

## CONTEST FACILITY FAILURE PROCEDURES

Mon Aug 5 14:09:38 EDT 2002

A general rule to keep in mind is that while most contest facility failures affect some teams more than others, this is all right as long as these affects are due to accident, luck, and the individual choices of the teams, and are not due to any bias that the contest judge and managers can remove. In other words, winning is a matter of being good AND being lucky (or at least not unlucky), and when things fail the luck factor rears its head.

There are several varieties of contest failure. All involve changing the contest rules, either temporarily or permanently. Such changes are best made in writing. At the end of this document are pages that can be printed, edited by hand, xeroxed, and handed out, in order to affect temporary and permanent rules changes.

In general, to prepare for failures, you need a printout of this document, a stapler, a supply of paper, and access to a Xerox machine. A good knowledge of the editors, compilers, and debuggers that are on your local computers can be useful, but is not necessary.

In general, in the presence of failures a paper trail must be generated. One copy of everything needs to be carefully preserved so a copy of the paper trail can be US mailed to the judge.

#### Temporary Failures in Central Judging

-----

The judge will fix these, and the contest will go on. There should be no site-specific bias favoring teams at some sites over teams at other sites.

For failures that last more than a few minutes, forms below should be used to notify teams to stop using computers, and to resume after the failure is fixed.

#### Temporary Failures at a Single Site

-----

The site manager will fix the problem, and ensure as much as reasonably possible that all teams at the site have equal facilities. For example, if half the site team PCs are disconnected from the net for an hour, the site manager will instruct ALL teams at the site to stop using their PCs during that hour. Forms below should be used to inform the teams.

If the problem results in an inequity between teams at the affected site and teams at other sites, the judge will effectively split the contest into two contests, one at the affected site and one at all other sites. The split will be accomplished solely by providing two separate scoreboards, one for each of the two contests. The judge can do this as contest support software can compute multiple scoreboards and assign each team to one of these. Submissions and judging of individual problems are not affected in any way by scoreboard computation. Although both contests will have the same judging and non-terminal computer facilities, they will be reported as two separate contests.

If the decision to split the contest is problematic, it is possible to compute both split and non-split scoreboards, to announce all results, and to decide in the following days whether the contest is split.

The general rule is that a single site network outage of 10 minutes or less, total accumulated over the entire contest, should NOT cause the contest to be split, and should be ascribed solely to the 'luck of the game'. Other disruptions can be subjectively compared to this to determine whether a contest should be split.

#### Permanent Failures at a Single Site

-----

The site manager must change the rules of the contest and issue a new set of rules in writing. The forms below are for that purpose.

You must chose whether to permit teams to use editors, compilers, and debuggers on local computers. If you are unfamiliar with what is available, or if the software available is not equal on all the team's computers, then use of the software should be prohibited. You can permit editors without permitting compilers, or compilers without permitting debuggers. You can start off prohibiting computers (using the Temporary Failure Contest Rules below), then later after investigating the available editors, allow editors only, and then still later, after investigating the compilers that are available, you can permit both editors and compilers to be used. For example.

A site manager should never attempt judging unless she/he has support software that has recently been tested and he/she has practiced judging with that software.

#### Forms to be Handed out upon Failures

-----

Various forms follow. Edit them by hand and fill in the times before Xeroxing them.



Temporary Failure Contest Rules:  
-----

As a consequence of \_\_\_\_\_

\_\_\_\_\_

the following contest rules go into effect as of

\_\_\_\_\_ AM PM

1. Each team will cease all use of computers, and work solely with paper and pencil or pen.
2. We expect to continue the contest normally well before the stop time of the contest. However, if we cannot, the rules may be changed so teams will submit solutions on paper that will be graded by the judge, after the contest is over, as if they were exam problems. Teams may wish to take this possibility into account when writing their solutions on paper, by writing more clearly, for example.
3. When we resolve our operating difficulties, you will be notified in writing.



Resumption of Normal Contest Rules:

-----

The previous computer or network difficulties have been fixed and normal contest rules resume effect as of

\_\_\_\_\_ AM PM

and will (hopefully) remain in effect until the contest stop time at

\_\_\_\_\_ AM PM

Thus you can resume using your computers as terminals in the normal way.

