29 |
| <i>Compiler Errors.....</i> | 29 |
| HOW TO CODE/TEST LOCALLY | 29 |
| ECLIPSE..... | 30 |
| INSTALLATION | 30 |
| PROJECT CREATION | 31 |
| DEBUGGING..... | 33 |
| <i>Trick 1:.....</i> | 37 |
| <i>Trick 2:.....</i> | 38 |
| <i>Trick 3:.....</i> | 39 |
| <i>Last tip:</i> | 39 |
| HOW TO GET HELP | 39 |
| UVAFORUM | 40 |
| TOPCODER FORUM | 40 |
| SPOJ FORUM | 40 |
| ONLINE RESOURCES | 40 |

Introduction

In this document, we introduce several programming competitions environments so that you could start, read, solve and test solutions for programming problems.

Contests: **Online** (e.g. topcoder SRMs, UVA contests) vs **Onsite** (IOI, ACM-ICPC, topcoder TCO finals)

Contests: **Individual** (e.g. IOI, topcoder SRMs) vs **Teams** (e.g. ACM-ICPC)

[**ACM ICPC**](#) is a programming contest holds annually. It is up to you know more about it. Here are some general points

- The Association for Computing Machinery (ACM) was founded in 1947 as the world's first scientific and educational computing society.
- International Collegiate Programming Contest (ICPC) is the oldest, largest, and most prestigious programming contest in the world.
- The ACM ICPC provides college students with opportunities to interact with students from other universities and to sharpen and demonstrate their **problem-solving**, **programming**, and **teamwork** skills.
- The contest provides a platform for ACM, industry, and academia to encourage and focus public attention on the next generation of **computing professionals** as they pursue excellence.

- It is a competition
 - Competition = Teams
 - Team = 3 members + 1 PC + **8-12** problems + 5 hours
- To be **successful**?!!
 - **Programming** skills (Coding & Debugging)
 - Organized **thinking**
 - Algorithms **knowledge** and **experience**
 - Contest solving **strategy**
 - Team management (PC & Members)
 - Time management

Popular Abbreviations

When you **submit** a solution, we express judge answers as following:

AC → Accepted (your code passed)

PE → Accepted, but you printed extra spaces or blank lines

WA → Wrong answer (may be idea is wrong, may be right and code has bugs)

TLE → Time Limit exceeded (may be you have infinite loop, or ur code take really much time)

MLE → Memory Limit exceeded

RTE → Run time error (Index out of boundary or stack overflow (recursion)....)

CE → Compiler Error (Even it may compile in ur machine, judge may have different compiler)

Problem Example

E.g. A problem may be:

"Read 2 numbers less than 1000 and display their sum and multiplication,

Sample Input

12 8

Sample Output

20"

In C++, You code should be like

```
#include<iostream>
Using namespace std;
Int main() {
    Int a, b;
    Cin>>a>>b;
    Cout<<a+b<<"\n";
    Return 0;
}
```

We depend on some of the following Judges; **UVA**, **Live Archive**, **SPOJ**, **TopCoder** and **USACO**. Each one has its own way to **read** the problem and **submit** your code.

UVA & Live Archive & SPOJ

They are similar. You simply first create an account like what you do at any site.

Their links are:

<http://www.spoj.pl/>

<http://uva.onlinejudge.org/>

<https://icpcarchive.ecs.baylor.edu/>

Let say you have created your account on SPOJ. Now we will do following steps.

- 1) Go to the problem you would like to solve
- 2) **Read** the problem and understand it
- 3) **Code** The Problem on your machine
- 4) Back to the problem page and click submit
- 5) Select the language you coded with it (E.g. C++)
- 6) Copy code from your machine, and paste it in the text area, click submit.
- 7) Check your results

Let's solve a problem

Following will be some snapshots. Let's solve a problem called **Test** on SPOJ, its link is:

<https://www.spoj.pl/problems/TEST/>

- Problem TEST - Mozilla Firefox

Bookmarks Tools Help

(S) https://www.spoj.pl/problems/TEST/ 

Latest Headlines

Search Web Answers Mail Bookmarks Local Sign Out

Sphere online judge

Log Out mostafa_saad

my account tutorials

status submit problems search

news contests ranks

forum comments

Server time:
2009-06-06
00 : 01 : 55

Submit My submissions All submissions Best solutions PS PDF Back to list

 SPOJ Problem Set (classical)

1. Life, the Universe, and Everything

Problem code: TEST

Your program is to use the brute-force approach in order to *find the Answer to Life, the Universe, and Everything*. More precisely... rewrite small numbers from input to output. Stop processing input after reading in the number 42. All numbers at input are integers of one or two digits.

Example

Input:
1
2
88
42
99

Output:
1
2
88

Added by: Michał Małafiejski
Date: 2004-05-01
Time limit: 3s
Source limit: 50000B

Figure 1 : Open & Read the problem & understands it

The screenshot shows the Eclipse C/C++ Platform interface. The title bar reads "C/C++ - SPOJ/SPOJ_TEST.cpp - Eclipse Platform". The menu bar includes File, Edit, Refactor, Navigate, Search, Run, Project, Window, Help. The toolbar has various icons for file operations like Open, Save, Cut, Copy, Paste, Find, etc. The Project Explorer view on the left shows a hierarchy of projects: ACMICPC, Contests, Dummy, Kasamba, Library, ProjectEuler, RegionalLibrary, SPOJ (selected), TopCoder, and UVA. Inside the SPOJ folder, there are sub-folders Binaries, Includes, Categories, Debug, others, and files SPOJ_TEST.cpp, 1800_CONTEST.cpp_WA, 1800_CONTEST.cpp_WA_, 185CHASE1.cpp_WA, and c.in. The main view shows the code for SPOJ_TEST.cpp:

```
1 #include<iostream>
2 using namespace std;
3
4 int main()
5 {
6     int num;
7     while(cin>>num) {
8         if(num == 42) break;
9         cout<<num<<"\n";
10    }
11
12    return 0;
13 }
14
```

The code editor has syntax highlighting for C/C++ (purple for keywords, blue for comments, red for strings). Below the code editor is the Problems view, which is currently empty. The Console view at the bottom shows the build log:

```
C-Build [SPOJ]
*****
because it uses the "MinGW GCC" ****
tool-chain that is unsupported on this system. *****

*****
Attempting to build... *****

g++ -O0 -g3 -Wall -c -fmessage-length=0 -oSPOJ_TEST.o ...
g++ -oSPOJ.exe SPOJ_TEST.o
Build complete for project SPOJ
Time consumed: 875 ms.
```

Figure 2: Code The Problem on your machine

J) - Submit a solution - Mozilla Firefox

marks Tools Help

(S) https://www.spoj.pl/submit/TEST/

Latest Headlines

Search Web Answers Mail Bookmarks Local Sign Out

Sphere online judge

Log Out mostafa_saad

my account tutorials

status submit problems search

news contests ranks

forum comments

Server time:
2009-06-06
00 : 06 : 42

Please insert your source code or choose a file:

```
#include<iostream>
using namespace std;

int main()
{
    int num;
    while(cin>>num) {
        if(num == 42) break;
        cout<<num<<"\n";
    }
    return 0;
}
```

Browse...

Language: C++ (g++ 4.0.0-8)

Problem code or id: TEST

Send

Figure 3: In problem page and click submit then select language put ur sol

The screenshot shows a Mozilla Firefox browser window displaying the Sphere online judge status page at <https://www.spoj.pl/status/>. The user is logged in as 'mostafa_saad'. The main content is a table titled 'judge status' showing submission details. The table has columns for ID, DATE, USER, PROBLEM, RESULT, TIME, MEM, and LANG. There are 11 rows of data, with row 1 being accepted and rows 2-11 being rejected (wrong answer or time limit exceeded). The last row shows a submission by 'srikanth' which has timed out.

| ID | DATE | USER | PROBLEM | RESULT | TIME | MEM | LANG |
|---------|---------------------|-------------------------|------------------------------------|----------------------|------|------|------|
| 2459928 | 2009-06-06 00:07:21 | Mostafa Saad | Life, the Universe, and Everything | accepted | 0.00 | 2.5M | C++ |
| 2459927 | 2009-06-06 00:07:17 | cpphamza | The Ant | wrong answer | 4.25 | 3.5M | C++ |
| 2459926 | 2009-06-06 00:06:45 | tyson | Java vs C ++ | runtime error (NZEC) | 0.23 | 218M | JAVA |
| 2459921 | 2009-06-06 00:04:51 | cpphamza | The Ant | time limit exceeded | - | 2.5M | C++ |
| 2459918 | 2009-06-06 00:03:29 | tarun singhal | The Next Palindrome | time limit exceeded | - | 3.4M | C++ |
| 2459913 | 2009-06-06 00:01:19 | Kirill Yurchenko (qks1) | Printer Queue | accepted | 0.49 | 3.6M | RUBY |
| 2459906 | 2009-06-05 23:59:25 | tarun singhal | The Next Palindrome | time limit exceeded | - | 3.4M | C++ |
| 2459904 | 2009-06-05 23:58:50 | srikanth | Bitmap | time limit exceeded | - | 44M | C++ |

Figure 4: Check you results, we are accepted

Focus again in the above code, it does **not read from files**. In all Online Judges (except USACO) you **never** read from a file or write to it.

The following are hints that keep programming / testing easier in your machine.

There are 2 issues, **first**, imagine each time you would like to see output of your program, u gives the console your input.

One way to solve this problem is to use files to read & write, and when u finish, you back everything as if it is console. It is really **annoying**. One magic way to solve this problem is **freopen**. Freopen allows you to read from file or write to file although still using cin & cout. It plays its tricks to direct everything to files. Check next code.

```

#include<iostream>
using namespace std;
int main()
{
    freopen("in.txt", "rt", stdin);
    freopen("out.txt", "wt", stdout);

    int num;
    while(cin>>num)
    {
        if(num == 42) break;
        cout<<num<<"\n";
    }

    return 0;
}

```

Test it. This program read from file in.txt & Write to file out.txt. before submit just **comment them**. If you removed 2nd freopen, u will be reading from file & writing on console.

One final annoying thing, you may forget to comment them. In SPOJ & UVA there is a trick. You can send ur codes as it is, and it will work, just add following:

```

#include<iostream>
using namespace std;
int main()
{
#ifndef ONLINE_JUDGE
    freopen("in.txt", "rt", stdin);
    freopen("out.txt", "wt", stdout);
#endif

    int num;
    while(cin>>num)
    {
        if(num == 42) break;
        cout<<num<<"\n";
    }

    return 0;
}

```

Try them & enjoy. The OJ(online judge) will behave as if these lines does not exist.

