

# The 3rd National Programming Competition

## Instructions and Guidelines

### Contents

1. Introduction.....	2
2. Contestants.....	2
3. Conduct of the Tournament.....	2
4. Problems Format.....	3
5. Programming Your Solutions.....	4
6. Judged Runs and Responses.....	5
7. Scoring.....	6
8. Example of Scoring.....	6
9. Awards and Prizes.....	7
10. F.A.Q.....	7

## 1. Introduction

The NC3 Programming Contest is the third national programming contest in Saudi Arabia, fostering creativity, teamwork, and innovation in building new software programs, and enabling students to test their ability to perform under pressure. The contest will follow the format of the traditional ACM International Collegiate Programming Contest (ICPC). Up to two teams, of three students each, will represent their university and attempt to solve eight or more complex, real-world problems, with a challenging five-hour deadline. Huddled around a single computer, competitors race against the clock in a battle of logic, strategy, and mental endurance. Teams must quickly rank the difficulty of the problems, deduce the requirements, design test beds, and build software systems that solve the problems under the intense scrutiny of expert judges. For a well-trained team, some of the problems require precision only. Others require a knowledge and understanding of advanced algorithms. Still others are simply too hard to solve – except, of course, for Saudi's brightest problem-solvers. The team that solves the most problems with the least penalty points (see rules) is declared the winner.

## 2. Contestants

For this competition, each contestant must simply be an undergraduate student currently enrolled in a participating Saudi university. Students should have studied at least one or two courses in programming and should be comfortable programming in JAVA as it will be the language used during the competition.

## 3. Conduct of the Tournament

- 3.1. Eight to ten problems will be given at the beginning of the contest.
- 3.2. Contestants will be given four hours to solve the problems.
- 3.3. Problems must be solved in in JAVA using the editors provided on the contest machine.
- 3.4. Contestants may bring any materials to the contest except computing devices (including calculators), machine-readable media, or devices requiring a wall outlet. The usage of electronic devices are explicitly prohibited during the competition. The coordinator can grant an exception for emergency communications.
- 3.5. Contestants may work on the problems in any order.

3.6. A contestant may submit a solution to the judges at any time during the contest. The solution is checked on the Judge's computer against a set of secret test cases. After a judged run, the contestant will receive a message with one of the following responses from the judges:

- Correct
- Compile-Time Error
- Run-Time Error
- Read from File
- Time-Limit Exceeded
- Output to Screen
- Wrong Answer
- Too Much Output
- Incorrect Output Format
- Incomplete Output

These responses are explained in Judged Runs and Responses (See Section 6).

3.7. The judges will not answer questions about any problem. If a contestant feels that a problem is stated ambiguously and requires clarification, he may submit a statement of ambiguity to the judges. The statement of ambiguity will be answered if the judges agree that a clarification is required. The statement and the answer will be provided to all teams if the judges deem it necessary. The judges may refuse to answer any statement of ambiguity.

3.8. Contestants may not discuss the contest problems with anybody except the contest staff.

3.9. The Contest Coordinators have the power to deal with any infraction of the above rules through penalty or disqualification, at their discretion.

## 4. Problems Format

4.1. Every effort is made to guarantee that problems avoid dependence on detailed knowledge of a particular application area or particular programming language.

4.2. Each problem will require reading input from a specific text file, and printing its output to the standard console (to the screen)

4.3. Since judging is a semi-automatic process, it is mandatory for the program output to match the output format specified in the problem description.

4.4. The judges' secret input file will test the program on multiple cases (or datasets.) The format of the input file will be designed so that multiple datasets can be included in a single text file.

4.5. Each problem in the contest is specified in the following four sections:

- Description: This section specifies the problem contestants are supposed to solve.
- Input format: Here you'll find detailed description on how the input file will be formatted.

- Output format: This section specifies the format of the output of your program. Your program's output must exactly follow the format specified.
- Sample Input/Output: Here you'll find a sample input/output that has successfully passed the judges' program.

**Note** that your program will be tested on a different (and more complex) dataset. Just because your program passed the sample Input/Output, doesn't mean it is correct.

4.6. In addition, the problem statement will specify the name you should use for your program, and the name of the input file.

4.7. Since compilation and testing is an automated process, it is important that you follow the naming convention exactly.

- For example, if the problem statement states that the files should be named easy then the program file must be named **easy.java**. input file will be named **easy.in**

## 5. Programming Your Solutions

5.1. Programs must be in a single source file with the name specified in the problem description; failure to meet this requirement is a "Syntax or Compilation Error".

5.2. Note: You can still have more than one top-level classes in a single source file as long as only the main one is declared public while the rest are unqualified (i.e. just 'class', without 'public').

5.3. Output will be judged using a file comparison utility. Output must be exactly as specified in the output format section: spelling, punctuation, spacing, and case (uppercase/lowercase) are all significant (unless specified otherwise in the problem statement.)

5.4. Your program cannot require any intervention from the user. All input must be read from the input file.

5.5. If you submit a program that requires user intervention, you'll most probably receive a "Time-Limit Exceeded" error.

- Contestants who have the habit of placing a "press any key to continue" statement at the end of their program should pay attention to this point.

5.6. Do not use drive and/or path specifications when naming input files. The input file will be in the same directory as your program.

5.7. All test cases used in judging will conform to the input specifications. There is no point for your program to try to detect invalid input or missing input file.