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

Thank you for your patience.





Computer Failure Contest Management Rules:  
-----

Effective as of \_\_\_\_\_ AM PM

1. Each team will cease all use of computers, and work solely with paper and pencil or pen.
2. Each team will write up a solution to each problem it wishes to submit, as a piece of code. This solution will be graded by the judge, after the contest is over, as if it were an exam problem. Partial credit will be given for partial solutions.
3. The solution may be commented. Comments may be in the code, may precede the code, or may follow the code, and need not be in computer format. The comments will not be directly graded per se, but if the judge cannot figure out a section of code in a reasonable period of time, the judge will give that section a failing grade, and comments may help avoid this.
4. All submissions must be clearly marked at the top of the first page with the team name and the problem name. Different submissions by the same team will be kept separate.
5. When a team is finished with a submission, it will take the submission to the site managers located at  
\_\_\_\_\_  
\_\_\_\_\_

There the submission will be stapled together and marked with the time of submission. Each individual submission by one team will be stapled separately.

Most submissions will have some defect, so there is no firm concept of incorrect submission. Time will be a minor contributor to the score of each submission.

If a team turns in two submissions for the same problem, only the later submission will be scored.

6. A team may ask the judge to have one of its submissions back. The returned submission will be marked with the time it was returned to the team. The team may modify the submission and resubmit. If they do not, the submission is treated as if it never existed. If a submission was turned in once, returned, and then resubmitted, the only factors affecting scoring will be the contents of the submission when it was last turned in and the time the submission was last turned in.
7. If a team has a question about a problem, it may submit it in writing to  

---

  

---

The site managers will convey questions to the judge who will convey answers back to the site managers, and then the site managers will hand out written answers to all teams.

8. At the end of the contest all submissions waiting in line at the site manager's location given in item 5 above will be collected and marked with the end-of-contest time.
9. After the contest, submissions will be US mailed to the judge, along with copies of all written rules changes, questions, and answers. The judge will grade the submissions over the next several days.

Network Failure Contest Management Rules:  
-----

Effective as of \_\_\_\_\_ AM PM

1. Each team will cease using the network, and only use its computer locally.

In using its local computer, the following are allowed or disallowed to each team as indicated:

editing	allowed	
compiling	allowed	disallowed
running	allowed	disallowed
using debugger	allowed	disallowed

The selections just given are designed to provide reasonably equitable support for the following computer languages:

C	C++	JAVA	_____
_____			

2. Each team will write up a solution to each problem it wishes to submit, as a piece of code in a single file. This solution will be graded by the judge, after the contest is over, as if it were an exam problem. Partial credit will be given for partial solutions.

3. The solution may be commented. Comments may be in the code, may precede the code, or may follow the code. The comments must be in the commenting format of the computer language being used, so as not to interfere with compiling the code (the judge may eventually fix and compile the code even if teams are not allowed to). The comments will not be directly graded per se, but if the judge cannot figure out a section of code in a reasonable period of time, the judge will give that section a failing grade, and comments may help avoid this.
4. Each submission file must be clearly marked at its beginning with the team name and problem name, or one or both of these may be included in the file name.
5. During the contest each team should keep on a single piece of paper a 'submission log' that lists for each of the team's submissions the problem name and file name. This log must be turned in immediately at the end of the contest, and should be kept up to date during the contest. Be sure the team name is on the log.
6. When a team is finished with a submission, it will execute the command:

```
md5sum FILENAME _____
```

to determine the MD5 checksum of the file. The team will then put this checksum, the name of the file, and the name of the team on a single piece of paper and take that paper to the site managers located at

\_\_\_\_\_  
\_\_\_\_\_

There the paper will be marked with the time of submission.

Many submissions will have some defect, so there is no firm concept of incorrect submission. Time will be a minor contributor to the score of each submission.

If the team changes the file after the time of submission, the file's MD5 sum will change, and any previous submission of the file will become INVALID. A new submission can be made, however, with the changed file.

If a team makes several submissions for the same problem, only the latest will be scored.

It is a good idea for a team to write the time of any submission it makes in this manner in the team's 'submission log', but this is not required.