Final Note: Sometimes you try to open UVA site and it do not. The site has bugs. If happened, please go to your browser and remove any **cookie** related to UVA (E.g. search name UVA and also judge). If you do not know how to remove cookies, please search Google (E.g. how to remove cookies in Firefox)

The screenshot shows the ACM-ICPC Live Archive Online Judge interface. On the left, there's a sidebar with links for 'ACM-ICPC Live Archive', 'Logout', 'Online Judge', 'Quick Submit' (which is highlighted), 'Migrate submissions', 'My Submissions', 'My Statistics', 'Browse Problems', 'Contests', and 'Site Statistics'. Below that is a Google search bar with 'Search' button and 'Web' selected. Under 'Web' is a link to 'uva.onlinejudge.org'. The main area is titled 'Quick Submit' and shows a form for problem 272. The 'Language' dropdown is set to 'C++ 4.1.2 - GNU C++ Compiler with options: -lm -lcrypt -O2 -pipe -ansi -DONLINE_'. The code area contains the following C++ code:

```
#include<iostream>
using namespace std;

int main() {
    bool appear = false;
    string line; // as u see, we read/write from console!
    while(getline(cin, line)) { //as long as read line
        for (int i = 0; i < line.size(); ++i) {
            if(line[i] == '') {
                if(!appear)
                    cout<<"^", appear = true;
                else
                    cout<<"", appear = false;
            } else
                cout<<line[i];
        }
        cout<<"\n";
    }
    return 0;
}
```

Figure 5: Code submitted for UVA problem 272

You can browse for problems in UVA. Use Problem Set Volumes [has first 900 problem], Contest Volumes, others to find your problem. Another nice way is using old external servers:

<http://uva.onlinejudge.org/external/6/673.html>

<http://uva.onlinejudge.org/external/105/10551.html>

just divide problem number by 100 [105 = 10551/100]

You could find **many solutions** on **web** for problems on **UVA**. One of them in organized way on steven halim website: <http://www.comp.nus.edu.sg/~stevenha/programming/acmoj.html>



PROBLEM SET ARCHIVE with ONLINE JUDGE

My UVa Online Judge status:

[533 solved before server migration, currently "only" 249.](#)
(rank [594](#) as of 27/04/09 - will go up again soon, probably top 200), [\(details\)](#).

My UVa Online Judge volume by volume statistics

VOL: AVAILABLE HINTS

[1: 52](#)
[2: 27](#)
[3: 51](#)
[4: 71](#)
[5: 45](#)
[6: 36](#)
[7: 36](#)
[8: 48](#)
[9: 29](#)

-
-
-
-
-
-
-
-

~395 hints so far

VOL: AVAILABLE HINTS

[100: 44](#)
[101: 43](#)
[102: 31](#)
[103: 30](#)
[104: 27](#)
[105: 36](#)
[106: 53](#)
[107: 30](#)
[108: 30](#)
[109: 37](#)
[110: 44](#)
[111: 35](#)
[112: 40](#)
[113: 30](#)
[114: 41](#)
[115: 55](#)
[116: 00](#)

~606 hints so far

Alternative view: [by category \(but not up to date\)](#)

TopCoder

In Topcoder at <http://www.topcoder.com/tc> we use it to create our accounts, but we do not use it to submit our codes, you have to use the arena. Topcoder contest [SRM] is 2 contest in same time classifying people by their skills (**Division 1** and **Division 2**). Each division has 3 problems weighted with their hardness (250, 500, 1000). Sometimes a problem may weight 600 instead of 500 if it is harder.

Let Say you want to read a problem and solve it

- 1) Register on the site: use following **important** points
 - a. **I Want to Compete:** on TopCoder

- b. **Email Notifications:** Mark all of them
 - c. **School:** Choose School. Write **you faculty.** Advice, copy school name from one of your in same university, so that you all classified in same location.
- 2) Setup Java JDK (<http://www.oracle.com/technetwork/java/javase/downloads/index.html>)
 - 3) On http://www.topcoder.com/tc , on the left you will find **algorithms** menu click it
 - 4) You will find **launch arena**, if it does not open, you have problems in java installation at ur machine
 - 5) Enter you user name & password of the registration (READ CREAFULLY WHILE REGISTERATION)
 - 6) To Open a problem: **SRM 271 Div2 250** on arena:
 - 7) Click **Practice rooms**
 - 8) Click **SRMS**
 - 9) Iterate until you see "**SRM 271 Div2**", Click it. Ignore the first number in the list, it is just sequence.
 - 10) Now you will see a menu of 3 problems, each one has score depends on ur time to solve it
 - 11) Click 250
 - 12) Read the problem
 - 13) Code it at ur machine, or in the arena
 - 14) Copy/paste the solution in the bottom area, click compile
 - 15) Test all tests on the arena, if they **all** passed, it is ok to submit
 - 16) Now do **full** testing, after submitting
 - 17) Click Practice room, the run system test.