## 6. Judged Runs and Responses

A program submitted for a judged run must pass the following criteria (in order):

1. It must successfully compile on the judges' machine.
2. It must execute on the judges' machine successfully (no abnormal termination, machine lock-up, etc.) and completely (within the given time limit).
3. It must read input from a file and give its output to the screen (stdout).
4. It must demonstrate that it solves the problem by producing correct answers.
5. It must present the correct answers in the proper format, specified in the problem statement.
6. It must produce complete output (answers for all cases given in the input file).

When you submit a program for a judged run, you will receive one of the following responses:

Response	Explanation
Correct	Your program has passed all the criteria listed above. You have solved that problem.
Compile-Time Error	Your program did not successfully compile at the judges' station.
Run-Time Error	Your program terminated abnormally, or locked-up the judges' machine.
Read from File	Your program is not reading from an input file as instructed.
Time-Limit Exceeded	Your program ran longer than one minute on the judges' machine.
Output to Screen	Your program is not outputting to the screen (stdout) as instructed.
Wrong Answer	Your program produced incorrect answers when tested on the judges' input data. It is not an acceptable solution to the problem.
Too Much Output	Your program produces correct answers, but also outputs information which was not asked for in the problem specification.
Incorrect Output Format	Your program produced correct answers to the problem, but did not present the output in the manner specified in the problem.  <b>Note:</b> If formatting is an integral part of solving the problem (e.g. generating some kind of report or a maze), then "Wrong Answer" will be given instead of this message.
Incomplete Output	Your program fails to produce an answer for every test case in the input file.

## 7. Scoring

The contestant which has had the most problems judged correct at the end of the contest is declared the winner. If more than one contestant has solved the same number of problems, the winner is the one with the fewest penalty points.

- 6.1. Penalty points are given only for solved problems.
- 6.2. One penalty point is assessed for each minute from the beginning of the contest until the problem was solved.
- 6.3. Twenty penalty points are assessed for each judged run except one "correct" submission (note that multiple correct submissions on a problem receive a penalty).
- 6.4. No penalty points are assessed for any problem which was not solved.
- 6.5. The Contest Coordinators reserve the right to adjust penalty points to allow for unforeseen circumstances.
- 6.6. The score will be "frozen" in the final hour (meaning that score updates will not be publicly shown) in order to preserve suspense for the end of the competition.
- 6.7. For more clarification, see the next section, Example of Scoring.

## 8. Example of Scoring

Consider a contest involving three contestants (A, B and C) in which three problems are posed. Contestant A solves Problem Two after 60 minutes, with two judged runs, and Problem One after 120 minutes with 1 judged run. Although the contestant submits programs for the third problem three times, it is not solved. Penalty Points for Contestant A:

- Problem 1: 120 for time plus 0 for 0 extra judged runs totaling 120
- Problem 2: 60 for time plus 20 for 1 extra judged run totaling 80
- Problem 3: 0 (No penalty points assessed for problems not solved) Total penalty points assessed is **200**.

Contestant B solves Problem One after 60 minutes, with five judged runs. The second problem is not solved. The contestant solves the third problem after 180 minutes with 1 judged run. Penalty Points for Contestant B:

- Problem 1: 60 for time plus 80 for 4 extra judged runs totaling 140
- Problem 2: 0
- Problem 3: 180 for time plus 0 for 0 extra judged run totaling 180 Total penalty points assessed is **320**.

Contestant C solves Problem Three after 60 minutes, with three judged runs, Problem One after 120 minutes with 3 three judged runs, and Problem Two after 180 minutes and 3 judged runs.

Penalty Points for Contestant C:

- Problem 1: 120 for time plus 40 for 2 extra judged runs totaling 160
- Problem 2: 180 for time plus 40 for 2 extra judged runs totaling 220
- Problem 3: 60 for time plus 40 for 2 extra judged run totaling 100

Total penalty points assessed is 480.

The final results of the contest are summarized below:

Place	Contestant	Number Solved	Penalty Points
1	C	3	480
2	A	2	200
3	B	2	320

**Note** that Contestant C wins as the one that has solved the most problems. Contestants A and B both solved two; so the tie is broken with penalty points. Contestant A places second due to fewer penalty points than Contestant B.

## 9. Awards and Prizes

TBD

## 10. F.A.Q.

### **I do not know JAVA , Should I enter the competition?**

All students are welcome to enter the competition, of course with the condition that you are familiar with JAVA programming languages.

### **Why are some of the problems so hard?**

We are impressed with the talent of many students. The problems, therefore, contain a variety of difficulty levels.

### **Why are some of the problems so easy?**

Students participating in the contest are from different levels in their study. Some students have only taken an introductory programming course. Others have only taken a course in object-oriented programming and even data structures and algorithms. And still others have taken

KINGDOM OF SAUDI ARABIA

Ministry of Education  
Saudi Universities Computing and  
Information Colleges Deans'  
Committee



المملكة العربية السعودية

وزارة التعليم  
لجنة عمداء كليات الحاسب والمعلومات  
بالجامعات السعودية

additional courses in algorithms. Therefore, the problems contain a variety of difficulty levels. Additionally, some problems may be very “easy” to code (less than 20 lines), but the problem may require strong knowledge of mathematics. So the term “easy” is relative: what is easy for one student may be very challenging for another.

### **Can I bring notes with me to the contest?**

You are welcome to bring any written or non-electronic materials that you wish! Of course, this includes notes, books, printouts, etc.

### **What specifically can I not bring or use?**

You cannot bring or use any electronic device. This includes anything and everything electronic: laptops, tablets, pads, phones, etc.

### **Contact**

**Mahmoud Khattab**

**n3c@kku.edu.sa**