

DOMJUDGE TEAM MANUAL

(FOR THE ACM SOUTH PACIFIC PROGRAMMING CONTEST)



SUMMARY

Here follows a short summary of the system interface. This is meant as a quick introduction, to be able to start using the system. It is, however, strongly advised that at least one of your team's members reads all of this manual.

DOMjudge works through a web interface – your documentation will give you the URL.

See Figure 4 on page 3 for an impression.

1 SUBMITTING SOLUTIONS

Solutions can be submitted from the web interface (see URL above).

In the left column (see figure 1) click **Select file...** to select the file for submission.

DOMjudge will try to determine the problem and language from the base and extension of the filename respectively. Otherwise, select the appropriate values from the drop down lists **problem** and **language**.

Note that the current compilers and versions on DOMjudge are: C/C++ – gcc version 6.3.0, Python 3.5.3, Java 1.8.0_131. C# Mono may be available for unofficial teams.

Compiler error messages will be provided to assist teams with different local environments.

After you hit the submit button and confirm the submission, you will be redirected back to your submission list page. On this page, a message will be displayed that your submission was successful and the submission should be present in the list. An error message will be displayed if something went wrong.

Submissions			
<div>Select file...problemlanguagesubmitcancel</div>			
time	problem	lang	result
19:01	F	JAVA	TOO-LATE
18:32	F	CPP	CORRECT
16:58	G	JAVA	WRONG-ANSWER
16:54	H	CPP	CORRECT
16:20	G	JAVA	TIMELIMIT
16:18	H	JAVA	WRONG-ANSWER
15:18	E	CPP	CORRECT

Figure 1 Making a submission

2 VIEWING THE RESULTS OF SUBMISSIONS

The left column of your team web page shows an overview of your submissions. It contains all relevant information: submission time, programming language, problem and status.

The top of the page shows your team's row in the scoreboard: your position and which problems you attempted and solved. Via the menu you can view the public scoreboard page with the scores of all teams (see figure 5 on page 4). The scoreboard will be 'frozen' one hour before the end of the contest. The full scoreboard view will not be updated any more, but your team row will. Your team's rank will be displayed as '?'.






RANK	TEAM	SCORE	A GOLF 	B MEMBERS 	C COINS 	D GEO PROG 
			[2 POINTS]	[2 POINTS]	[2 POINTS]	[3 POINTS]
1	 Team Manual	9 256	1/16	0	5	2/220

Figure 2 Team scoreboard

Clarifications				
time	from	to	subject	text
10:34	Jury	C# Solutions	general	Yes, just make sure you enter the correct problem letter.
Clarification Requests				
time	from	to	subject	text
10:32	C# Solutions	Jury	general	May we submit solutions in any order?
request clarification				

Figure 3 Clarifications

[overview](#) [scoreboard](#)

time left: 1:17:34
 contest: FProg2-3
 logged in as admin

RANK	TEAM	SCORE	A GOLF	B MEMBERS	C COINS	D GEO PROG
			[2 POINTS]	[2 POINTS]	[2 POINTS]	[3 POINTS]
1	 Team Manual	9 256	1/16	0	5	2/220

Submissions

Browse... No files selected. problem language submit cancel

time	problem	lang	result
13:42	C COINS	CSHARP	WRONG-ANSWER
13:40	D GEO PROG	CSHARP	CORRECT
13:38	C COINS	CSHARP	WRONG-ANSWER
13:10	D GEO PROG	CSHARP	WRONG-ANSWER
12:52	C COINS	CSHARP	WRONG-ANSWER
12:03	C COINS	CSHARP	WRONG-ANSWER
11:41	C COINS	CSHARP	WRONG-ANSWER
10:16	A GOLF	CSHARP	CORRECT

Clarifications

time	from	to	subject	text
10:34	Jury	C# Solutions	general	Yes, just make sure you enter the correct problem letter.