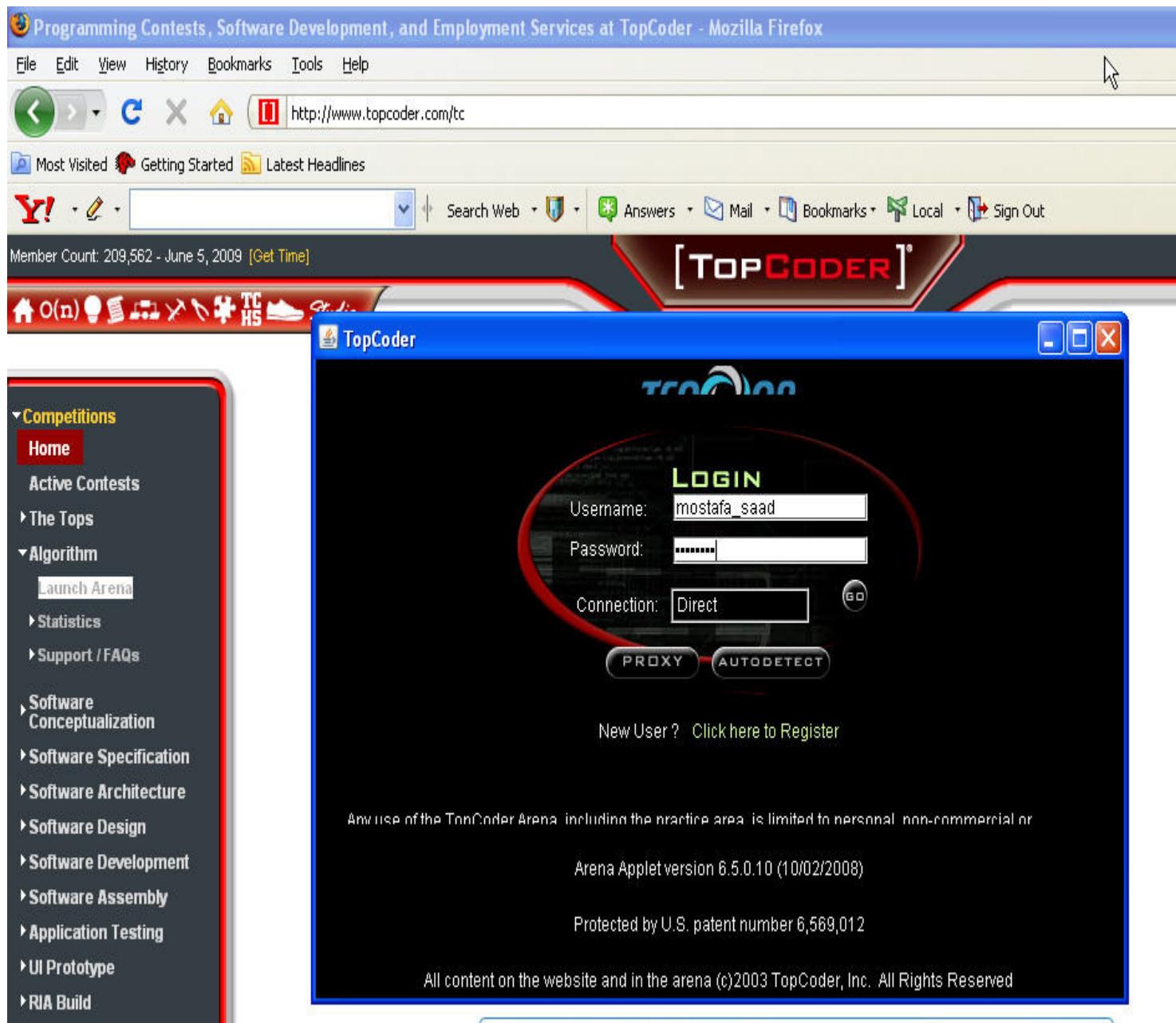


Figure 6: TC login through arena application

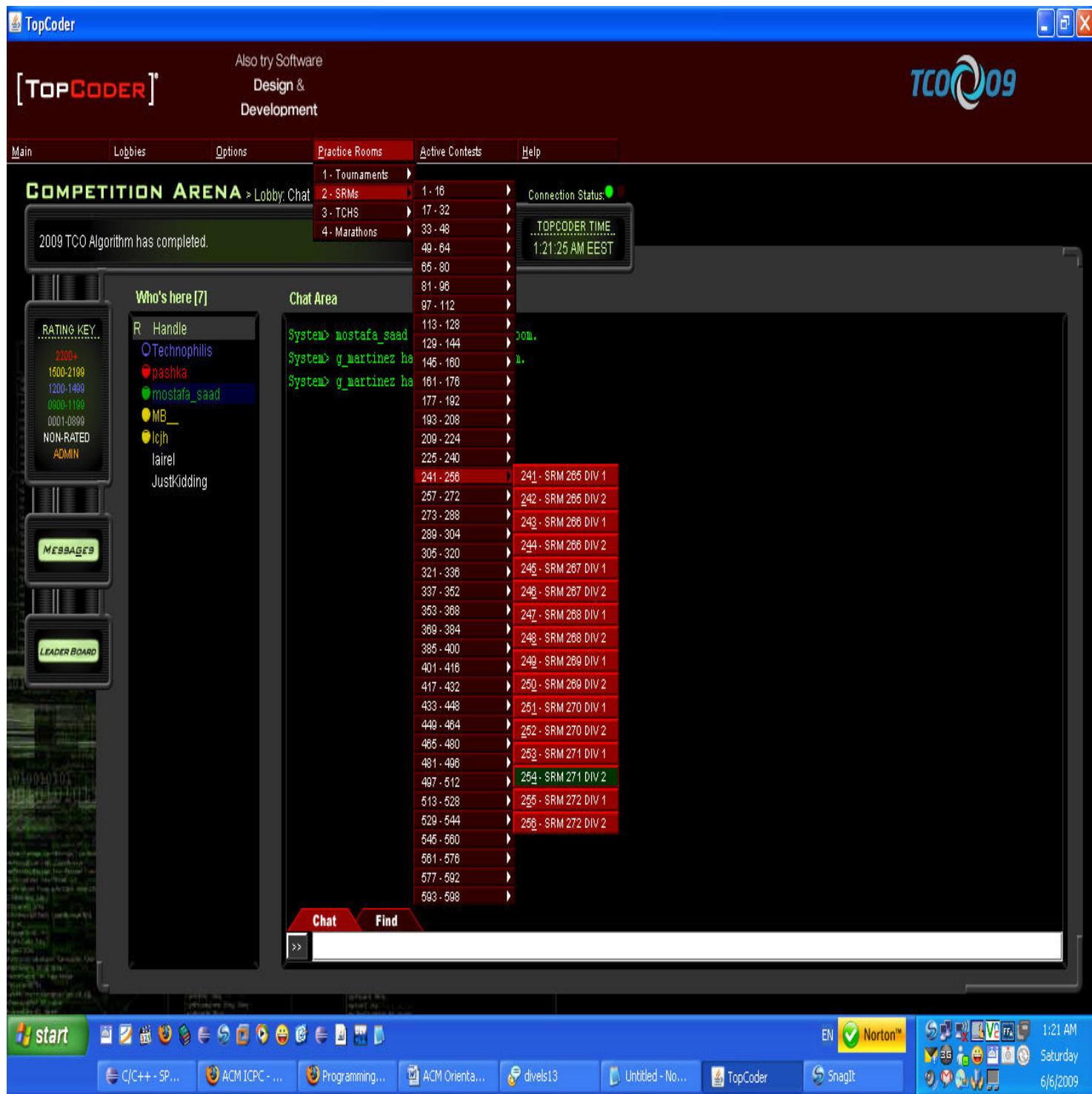


Figure 7: Select problem, Ignore left number, just sequence

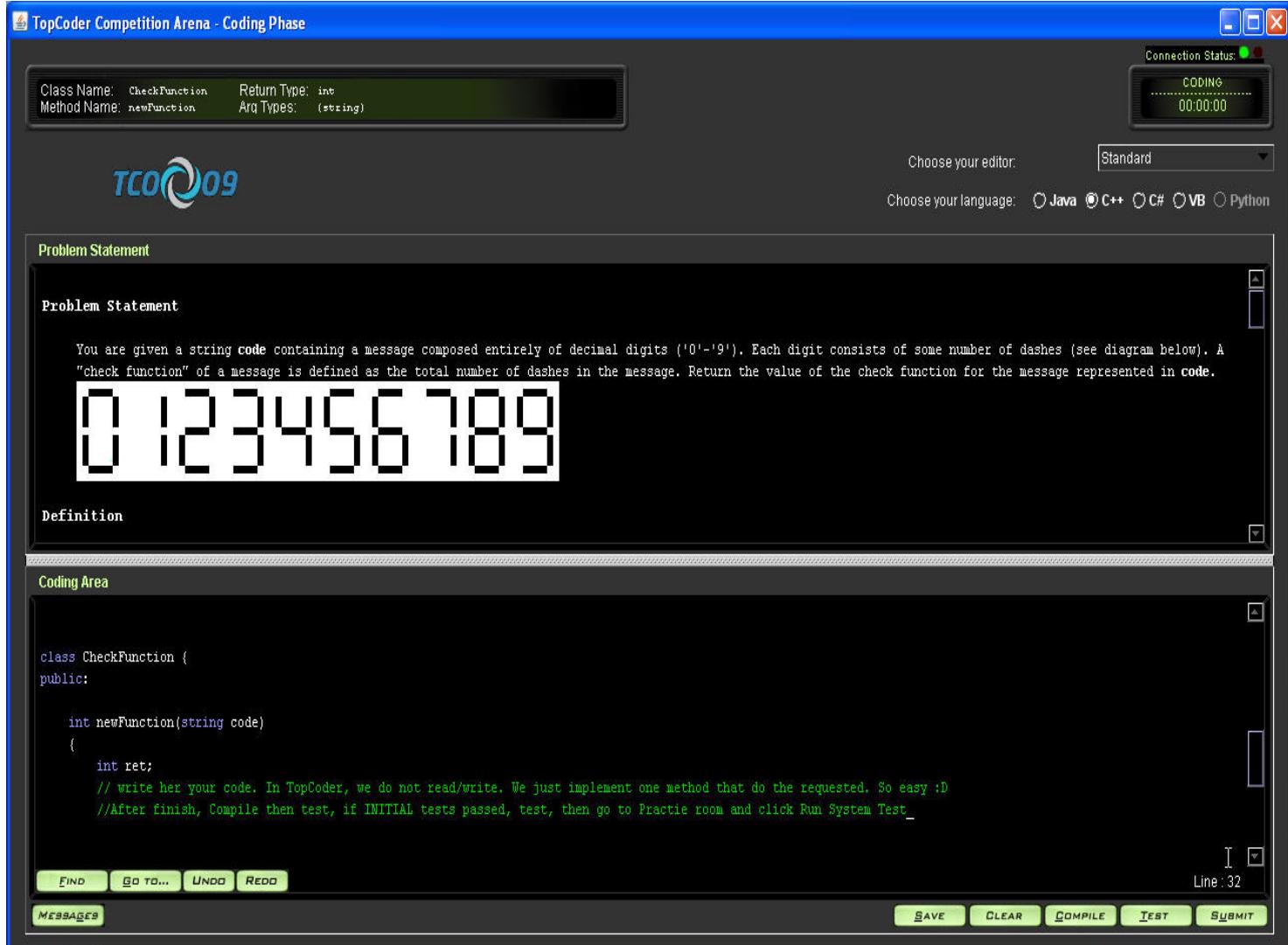


Figure 8: Read body of my code

TopCoder Code is a bit different. You have to submit a **class/structure**, its details as mentioned in the problem. Class name as mentioned and ONE METHOD as mentioned. Check CAREFULLY the image above. Use **exactly** the name of class and name of method (case sensitive)

Another code snippet:

```

//SRM 144 1st problem
//As you see, in TC we do not read & we do not submit main

class Time {
// class name as exactly in problem
public:

    // TC will initialize your function with necessary data
    string whatTime(int seconds) { //method as exactly in problem
        string ret;
        int hr = seconds/3600;
        int min = (seconds%3600)/60;
        seconds = (seconds%3600)%60;
        stringstream s;
        s << hr << ":" << min << ":" << seconds;
        ret = s.str();
        return ret;
    }
};

```

You do not need to know almost anything about classes, think in them as wrapper for your methods. In **testing**, TopCoder is different. To know that you are accepted, Click **Practice room**, then **run system test**. Check next screen:

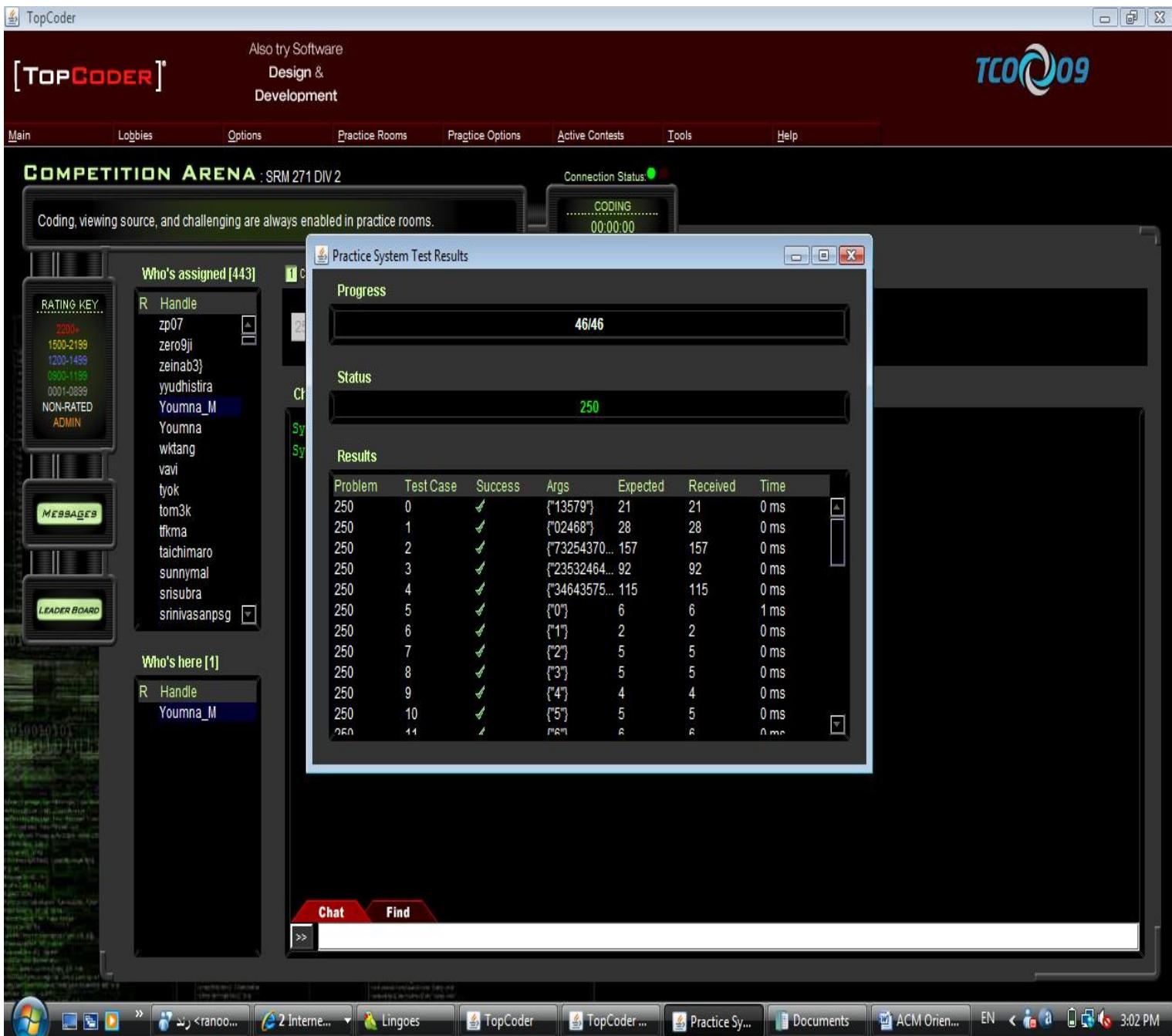


Figure 9: Run system test to check if your code totally correct

As you see, the code **passed 46/46** test cases. In case you did not pass, TopCoder will show you the test case and its details.

Finally, after SRMs, topcoder starts "**Challenge Phase**" where you could see other contestants code and challenge it with a test case. If passed and the code failed you got 50 points, else you lose 25 points.

Finally, this is TC home page, we care with algorithms part. Inspect its **entire links**:

The screenshot shows the TopCoder website in a Mozilla Firefox browser. The title bar reads "Programming Contests, Software Development, and Employment Services at TopCoder - Mozilla Firefox". The menu bar includes File, Edit, View, History, Bookmarks, Yahoo!, Tools, and Help. The toolbar has icons for Back, Forward, Stop, Home, and Search. The address bar shows the URL http://www.topcoder.com/tc. Below the toolbar, there are links for Most Visited, Getting Started, and Latest Headlines. A search bar with "Search Web" and a mail icon is also present. The main content area features the TopCoder logo and a banner for the "REMIX CHALLENGE". On the left, a sidebar menu under "Competitions" lists Home, Active Contests, The Tops, Algorithm, Launch Arena, Statistics, Match Archive, Match Overviews, Match Winners, Match Results, Match Editorials, Problem Archive, Recent Color Changes, Data Feeds, and Support / FAQs. The main content area displays information about SRM 447, which is brought to you by Facebook. It includes a large "SRM 447" text, the date August 25, a Facebook logo, and a link to "Facebook Apply". Below this, there is a section titled "SRM 447: Chat Session with Facebook and \$5,000" featuring a file icon and a brief description: "Don't miss the Facebook chat session in the Competition Arena one hour prior to the competition. Facebook representatives will be present to discuss their exciting career opportunities." It also mentions that Facebook is giving out \$5,000 in prizes and provides a link to learn more.

Figure 10: TC home, inspect Match Editorial, Problem Archive and Match Archive

One of the links is the **editorial**; it is place where you find **solutions description** for SRMs.

The screenshot shows a web browser window for the TopCoder website at apps.topcoder.com/wiki/display/tc/Algorithm+Problem+Set+Analysis. The page title is "Algorithm Problem Set Analysis". The top navigation bar includes links for Facebook, Member Count (468,672 - March 13, 2013), and various TopCoder services like O(n), Studio, and TC HS.