7. If a team has a question about a problem, it may submit it in writing to

---

The site managers will convey questions to the judge who will convey answers back to the site managers, and then the site managers will hand out written answers to all teams.

8. At the end of the contest each team must immediately submit its 'submission log'. The team name must be on the log. All submissions on this log that have not also been made by the means of item 6 above will be marked with the 'end-of-contest' time.

A team must not submit two end-of-contest files for one problem: if it does so, it will get no credit for that problem.

9. The site managers will collect all submission files immediately after the contest and verify MD5 sums of files submitted before the contest end. All files that verify will be conveyed by the site managers to the judge, along with copies of team logs, team MD5 submission papers, and all written contest rules changes, questions, and answers. The judge will score the files over the next several days.
10. Unfamiliarity with the computer facilities you will use in the event of network failure is considered to be part of the 'luck of the game', and not a cause for changing the contest rules.

You are entitled to a few minutes of instruction using the following procedures. Put up a sign naming in large letters the kind of instruction you want, chosen from the following list. An instructor will visit you. Instructors give priority to the kinds of instruction appearing earlier in this list, and otherwise try to use an unbiased (random or round-robin) approach to who they see next. The kinds of instruction are:

1. Editing [This is just enough to edit a file.]
2. Printing
3. Compiling and Running
4. Advanced Editing
5. Debugging

You are only entitled to a few minutes instruction at one time. But that should be all that is needed. First learn to edit a file using very few editor commands if the editor is new to you. Then edit in a simple program that will execute, and start editing in solutions while waiting for instruction in compiling and running the simple program. Ask for other kinds of instruction when you are ready to absorb the extra instruction.

If you have an emergency issue, like your editor is hung up and you cannot proceed, put up a sign that says EMERGENCY or contact the site manager directly.

List of commands:

list the files in the current directory:

```
ls _____
```

rename a file:

```
mv OLD-FILENAME NEW-FILENAME
```

FILENAMES:

JAVA: PROGRAM.java where PROGRAM also names the class with the 'main' static function.

```
_____
```

C: PROGRAM.c \_\_\_\_\_

C++: PROGRAM.cc \_\_\_\_\_

edit a file:

vi FILENAME \_\_\_\_\_

Quit editor: :q \_\_\_\_\_

Emergency quit: :q! \_\_\_\_\_

Write file: :w \_\_\_\_\_

Move around: <arrow key> \_\_\_\_\_

Delete character: x \_\_\_\_\_

Insert text before cursor:

iTEXT<escape>

\_\_\_\_\_

Insert text after cursor:

aTEXT<escape>

\_\_\_\_\_

Insert text after current line:

oTEXT<escape>

\_\_\_\_\_

Insert text before current line:

OTEXT<escape>

\_\_\_\_\_

Delete current line: dd \_\_\_\_\_

Undo last change: u \_\_\_\_\_

Undo last undo: u or control-R

\_\_\_\_\_



edit a file:

emacs FILENAME \_\_\_\_\_

Quit editor:

<control-X><control-C> \_\_\_\_\_

Write file:

<control-X><control-S> \_\_\_\_\_

Move around: <arrow key> \_\_\_\_\_

Delete character before cursor:

<delete>  
\_\_\_\_\_

Insert text before cursor:

just type the text \_\_\_\_\_

Delete to end of line, or delete line if  
already at end:

<control-K> \_\_\_\_\_

Undo last change:

<control-\_> \_\_\_\_\_

Get help: control-H a TOPIC ENTER

Go to other window: control-X o

\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_  
\_\_\_\_\_

compile a program:

```
javac FILENAME
gcc -g -o PROGRAM FILENAME -lm
g++ -g -o PROGRAM FILENAME -lm

[-g includes debugging information]
[Normally PROGRAM is FILENAME without its
extension.]
```

---

---

---

---

---

---

---

---

run a program

```
java PROGRAM _____
INPUT-LINES
<control-D> _____
```

```
java PROGRAM < INPUT-FILENAME
```

---

```
PROGRAM _____
```

```
PROGRAM < INPUT-FILENAME
```

---

debug a program

jdb PROGRAM \_\_\_\_\_

get help: help \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

gdb PROGRAM \_\_\_\_\_

get help: help \_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_

\_\_\_\_\_