Olympiads in Informatics, 2015, Vol. 9, 163–172
DOI: http://dx.doi.org/10.15388/ioi.2015.13

# Conducting Complex Competitions in Informatics with Individual Tasks

Pavel S. PANKOV, Jyldyz R. JANALIEVA
*International University of Kyrgyzstan*
*e-mail: pps50@rambler.ru, noledi@yandex.ru*

**Abstract.** Most olympiads in informatics are actually programming contests, sometimes it is stressed by the rules „tasks must be of algorithmical nature“. Meanwhile, now informatics covers almost all branches of human activity and participation in competitions with wide spectrum would be of interest to young people. To prevent participants from copying answers from others, we propose to generate individual tasks from personal data to achieve uniqueness (all contestants would receive different versions of tasks). Also, we draw attention to tasks of "output-only" and "open-ended" types (which can be also called "semi-algorithmical" because the contestant combines intuition with using algorithms, which may be known in advance or devised while solving) and optimization tasks with unknown exact answers. The main problem is automated assessment of solutions of various tasks, some techniques are proposed.

**Keywords:** complex competition, informatics, individual tasks.

## 1. Introduction

Let us retell general items of Calls for tasks, 2011–2015: "IOI tasks are typically focused on the design of efficient, correct algorithms, etc. … the submission of novel task types not yet seen in IOIs, tasks whose basic rules (if not optimal strategy) are accessible to a *wide audience* … tasks that illustrate algorithms and computational problems that arise in *a variety of human endeavours* … open-ended tasks, ones that do not necessarily have a known efficient or optimal solution … tasks that go beyond the typical format in which a program collects input, performs some computation, and returns output … tasks with some measure of solution effectiveness other than CPU time consumption … are encouraged" (emphasis by us).

But "Examples include "reactive" and "output only" tasks which have been used occasionally in previous IOIs…", that is, most IOI tasks were of the only type: to write an effective (discrete) algorithm.

We may add that informatics now covers all directions of human activity. The aim of this paper is an attempt to evolve a variety of tasks meeting more its branches and its applications.

One of such scopes was proclaimed in well-known and widely conducted *Bebras* under V. Dagiene‘s leadership:

- Information: conception of information, its representation (symbolic, numerical, graphical), encoding, encrypting.
- Algorithms: action formalization, action description according to certain rules.
- Computer systems and their application: interaction of computer components, development, common principles of program functionality, search engines…
- Structures and patterns … of discrete mathematics … combinatorics and actions with them;
- Social effect … cognitive, legal, ethical, cultural, integral aspects…
- … puzzles: logical games, mind maps, used to develop technology-based skills.

Also, another of such attempts is our conducting competitions on “applied mathematics” (new genre, as we hope) in Kyrgyzstan, examples of tasks proposed to them are marked (*) below.

To avoid cribbing and make such competitions more interesting we propose to use individually generated tasks.

Section 2 contains definitions of extended tasks and individually generated tasks.

Section 3 contains examples of tasks with simple discrete response.

Section 4 describes various types of tasks with objects outside of the examination program.

Some non-standard tasks on programming to be solved without translators are proposed in Section 5.

Tasks on interactive search of a “hidden object” are considered in Section 6.

Sections 7 and 8 apply the ideas of output-only and open-ended tasks on approximate calculations and optimization to proposed competitions.

## 2. Definitions of Extended and Individually Generated Tasks

In our observation, almost all young persons who could master programming begin to write test programs with standard “multiple choice” of some preliminary written answers. Diversification is almost always reduced to a random choice of data base with ready tasks and to a random permutation of answers. It does not give opportunity to evince programming skills and knowledge of subjects previously studied. Meanwhile, since the eighties, we as well as others proposed more flexible methods to develop software involving various peculiarities of the subjects studied, and examining various abilities of persons tested.

Besides of common requirements (Validity, Objectivity and Reliability), to improve efficiency of evaluation of knowledge and skills, we offered the following ones:

Generativity: the complete text (content) of a task must not exist before examination (competition) and must be generated randomly just before it.

Uniqueness: all examinees (contestants) must receive different versions of tasks.

Concreteness: the examinee’s (contestant’s) response must be in the form of number, word, action.

**Definition 1.** (Pankov and Janalieva, 1995). An extended task is an algorithm generating different logically correct and methodically proper tasks (of the same level of difficulty) and corresponding right answers from the initial data (randomly) chosen from finite but sufficiently large sets (ranges). Another term for such algorithms is "parameterized questions".

**Definition 2.** The number of ranges being used for generating the task in sufficient different ways is said to be dimension of the extended task.

**Definition 3.** (Janalieva, 2009). An algorithm permitting one (a teacher) to choose subsets of sets of initial data and generating different logically correct and methodically proper tasks by initial data, (randomly) chosen from these subsets is said to be an adjustable extended task.

**Definition 4.** (Janalieva, 2012). An individually generated extended task consists of two algorithms. The first algorithm composes different, logically and methodically correct tasks using personal information about a contestant. The second algorithm generates corresponding right answers using the same information.

Correspondingly, at distant competitions the contestant is to enter his/her personal data and his/her answers and the second algorithm generates right answers for checking using the same data. And same data are printed into the certificate.

We give examples of both easy and difficult tasks to present the scope of possibilities. Also, weaker participants will spend more time to solve easy tasks and obtain non-zero points; stronger participants will spend less time to solve easy tasks and have more time to solve difficult tasks.

Simple tasks in different sciences (see, for example, Pankov, 2010) are constituents of such examination too, because "…algorithms and computational problems that arise in a *variety of human endeavours*."

All the tasks in the following sections are extended ones. Randomly generated elements are denoted by italics (some of them are "primary"; others are calculated from them).