The left sidebar contains a navigation menu:

- Competitions
 - Overview
 - Copilot Opportunities
 - Design
 - Development
 - UI Development
 - QA and Maintenance
 - Algorithm
 - Single Round Matches (SRM)
 - Overview
 - Track Information
 - Statistics
 - Match Archive
 - Match Overviews
 - Match Winners
 - Match Results
 - Match Editorials**
 - Problem Archive
 - Recent Color Changes
 - Data Feeds
 - Launch Arena
 - Marathon Match
 - High School

Open any SRM and check content.

| |
|--------------------------------------|
| Match Winners |
| Match Results |
| Match Editorials |
| Problem Archive |
| Recent Color Changes |
| Data Feeds |
| Launch Arena |
| ► Conceptualization |
| ► Specification |
| ► Architecture |
| ► Component Design |
| ► Component Development |
| ► Assembly |
| ► Test Scenarios |
| ► Test Suites |
| ► UI Prototype |
| ► RIA Build |
| ► Bug Race |
| ► Marathon Match |
| ► High School |
| ► Conilot Opportunities |

In Division 2, wshtb had the fastest time on the hard problem and won the match, with olive in second, and X_X in the third place fifth with the help of the fastest time on the medium problem.

The Problems

CheckFunction

[Rate It](#)

[Discuss it](#)

Used as: Division Two - Level One:

| | |
|------------------------|--|
| Value | 250 |
| Submission Rate | 262 / 277 (94.58%) |
| Success Rate | 261 / 262 (99.62%) |
| High Score | batman7 for 249.30 points (1 mins 30 secs) |
| Average Score | 229.73 (for 261 correct submissions) |

The best way to do problems like this one is to put all the constants from the problem statement into constants in your `dashes` array that contains the number of dashes for each of the digits. Then for each digit in `code` we look into this array and add the value for that digit to our result. The code follows:

```
// a zero has 6 dashes, a one has 2 dashes, ...
int dashes[] = { 6, 2, 5, 5, 4, 5, 6, 3, 7, 6 };

int newFunction(string code) {
    int result = 0;
    for (unsigned int i=0; i<code.size(); i++)
        result += dashes[ code[i]-'0' ];
```

Figure 11: Solution hints for problem check function

What about seeing the submitted solutions during contest? Go to problems archive.

community.topcoder.com/tc?module=MatchList

Facebook

- Competitions
 - Overview
 - Copilot Opportunities
 - Design
 - Development
 - UI Development
 - QA and Maintenance
 - Algorithm
 - Single Round Matches (SRM)
 - Overview
 - Track Information
 - Statistics
 - Match Archive**
 - Match Overviews
 - Match Winners
 - Match Results
 - Match Editorials
 - Problem Archive
 - Recent Color Changes
 - Data Feeds
 - Launch Arena
 - Marathon Match
 - High School

<< prev | next >>

* resubmissions do not increase submission totals or averages

| Algorithm Match Archive | | | | | | |
|-------------------------|------------|-------------|------|-------|-----------|----------|
| Match | Date | Competitors | | | Solutions | |
| | | Total | DivI | DivII | Total | Accepted |
| TCO13 Round 1C | 03.09.2013 | 1826 | 1826 | 1826 | 2496 | 1446 |
| SRM 572 | 03.06.2013 | 1365 | 634 | 731 | 848 | 530 |
| TCO13 Round 1B | 03.02.2013 | 1820 | 1820 | 1820 | 3099 | 1820 |
| TCO13 Round 1A | 02.23.2013 | 1752 | 1752 | 1752 | 2406 | 1406 |
| SRM 571 | 02.19.2013 | 2062 | 769 | 1293 | 1058 | 634 |
| SRM 570 | 02.13.2013 | 1780 | 740 | 1040 | 768 | 468 |
| SRM 569 | 02.06.2013 | 1346 | 567 | 779 | 827 | 487 |
| SRM 568 | 01.29.2013 | 2020 | 723 | 1297 | 692 | 402 |
| SRM 567 | 01.21.2013 | 1646 | 654 | 992 | 904 | 484 |
| SRM 566 | 01.12.2013 | 1905 | 701 | 1204 | 634 | 404 |
| SRM 565 | 12.20.2012 | 1759 | 716 | 1043 | 914 | 484 |
| SRM 564 | 12.12.2012 | 1556 | 694 | 862 | 1013 | 454 |
| SRM 563 | 12.08.2012 | 1193 | 553 | 640 | 702 | 382 |
| SRM 562 | 11.30.2012 | 1343 | 571 | 772 | 589 | 359 |
| SRM 561 | 11.20.2012 | 1344 | 618 | 726 | 541 | 341 |
| SRM 560 | 11.10.2012 | 1666 | 645 | 1021 | 737 | 407 |
| SRM 559 | 10.30.2012 | 869 | 435 | 434 | 383 | 183 |
| SRM 558 | 10.19.2012 | 1582 | 674 | 908 | 601 | 281 |
| SRM 557 | 10.10.2012 | 1872 | 753 | 1119 | 1108 | 484 |

Open any SRM. You will find list of solutions, one for Div1 and other for Div2. To open a solutions of a contestant, click **circle** before his name.

Please select a round:

Single Round Match 567 > Round 1



Discuss it

| Division I Leaders | Score | Placed | Results | Division II Leaders |
|--------------------|---------|--------|---------|---------------------|
| [*] [[iwi]] | 1219.46 | 1 | Room 2 | [*] SegmentFault |
| [*] tom612pl | 1152.04 | 2 | Room 20 | [*] hyy5159 |
| [*] Hujiwara | 1085.81 | 3 | Room 30 | [*] IH19980412 |
| [*] hirosegolf | 1075.31 | 4 | Room 3 | [*] naranbayar_mon |
| [*] tourist | 1067.28 | 5 | Room 1 | [*] kevin00050 |
| [*] cafelier | 879.05 | 6 | Room 31 | [*] sharatiitr |
| [*] Petr | 805.77 | 7 | Room 5 | [*] hnu09dm |
| [*] narri | 776.14 | 8 | Room 10 | [*] wilsonlym |
| [*] gusakov | 770.23 | 9 | Room 20 | [*] cs_Diablo |
| [*] xujie | 717.13 | 10 | Room 3 | [*] noodle1 |

Division I Problem Stats

| | Problem Name | Submissions | Correct % |
|-------------|----------------------|-------------|-----------|
| Level One | TheSquareRootDilemma | 537 | 85.10% |
| Level Two | StringGame | 346 | 49.42% |
| Level Three | Mountains | 21 | 38.10% |

Division II Problem Stats

| | Problem Name | Submissions | Correct % |
|-------------|----------------------|-------------|-----------|
| Level One | NinjaTurtles | 854 | 63.82% |
| Level Two | TheSquareRootDilemma | 372 | 23.12% |
| Level Three | MountainsEasy | 8 | 37.50% |

Viewing top 10 [submit]

Solution of contestant

Click the [+] icon to view problem information.

Statistics for Single Round Match 567 > Round 1 > Room 82

| Coders | Submissions | | Defenses | | Challenges | | System | | Point Total | Ratings | |
|---------------------|-------------|----------|----------|----------|------------|----------|---------|-----------|-------------|-----------|---|
| | Qty | / Points | Qty | / Points | Qty | / Points | Test | Old / New | | Old / New | |
| [+] cs_Diablo | 2 | 621.15 | 1 | 0.00 | 2 | 100.00 | 0.00 | 721.15 | 1112 | 1255 | ▲ |
| [+] noodle1 | 2 | 690.12 | 0 | 0.00 | 6 | 225.00 | -237.62 | 677.50 | 1200 | 1681 | ▲ |
| [+] sos5403 | 1 | 222.05 | 0 | 0.00 | 1 | 50.00 | 0.00 | 272.05 | 764 | 889 | ▲ |
| [+] tduong | 1 | 207.83 | 1 | 0.00 | 0 | 0.00 | 0.00 | 207.83 | 1200 | 1247 | ▲ |
| [+] yashodhandivaka | 2 | 384.18 | 1 | -208.12 | 0 | 0.00 | 0.00 | 176.06 | 635 | 708 | ▲ |
| [+] gemy_eldin | 1 | 160.59 | 1 | 0.00 | 0 | 0.00 | 0.00 | 160.59 | 885 | 899 | ▲ |
| [+] nanook | 1 | 152.92 | 2 | 0.00 | 0 | 0.00 | 0.00 | 152.92 | 1200 | 1128 | ▼ |
| [+] code.freak | 2 | 620.29 | 2 | -446.03 | 1 | -25.00 | 0.00 | 149.26 | 1200 | 1119 | ▼ |
| [+] raki_23 | 2 | 438.62 | 2 | -438.62 | 9 | 75.00 | 0.00 | 75.00 | 490 | 535 | ▲ |
| [+] apusing | 1 | 157.44 | 1 | -157.44 | 1 | 50.00 | 0.00 | 50.00 | 811 | 788 | ▼ |
| [+] Chan04 | 2 | 635.99 | 2 | -442.15 | 1 | 50.00 | -193.84 | 50.00 | 549 | 585 | ▲ |
| [+] Brid | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0.00 | 1200 | 885 | ▼ |
| [+] albelaa | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0.00 | 1200 | 885 | ▼ |
| [+] alyessam | 2 | 396.20 | 2 | -396.20 | 0 | 0.00 | 0.00 | 0.00 | 1200 | 885 | ▼ |
| [+] nnesterov | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0.00 | 1200 | 885 | ▼ |
| [+] batbaatar | 2 | 337.63 | 2 | -337.63 | 0 | 0.00 | 0.00 | 0.00 | 696 | 660 | ▼ |
| [+] tarashor | 0 | 0.00 | 0 | 0.00 | 0 | 0.00 | 0.00 | 0.00 | 700 | 654 | ▼ |
| [+] wipkirill | 2 | 529.59 | 3 | -529.59 | 0 | 0.00 | 0.00 | 0.00 | 692 | 628 | ▼ |
| [+] Minail | 2 | 409.85 | 2 | -409.85 | 1 | -25.00 | 0.00 | -25.00 | 1200 | 602 | ▼ |