Clarification Requests

time	from	to	subject	text
10:32	C# Solutions	Jury	general	May we submit solutions in any order?

request clarification

Figure 4 Overview

Scoreboard ICPC South Pacific Divisional, 2016

final standings



RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J	K	L
1	AbishforeST University of New South Wales	12 918	1/5	2/116	1/12	1/30	1/37	4/199	1/18	1/46	1/149	1/15	2/137	1/54
2	nevered University of New South Wales	12 1148	1/6	1/91	1/15	1/105	1/27	2/236	1/12	1/123	2/251	1/19	2/164	1/39
3	Monash Epsilon Monash University	11 997	1/7	2/252	1/9	2/67	1/37	3/187	1/14	2/103	4	1/28	2/101	3/32
4	RidgeRacer The University of Western Australia	11 1307	1/4	3/183	1/10	3/88	1/95	7	1/20	2/282	1/246	1/29	1/84	1/166
5	PSPACEcreated University of Canterbury	9 962	1/11	1/272	2/25	1/194	1/104	0	1/43	0	0	1/33	2/153	1/87
6	the cake is a lie University of Canterbury	9 1206	1/6	1/292	1/19	3/142	1/77	0	1/16	3/265	0	3/41	3/188	1
7	Monash Omicron Monash University	8 781	1/7	1/232	1/15	4	1/58	0	1/12	0	0	1/25	2/147	3/225
8	UTAS University of Tasmania	8 866	2/7	0	1/10	1/142	1/37	3	1/43	0	0	3/189	1/284	1/94
9	OLD BOYS University of Melbourne	8 1078	1/10	0	1/14	3/178	1/85	0	1/24	0	0	2/55	5/242	4/270
10	Eliot Courtney University of Adelaide	7 615	1/3	0	1/9	3/278	1/58	0	1/21	0	0	1/71	2/115	0
11	O(1/n) University of Melbourne	7 659	1/7	0	1/22	3/152	2/89	0	1/17	0	0	1/48	3/224	3

Figure 5 Public Scoreboard

2.1 Possible results

A submission can have the following results:

CORRECT	The submission passed all tests: you solved this problem!
COMPILER-ERROR	There was an error when compiling your program. On the submission details page you can inspect the exact error.
TIMELIMIT	Your program took longer than the maximum allowed time for this problem, therefore it has been aborted. This might indicate that your program hangs in a loop or that your solution is not efficient enough.
RUN-ERROR	There was an error during the execution of your program. This can have a lot of different causes like division by zero, incorrectly addressing memory (e.g. by indexing arrays out of bounds), trying to use more memory than the limit, etc. Also check that your program exits with exit code 0!
NO-OUTPUT	Your program did not generate any output. Check that you write to standard output.
WRONG-ANSWER	The output of your program was incorrect. This can happen simply because your solution is not correct, but remember that your output must comply exactly with the specifications of the jury.
TOO-LATE	Unfortunately you submitted after the contest ended! Your submission is stored but will not be processed any more.

3 CLARIFICATIONS

All communication with the jury is to be done with clarifications (see figure 3). These can be found in the right column on your team page. Both clarification replies from the jury and requests sent by you are displayed there.

There is also a button to submit a new clarification request to the jury. This request is only readable for the jury and they will respond as soon as possible. Answers that are relevant for everyone will be sent to everyone.

4 HOW ARE SUBMISSIONS BEING JUDGED?

The DOMjudge jury system is fully automated. In principle no human interaction is necessary. The judging is done in the following way:

4.1 Submitting solutions

You submit a solution to a problem to the jury from the Web interface. Note that you have to submit the source code of your program (and not a compiled program or the output of your program).

There your program enters a queue, awaiting compilation, execution and testing on one of the jury computers.

4.2 Compilation

Your program will be compiled on a jury computer running Linux. The submitted source file will be passed to the compiler which generates a program to run.

Using a different compiler or operating system than the jury should not be a problem. Be careful however, not to use any special compiler and/or system specific things (eg Microsoft libraries for C++).

4.3 Testing

After your program has compiled successfully it will be executed and its output compared to the output of the jury. Before comparing the output, the exit status of your program is checked: if your program gives the correct answer, but exits with a non-zero exit code, the result will be a RUN-ERROR! (see Restrictions below).