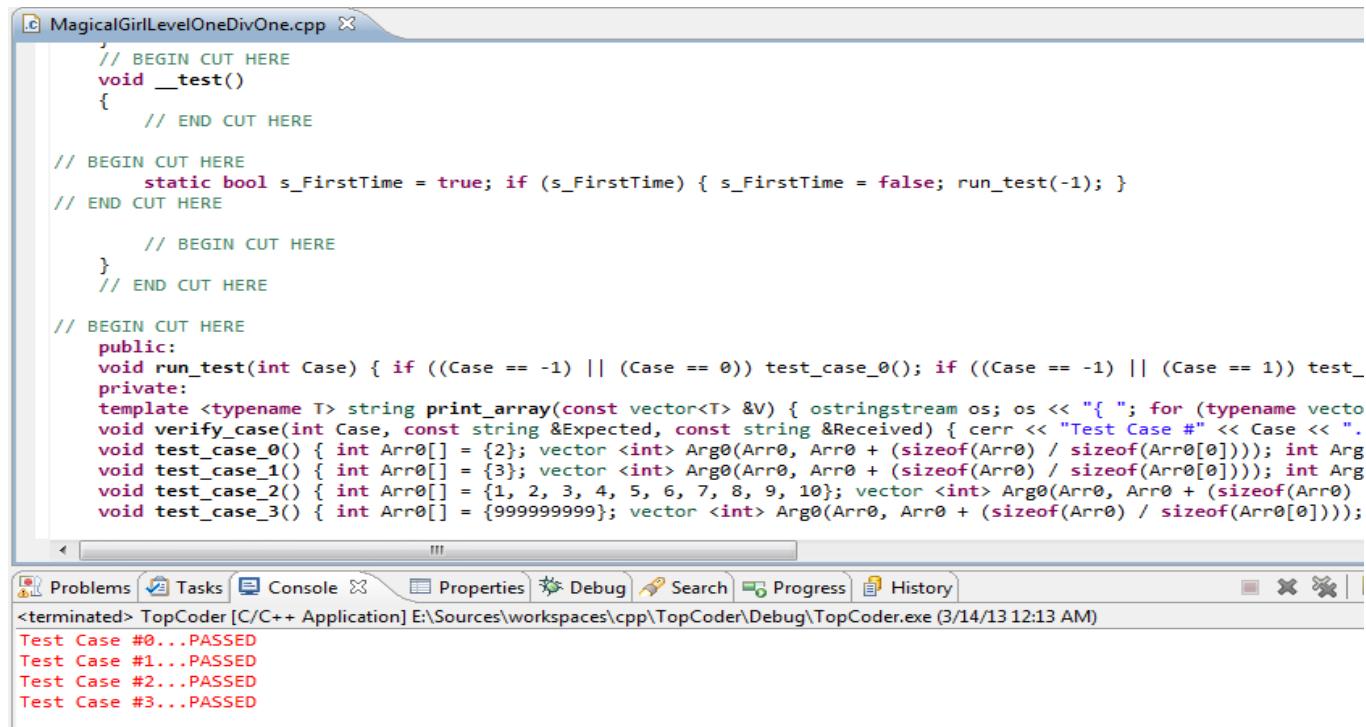
Problem Information for noodle1

Problems

| Class Name | Method Name | Difficulty | Coding Time | Status | Points |
|----------------------|-----------------|-------------|-------------|--------------------|--------|
| NinjaTurtles | countOpponents | Level One | 0:06:32.678 | Failed System Test | 237.62 |
| TheSquareRootDilemma | countPairs | Level Two | 0:09:23.870 | Passed System Test | 452.50 |
| MountainsEasy | countPlacements | Level Three | 0:57:53.339 | Compiled | 0.00 |

Plugins

Most topcoders use plugins. Through it, we code away of arena or using it but in a more perfect way. This avoid u **manual testing** and helps in **connection problems**.



The screenshot shows the Eclipse IDE interface with a C++ plugin for TopCoder. The code editor window displays a file named "MagicalGirlLevelOneDivOne.cpp" containing C++ code with various sections commented out as "CUT HERE". Below the code editor is the Eclipse toolbar with icons for Problems, Tasks, Console, Properties, Debug, Search, Progress, and History. The status bar at the bottom shows the message "<terminated> TopCoder [C/C++ Application] E:\Sources\workspaces\cpp\TopCoder\Debug\TopCoder.exe (3/14/13 12:13 AM)". The console output window shows four test cases all passed: "Test Case #0...PASSED", "Test Case #1...PASSED", "Test Case #2...PASSED", and "Test Case #3...PASSED".

```
// BEGIN CUT HERE
void __test()
{
    // END CUT HERE

// BEGIN CUT HERE
    static bool s_FirstTime = true; if (s_FirstTime) { s_FirstTime = false; run_test(-1); }
// END CUT HERE

    // BEGIN CUT HERE
}
// END CUT HERE

// BEGIN CUT HERE
public:
void run_test(int Case) { if ((Case == -1) || (Case == 0)) test_case_0(); if ((Case == -1) || (Case == 1)) test_
private:
template <typename T> string print_array(const vector<T> &V) { ostringstream os; os << "{ "; for (typename vecto
void verify_case(int Case, const string &Expected, const string &Received) { cerr << "Test Case #" << Case << "."
void test_case_0() { int Arr0[] = {2}; vector <int> Arg0(Arr0, Arr0 + (sizeof(Arr0) / sizeof(Arr0[0]))); int Arg
void test_case_1() { int Arr0[] = {3}; vector <int> Arg0(Arr0, Arr0 + (sizeof(Arr0) / sizeof(Arr0[0]))); int Arg
void test_case_2() { int Arr0[] = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}; vector <int> Arg0(Arr0, Arr0 + (sizeof(Arr0)
void test_case_3() { int Arr0[] = {999999999}; vector <int> Arg0(Arr0, Arr0 + (sizeof(Arr0) / sizeof(Arr0[0])));
```

One easy way is **CodeProcessor**. See here guideliens:

<http://apps.topcoder.com/wiki/display/tc/How+to+install+The+Arena+plug-ins>

KawigiEdit is another option. Check how to use it from her:

<http://community.topcoder.com/contest/classes/KawigiEdit/KawigiEdit.html>

An **update** for this plugin is done by **ahmed aly**. Target of plugin is to remove any unused code to avoid the unused code rule. Check: <http://apps.topcoder.com/forums/?module=Thread&threadID=697303>

How the Judge Works

NOTE: FOLLOWING IS NOT ACCURATE; IT IS JUST TO PUT YOU INTO PICTURE.

The online judge receives your code, then put it in a file (E.g. main.cpp) and compile it (E.g. main.exe). Each problem has input file and output file. They are not like the ones in the problem statement, they are much **bigger**. After that, it gives your code the input file in a console style and records your answer in a file. Then it compares it with the output file. If your output file similar, then accepted, else wrong. If you needed much time or memory to find solution, you will get TLE or TME. Do not forget, we do not write our codes using files, **CONSOLE**

Let say the problem is " read N, number of test cases, then read N lines, each line has 2 numbers, print their sum

Sample Input

3

10 10

2 8

1001 -1

Sample Output

20

10

1000

"

Now, Let Say, you wrote following code:

```
#include<iostream>
using namespace std;

int main() {
    int a, b, cases;
    cin>>cases;
```

```
for(int I = 0 ; i<cases;i++) {  
    cin>>a>>b;  
    cout<<a+b<<"\n";  
}  
return 0;  
}
```

Let Say, judge files were:

Judge Input

5

10 10

2 8

1001 -1

0 0

-100 -100

Judge Output

20

10

1000

0

-200

Now, Judge will run your code and get your output file, and compare it with your output file. If it matches, you win.

WARNING: Never to print any **extra** data or statements, FOLLOW the specs.

What do u think next code will do?

```
Int main() {
```

```
    Int a, b, cases;
```

```
    Cin>>cases;
```

```
For(int I = 0 ; i<cases;i++) {
```

```
Cout<<"Please Enter 2 numbers: ";
```

```
Cin>>a>>b;
```

```
Cout<<a+b<<"\n";
```

```
}
```

```
Return 0;
```

```
}
```

Can you guess how will be your output file? Do you think you will pass the judge file??!

Watch out

Blank lines

Most of problems ask you to answer many test cases, so you have to read whole input and answer it. Usually the data of each test case separated in one of 3 ways (relative to **blank lines**)

Let say we are solving factorial problem, represent +--+ (means blank/empty line) just to be clear

- 1) Print factorials of numbers between 1-3, **print each test case answer on line**

```
1  
2  
3  
+--+
```

- 2) Print factorials of numbers between 1-3, **print blank line after each test case** answer

```
1  
+--+  
2  
+--+  
3  
+--+  
+--+
```

- 3) Print factorials of numbers between 1-3, print blank **line between consecutive test cases** answer

```
1  
+--+  
2
```

```
++++  
3  
++++
```

You must print exact blank lines, or you will get Wrong answer.

Extra Spaces

What if problems ask you to print numbers from 1 to 5. Print space after each number.

What if problems ask you to print numbers from 1 to 5. Print space between each number.

Let say space is * (for clarification)

Answer 1: 1*2*3*4*5*

Answer 2: 1*2*3*4*5

Compiler Errors

Some times in judges, you may get compiler error because:

- 1- Your code is already has compile issues or your machine
- 2- You selected wrong compiler (E.g. C or Java instead of C++)
- 3- You are using different compiler **version** or type
 - a. Sometimes even you r coding in C++, and do not have compiler errors and selecting C++, u got CE, this may happen as there are different versions. Check reasons on site. Sometimes you need to **add** includes (E.g. #include<string>; sometimes you are using **global** variables used by these compilers.

How to code/test locally

1. Use Nice and effective IDEs to code (E.g. Eclipse or visual), DEV is not the best at all.
2. Always use **files** to read from. Many times our mistake due to wrong initialization and un-clearing for data. E.g. someone if tried test cases in wrong order may see different errors and catch some bugs.
3. Take your test cases and duplicate them.
4. Each problem usually has boundary test cases. Boundaries are either the too small ones or too big ones. Testing the too small may help you catch the WAs and testing the too large may help you in catching the RE, TLE and MLE. E.g. If you are reading a line of string of max length 10. Test String lengths 0, 1, and 10.

5. You also should learn **debugging**, so that you may iterate on your code to check code behavior.
Using ***print line method*** most of times is the wrong, slowest way.

Eclipse

Eclipse is a nice and strong tool that can be used during programming. It has many nice features, like formatting code, debugging, hot fixing and others. Following is a guide for people who are interested in trying it. Why do you it is named that name?!!!

Installation

1. Download Dev C++
2. Install it
3. Download Eclipse IDE for C/C++ Developers from <http://www.eclipse.org/downloads/>
4. Unzip and open it.
 - 4.1 Use path for workspace without **SPACES**.
 - 4.2 Use to create ur workspace away from C directory, so that when u remove windows, ur code remain.
5. Go to window, **preferences** and Check next Pic
6. **Add variable path**, with Dev installation location
 - 6.1 Follow the picture

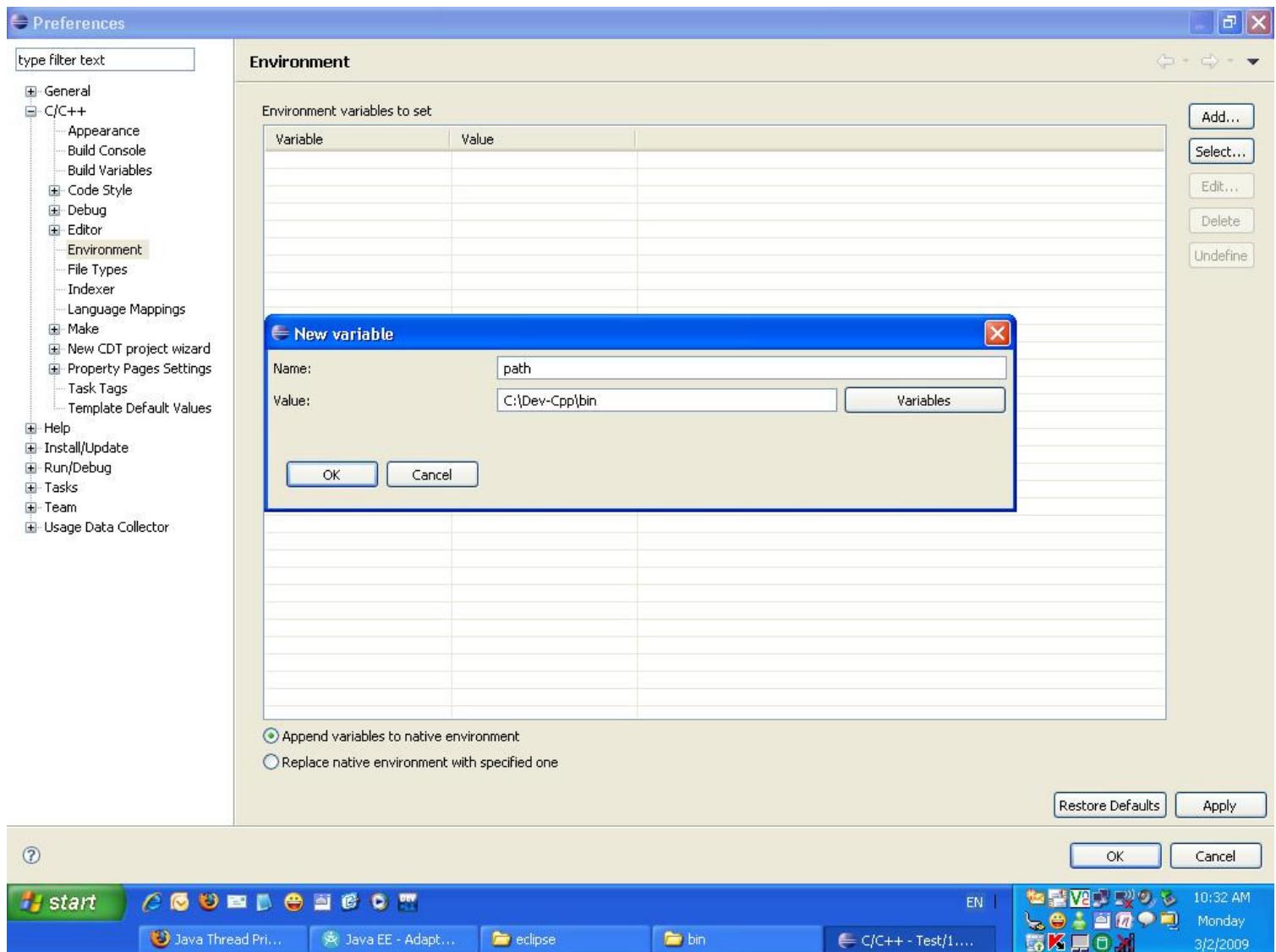


Figure 12: Eclipse Environment

Project Creation

1. Follow Next to create project
2. Now u can start.
3. Right click on ur project, select new source file
4. Name it something.cpp (E.g. 1.cpp)
5. Start to write ur code, compile and run

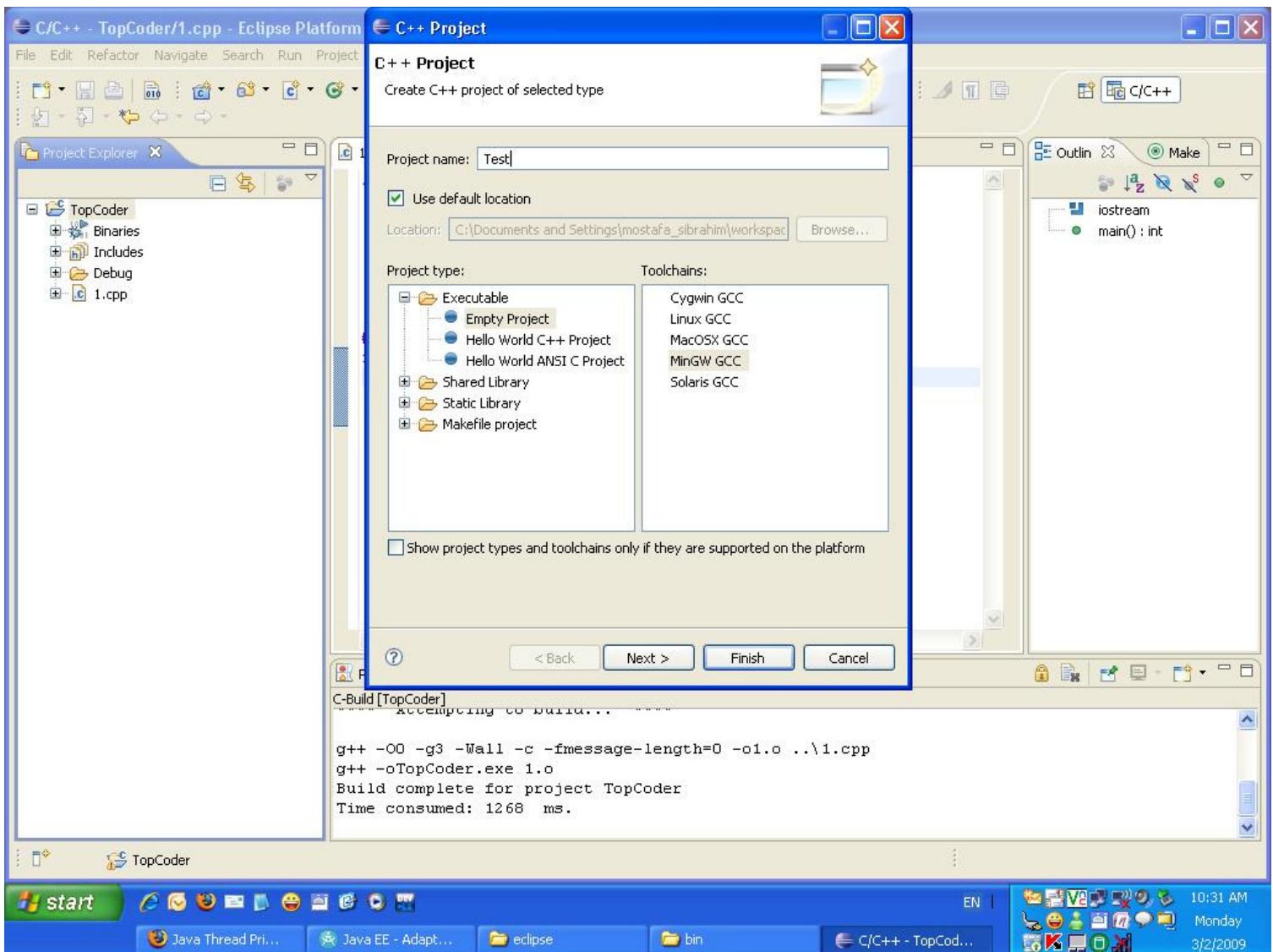


Figure 13: Eclipse Create Project

To let code rebuild after each save of code, do following

1. right click on project and select properties
2. From right menu select C++ Build
3. Select Tab Behavior (on left)
4. Click on "Build on resource save"
5. Apply -> OK

Now try to follow these steps and write a hello world program. You must see the **binaries** folder on right side; it means a .exe is generated. Note, do not have **more than one** file ends with .cpp. In case binaries did not appear, you may try to clean the project [from build menu]. In case problems, you may search internet.

Debugging

Sure from time to time, you have bugs. Doing couts every where or over looking for code is not always the best solution. You may need a **debugger** to help you!

So let's debug following code little and see some debugging tricks!

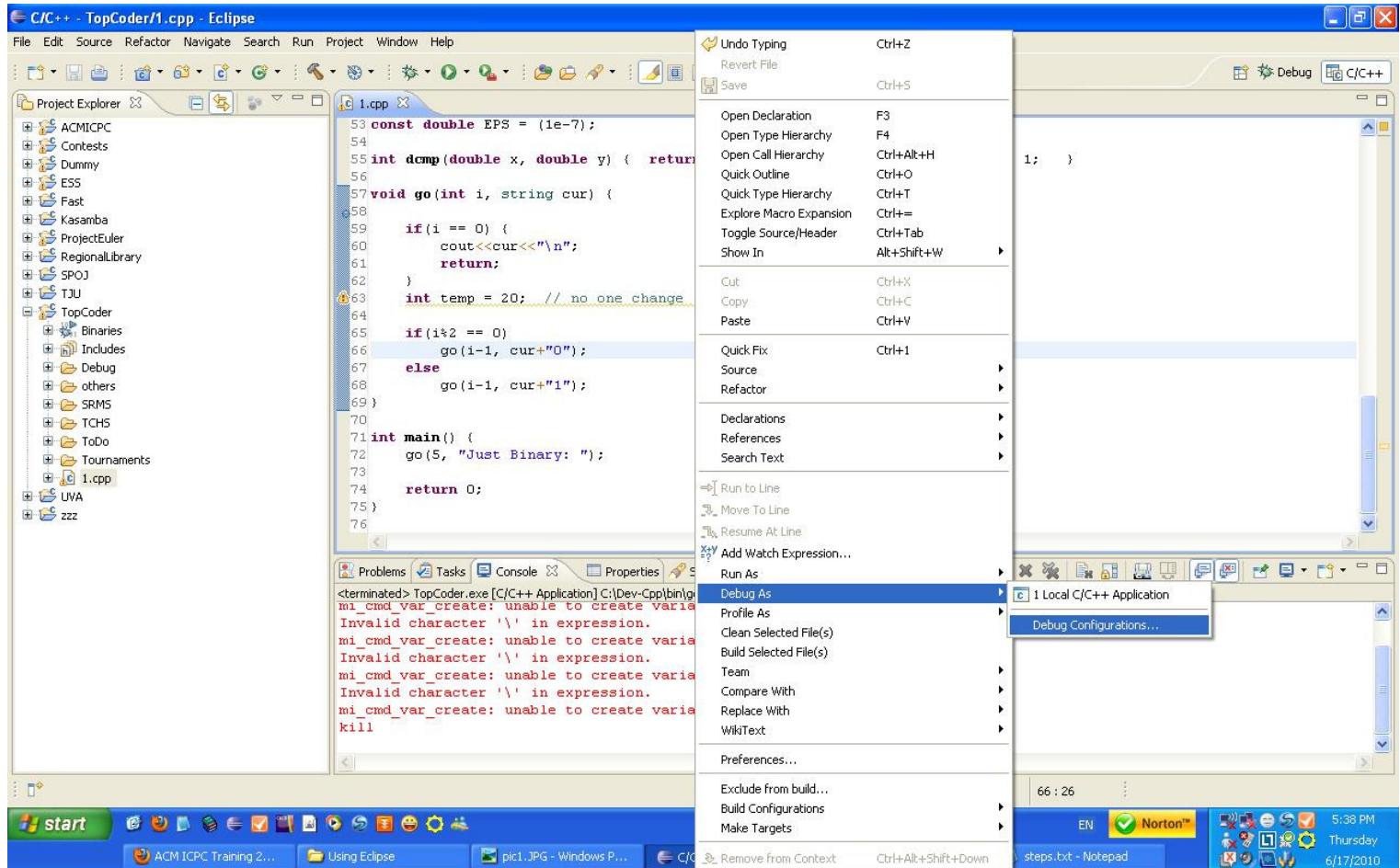


Figure 14: Select Debug as to configure it first time

Do you see line 58? It has a circle, we call it **break point**. We may like to jump to some places in code or continue process after a loop. We create break point, and just click **resume**