Java does not require an exit code. However, be careful about catching exceptions. If you handle all exceptions in your code, you may get WRONG-ANSWER instead of the potentially more useful RUN-ERROR.

When comparing program output, it has to exactly match to output of the jury. So take care that you follow the output specifications. In case of problem statements which do not have unique output (e.g. with floating point answers), the jury may use a modified comparison function.

4.4 Restrictions

To prevent abuse, keep the jury system stable and give everyone clear and equal environments, there are some restrictions to which all submissions are subjected:

compile time Compilation of your program may take no longer than a specified time. After that compilation will be aborted and the result will be a compile error. In practice this should never give rise to problems. Should this happen to a normal program, please inform the jury right away.

source size The total amount of source code in a single submission may not exceed 256 kilobytes, otherwise your submission will be rejected.

memory During execution of your program, there are 2 gigabytes of memory available. This is the total amount of memory (including program code, statically and dynamically defined variables, stack, Java VM, etc)! If your program tries to use more memory, it will abort, resulting in a run error.

number of processes You are not supposed to create multiple processes (threads). This is to no avail anyway, because your program has exactly 1 processor fully at its disposal. To increase stability of the jury system, there is a maximum of 15 processes that can be run simultaneously (including processes that started your program).

People who have never programmed with multiple processes (or have never heard of "threads") do not have to worry: a normal program runs in one process.

APPENDIX : CODE EXAMPLES

Below are a few examples on how to read input and write output for a problem.

The examples are solutions for the following problem: The first line of the input contains the number of test cases. Then each test case consists of a line containing a name (a single word) of at most 99 characters. For each test case output the string Hello <name>! on a separate line.

Sample input and output for this problem:

Sample Input	Output for Sample Input
3 world Jan SantaClaus	Hello world! Hello Jan! Hello SantaClaus!

Note that the number 3 on the first line indicates that 3 test cases follow.

1 Solutions for Contest Languages

i C:

```
1 #include <stdio.h>
2
3 int main()
4 {
5
6     int i, ntests;
7     char name[100];
8
9     scanf("%d\n", &ntests);
10
11     for(i=0; i < ntests; i++)
12     {
13         scanf("%s\n", name);
14         printf("Hello %s!\n", name);
15     }
16     return 0;
17 }
```

Notice the last line, return 0, which prevents a RUN-ERROR!

ii C++:

```
1 #include <iostream>
2 #include <string>
3
4 using namespace std;
5
6 int main()
7 {
8
9     int ntests;
10    string name;
11
12    cin >> ntests;
13
14    for(int i=0; i < ntests; i++)
15    {
16        cin >> name;
17        cout << "Hello " << name << "!" << endl;
18    }
19    return 0;
20 }
```

Notice the last line, return 0, which prevents a RUN-ERROR!

iii Java:

```
1 import java.io.*;
2
3 public class Main
4 {
5     public static BufferedReader in;
6
7     public static void main(String[] args) throws IOException
8     {
9         in = new BufferedReader(new InputStreamReader(System.in));
10
11         int nTests = Integer.parseInt(in.readLine());
12
13         for(int i = 0; i < nTests; i++)
14         {
15             String name = in.readLine();
16             System.out.println("Hello " + name + "!");
17         }
18     }
19 }
```

iv Python 3

```
1 num=int(input())
2
3 for i in range(num):
4     name = input()
5     print("Hello " + name + "!")
```

2 Solutions in Other Languages

i C#

```
using System;
namespace HelloCS
{
    class Program
    {
        static void Main(string[] args)
        {
            int nTests = Convert.ToInt32(Console.ReadLine());

            for (int i = 0; i < nTests; i++)
            {
                string name = Console.ReadLine();
                Console.WriteLine("Hello {0}!", name);
            }
        }
    }
}
```

NOTE If you develop your solution in Visual Studio, a number of libraries will be referenced with using statements. Many of these are not necessary (eg System.Linq) and may be removed.

ii Python 2

```
1 num = int(raw_input())
2
3 for i in range(num)
4     name = raw_input()
5     print "Hello " + name + "!"
```