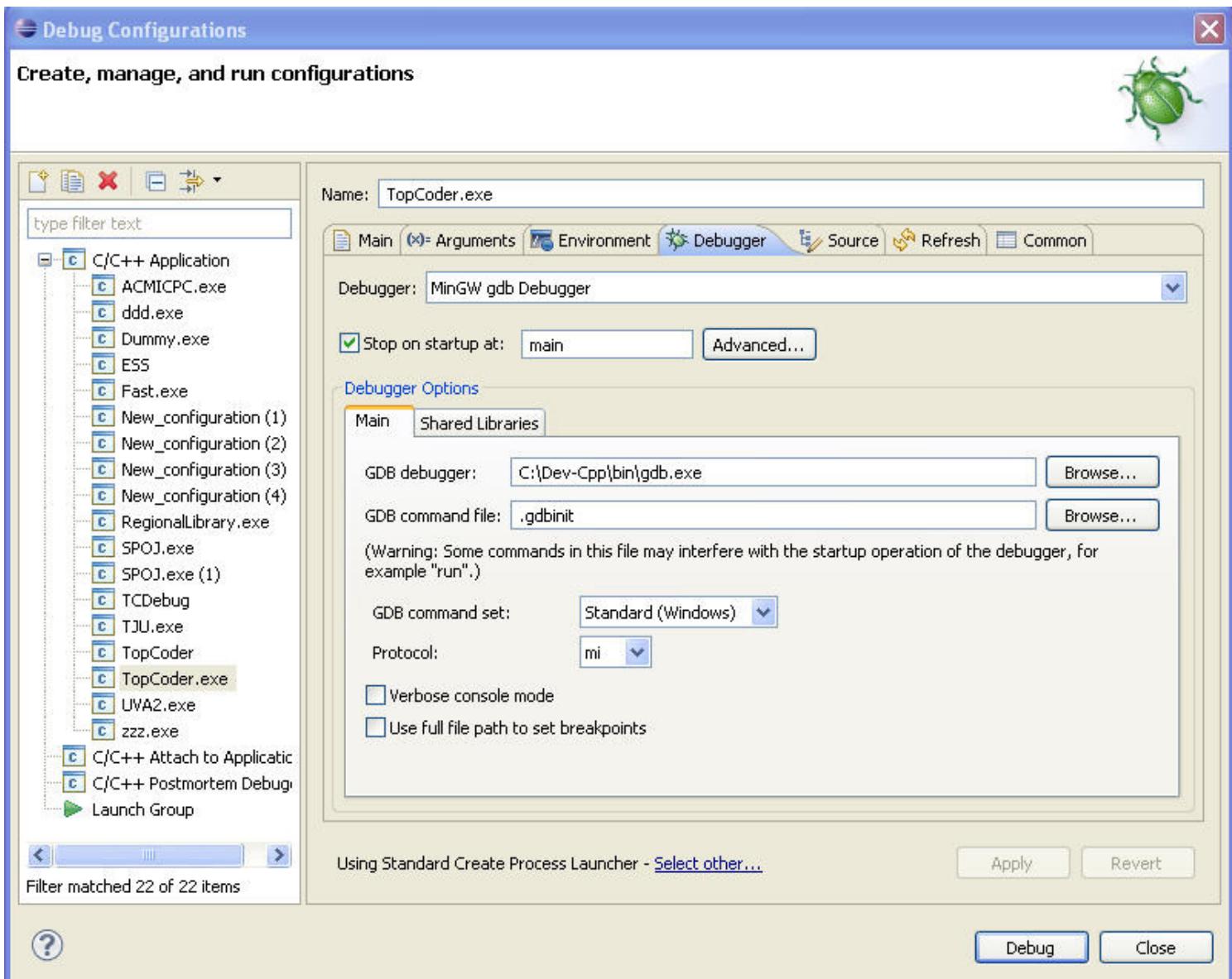


Figure 15: you may select minGW debugger

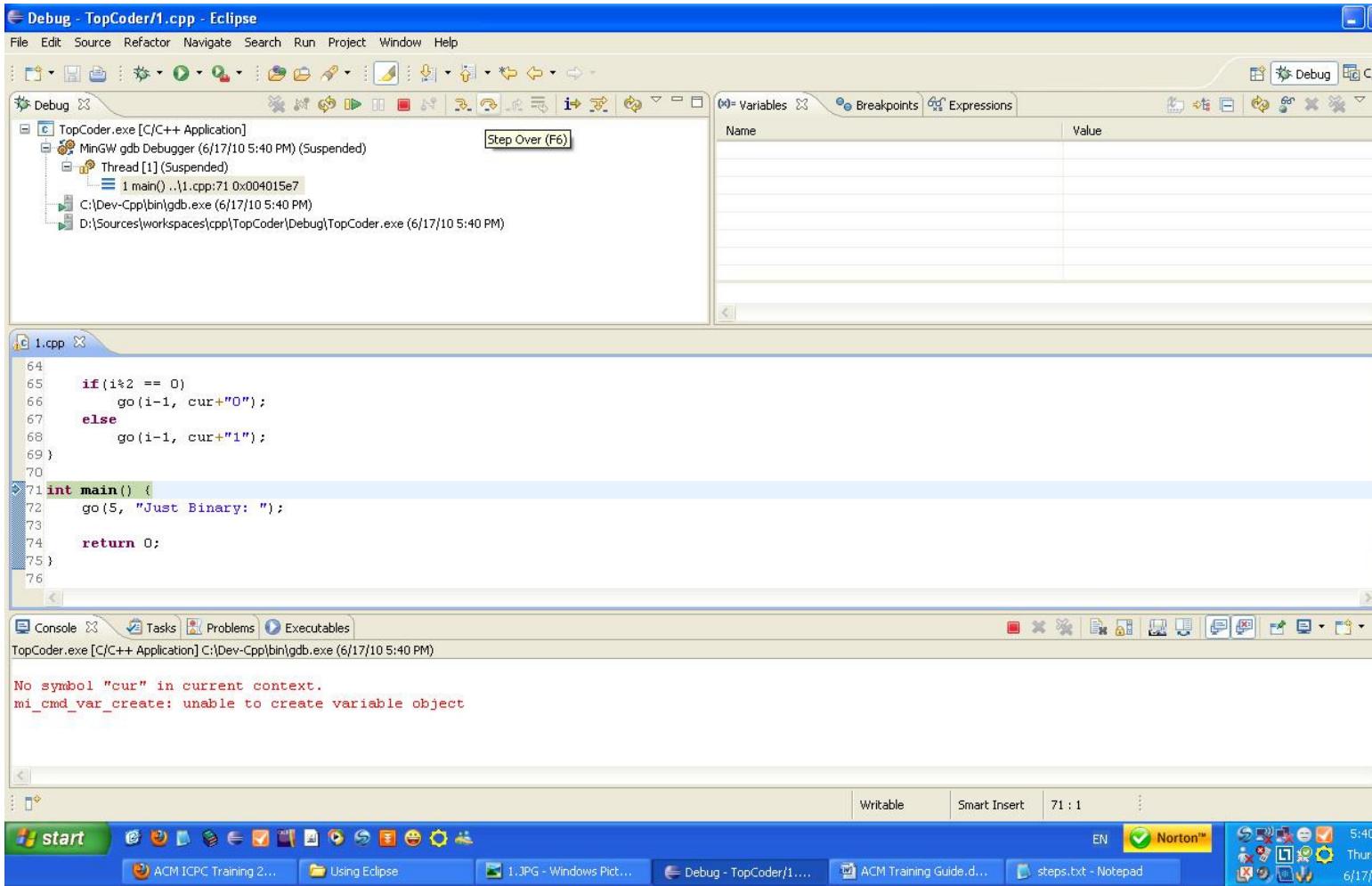


Figure 16L Step over button

We have 6 important instructions

- 1- **Resume** (the triangle), it means jump for next break point
- 2- **Step into**: this means, go inside details of that function
- 3- **Step over**, means pass on it as block statement, so if it is a function it may simply only return its answer
- 4- **Step out**, get you out of current function.
- 5- **Return**, if you are inside a function, you may return. This simply let reminder code be executed. Also sometimes debugger go into a built in functions, that time you need return
- 6- **Stop**

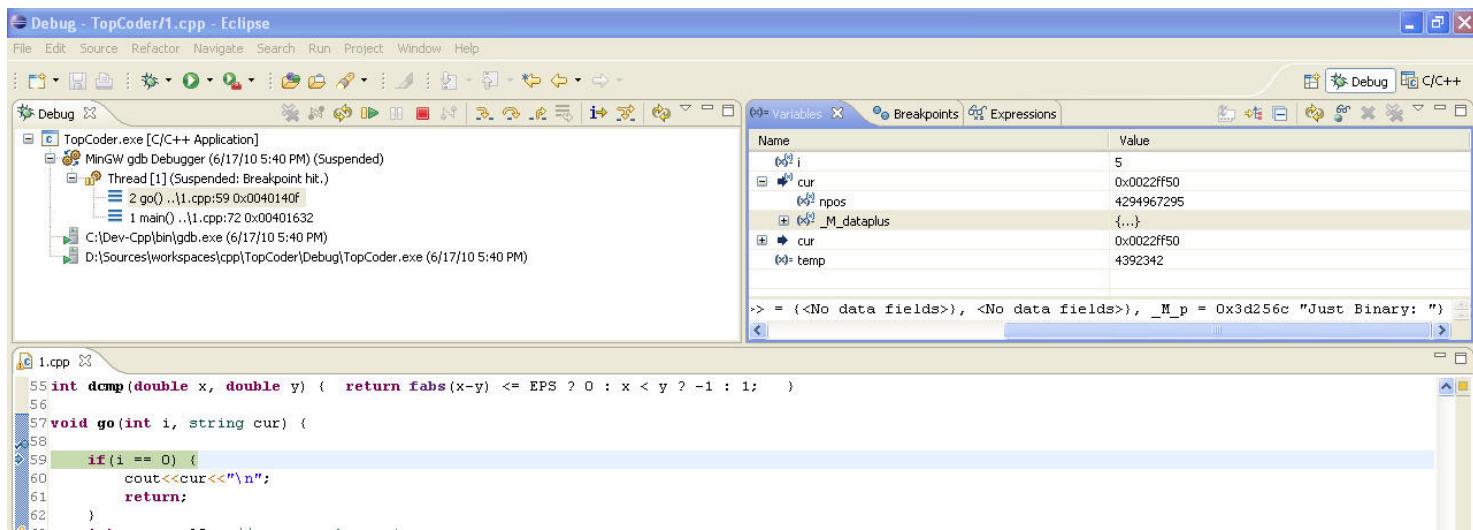


Figure 17: jump is applied

Now we clicked resume, so we jumped inside function! As you see in **variables section** on right. $i = 5$, $cur =$ and address. Clicking it, will shot string value. Hang over a variable and you could see its value(variable temp has value 20)

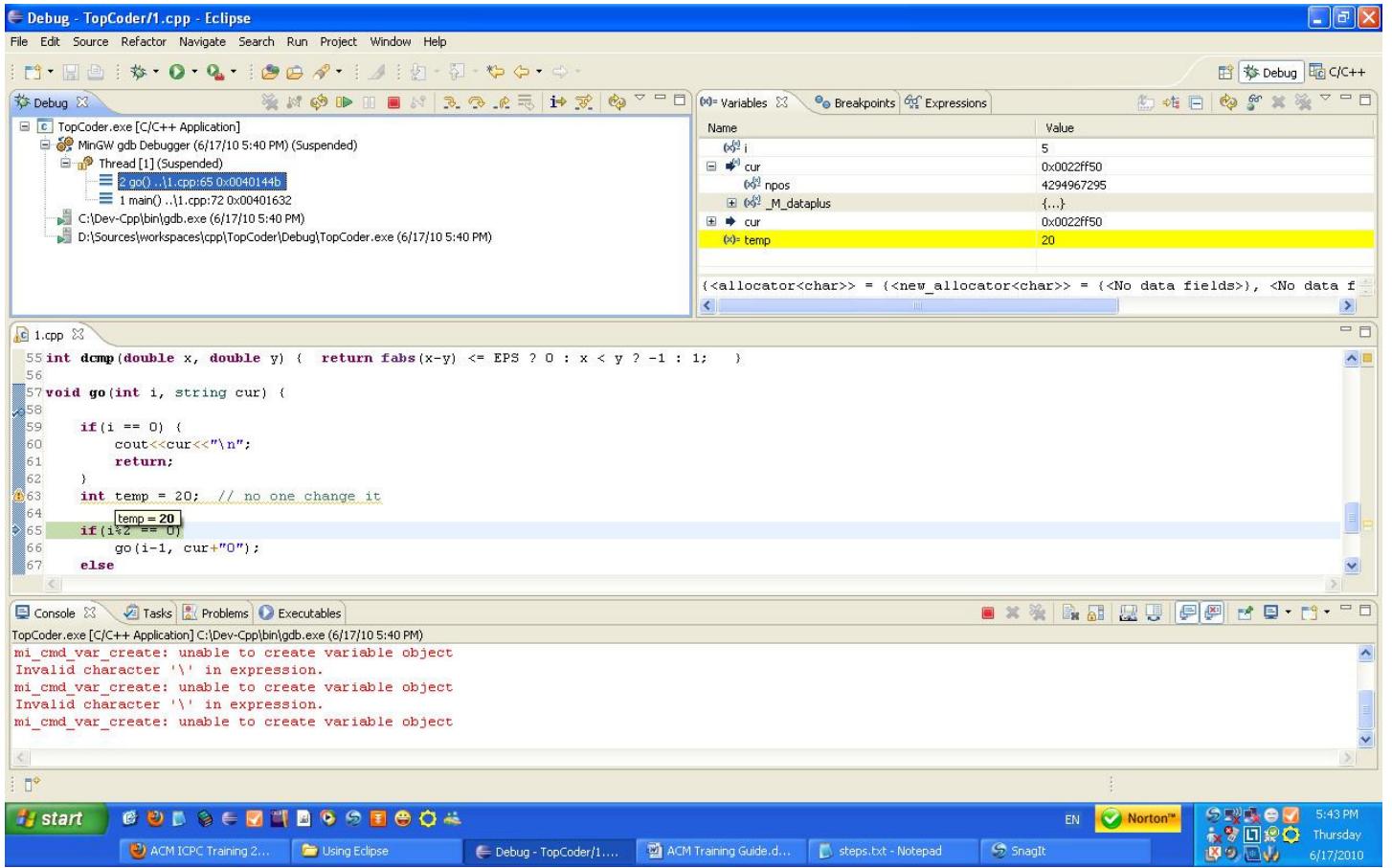


Figure 18: Browse a variable

Trick 1:

Here is a nice trick. Imagine that you are in **mid of debugging** and you realized that certain value is wrong. So, naturally you will stop debugging and fix code and rebug. One nice feature is “**hot fix**” where you can assign in run times values for variables. As you see we assigned temp to -11111, no one changes that variable!

The screenshot shows the MinGW gdb Debugger interface. On the left, the stack trace indicates a suspended thread at line 65 of file 1.cpp. The code editor shows a function named go with a line of code where temp is assigned the value -11111. A tooltip for this assignment says "temp = -11111". On the right, the registers window displays the current register values.

| Name | Value |
|-------------|------------|
| i | 5 |
| cur | 0x0022ff50 |
| npos | 4294967295 |
| _M_dataplus | {...} |
| cur | 0x0022ff50 |
| temp | -11111 |

```

TopCoder.exe [C/C++ Application]
MinGW gdb Debugger (6/17/10 5:40 PM) (Suspended)
Thread [1] (Suspended)
  2 go() ..1.cpp:65 0x0040144b
  1 main() ..1.cpp:72 0x00401632
C:\Dev-Cpp\bin\gdb.exe (6/17/10 5:40 PM)
D:\Sources\workspaces\cpp\TopCoder\Debug\TopCoder.exe (6/17/10 5:40 PM)

it dcmp(double x, double y) { return fabs(x-y) <= EPS ? 0 : x < y ? -1 : 1; }

id go(int i, string cur) {
    if(i == 0) {
        cout<<cur<<"\n";
        return;
    }
    int temp = 20; // no one change it
    temp = -11111
    if(i%2 == 0)
}

```

Figure 19: Hot fix for temp variable

Trick 2:

Another trick is **conditional breakpoint**. Imagine we have code that takes do many steps, and you suspect that when k = 2 the mistake will occur. You do not like to simulate all that, so let's jump to k only when it has that value!

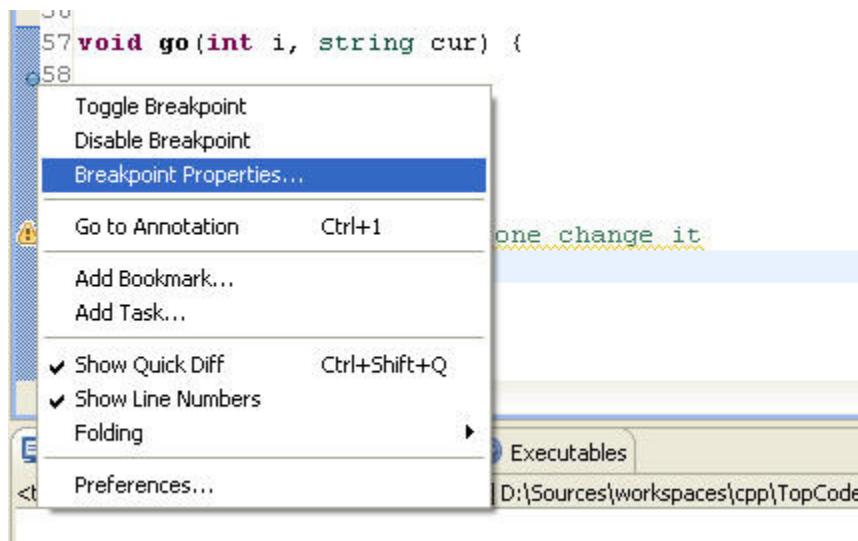


Figure 20: Adding properties to break point

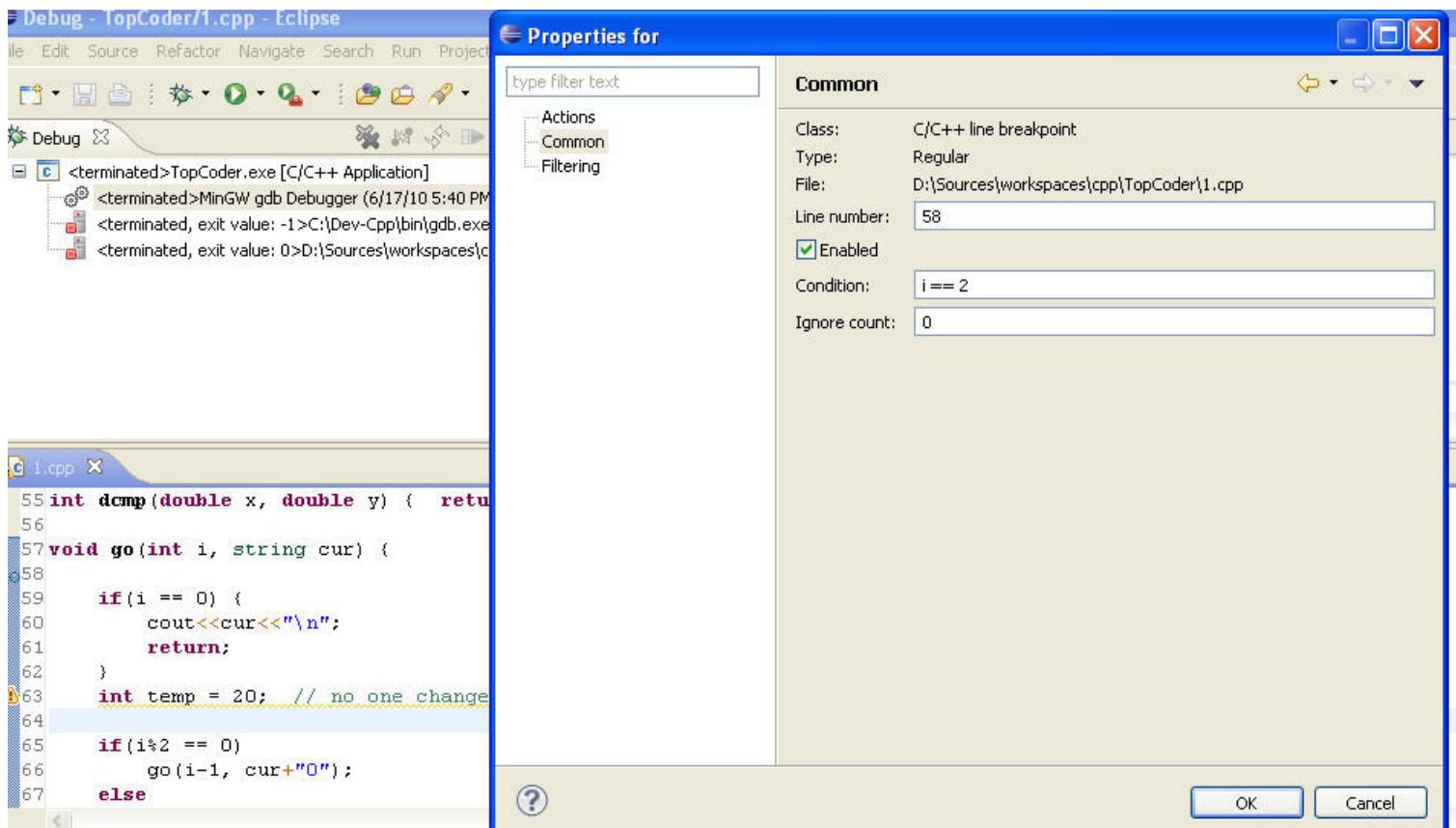


Figure 21: Put a condition similar to one in if else

Trick 3:

Finally, but I don't advice. In Visual Studio 2005 and up versions, you could use "Set Next Statement" from right click menu. This one will let the source code jump to this location and process again. This code help when you want to see the processing again of some steps. Most of time data will be corrupted :(.

Last tip:

In my opinion, visual studio debugger is the most powerful. You could view values of some data types much better than described above.

How to get help

Each online judge has its forms. Each form has people like you who suffered to find their mistakes and asked for help. It is always better to FIRST check other people efforts before start to fire the group with mails.

UVAFORUM

<http://online-judge.uva.es/board/>

UVA is a very old, its forms contains many responses. And even you can send a reply asking for more tests or helping catching a bug in your code. In the above box try to write the problem number, If no much results try the problem name.

TopCoder Forum

<http://forums.topcoder.com/>

TopCoder forums is more general. It is divided 2 many parts. Scrolling down to [Algorithm Matches](#), you will find the SRMs discussion. You can check what people asked and you can even re-ask.

Also in the [Educational Discussion](#) you can ask for help (but not for SRMs problems), but try to make your questions not be trivial and show that you did some effort.

Finally, topcoder is not like others, for each SRM you will find editorial that show you hints for the problem and some solutions at [Match Editorials](#). Open the SRM, and read it. In the above of editorial, you will find link, "**Discuss this match**", click it, you will go to this algorithm match in the forum.

SPOJ Forum

<http://www.spoj.pl/forum/>

I do not believe SPOJ forum are very good, but you can check them. Also you may search topcoder forum for UVA & SPOJ Problems.

Online Resources

<http://www.uwp.edu/sws/usaco/>

<http://www.spoj.pl/>

<http://uva.onlinejudge.org/>

<https://icpcarchive.ecs.baylor.edu/>

<http://www.topcoder.com/tc>

<http://www.topcoder.com/wiki/display/tc/Algorithm+Problem+Set+Analysis>

<http://www.topcoder.com/tc?module=ProblemArchive>

<http://online-judge.uva.es/board/>

<http://forums.topcoder.com/>

<http://www.spoj.pl/forum/>

<http://www.comp.nus.edu.sg/~stevenha/programming/acmoj.html>

http://www.algorithmist.com/index.php/Main_Page