## 3. Tasks with Discrete Response

These tasks are intended for "Objective evaluation of understanding by means of discrete response" instead of traditional "Evaluation of understanding by means of multiple choices". They are relatively easy because their aim is not checking of skills, but checking of knowledge of notions.

### 3.1. *Tasks with Simple Calculations on Digits*

**Task 1.** (Notion of symmetry). What is the minimal total amount to be added to the digit(s) to obtain a symmetric pattern:

```
1.1. [Axial symmetry]          65299756
                               14388342        [answer: 6]
```

*Remark.* Traditional multiple-choice version of this task is: choose the symmetrical:

| | | | |
|---|---|---|---|
| A) 65299756<br>14388342 | B) 65299256<br>14388341 | C) 65299756<br>14388341 | D) 65299256<br>65299252 |

```
1.2. [Transitional symmetry]   65296526
                               14381436        [answer: 5]

1.3. [Rotational symmetry]     65298381
                               14389256        [answer: 4]
```

**Task 2.** (Notion of numeration). How many asterisks are here in *binary/hexadecimal* system?

```
******* **** ****
******** **** *
```

*Remark.* Random initial data are presented in such a way that it would be easier to write out the answer immediately than to count in decimal and convert.

**Task 3.** (Notion of graphical information). How many *bits/bytes/kilobytes (rounded)* are necessary to encode a 400 × 560 pixels image with *the colours: white; light grey; dark grey; black* / 16 *colours* / 256 *colours*?

**Task 4.** (Notion of error correction). 25 digits, 5 checksums (mod 10) by rows and 5 checksums (mod 10) by columns were transmitted, but one of the 35 digits was changed. Which one?

```
652980
143895
256014
558273
087623
 47617
```

Write the answer in the form: <changed digit>#<restored digit> [answer: 2#8]

**Task 5.** (Notion of directed rounding). Given: $7 \le X \le 20$; $50 \le Y \le 70$. Find the width of the narrowest closed interval with integer boundaries containing all quotients $Y/X$: [answer: 8]

**Task 6.** (Notion of chemistry). How many whole molecules of $A_4B_3C_2$ can be made of 45 atoms of A, 40 atoms of B and 30 atoms of C? [answer: 11]

### 3.2. *Tasks with Simple Calculations on "Big Pixels"*

Tasks like Task 1 also can be presented as follows:

**Task 7.** (Notion of symmetry). How many black squares, at least, are to be *added/*

*erased/shifted* to obtain a symmetric pattern (in real task, the table should be about 5 × 10 pixels):

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| □ | □ | ■ | ■ | □ | ■ | ■ | □ | □ |
| □ | ■ | □ | ■ | ■ | □ | □ | ■ | □ |
| ■ | □ | □ | □ | □ | □ | □ | □ | ■ |

**Task 8.** (Notion of connectedness). (Also, given pattern of white and black squares).

A) How many, at least, black squares are to be *added/shifted* to obtain a connected set [from non-connected one]?

B) … to be erased to make the set non-connected [from connecting one]?

### 3.3. *Interactive Tasks with a Simple Discrete Response*

Tasks of the above types can be also presented as follows:

"Change as few digits as possible to (obtain the result)";
"*Add/Erase/Shift* as few black squares as possible to (obtain the result)"

(An interactive interface is to be programmed for such actions).

## 4. Tasks with Objects Outside of the Examination Program

### 4.1. *Tasks with Real Objects Outside of the Examination Program*

**General task 9.** *(Measuring a real object). The real object contains many marks (numbered points); their coordinates are in the examination program; it is prepared and copied by the jury in advance.*

Each contestant is given a copy of the object and a ruler.

"What is the (quantitative characteristic) of the object formed by … (*random*) marks (points)?"

The examination program calculates the result by means of coordinates of these marks and compares it with the contestant's answer; some error is permitted.

Examples:

**Task 10.** (Area of a real triangle). Given a sheet with some dozens of numbered points and a ruler.

"What is the area (in square cm) of the triangle formed by point 32, point 14 and point 56?"

*Remark.* If the area is too small then the examination program picks another triple.

**Task 11.** (Geodesy). The same sheet.

"Denote the imaginary intersection of straight lines point 32 – point 14 and point 12 – point 65 as point 100. What is the length (in cm) of the segment point 32 – point 100?"

## 4.2. *Tasks with Extracting Information of Files Outside of the Examination Program*

These tasks are intended to evaluate skills in applying various existing software.

**Task 12.** (Geography). Given a file with geographical maps and there is a vast list of names of geographical points (with their coordinates) in the examination program.

"Find (very approximately) the distance (km) between *point*1 and *point*2 along the earth's surface."

The examination program calculates the distance by means of formulas of spherical trigonometry.

For the following tasks, files are formed in advance such that there is a functional relation between a file's name and its content but this relation is too complex to be guessed by the contestant by two-three examples. For instance, let the file file*N*.txt, $N = 10 .. 90$ contain the number $3980 - N * (N - 44)$.

**General task 13.** (Searching for and extracting information outside). (Find and) open the file(s) with *name(s)* and extract *information* (a *number* or a *word*).

*Examples*:

**Task 14.** (Search in text). Find the distinguished record (the least number, the longest word, etc.) in the announced (short) files, for instance d:\dir12\file5*.txt .

**Task 15.** (Search in text). Open the (vast) file Task15.doc and input the least one among all the numbers written immediately before numbers 16.

**Task 16.** (Excel). Open the file Task16.xls and find the sum of all positive numbers in the rectangular area *B*4–*F*29.

## 4.3. *Tasks with Treating Files Outside of the Examination Program*

**General task 17.** (Transforming files). Open the file …, make the following *transformation,* extract *information* and input it.

The idea is the following. Any *transformation* can be made by several means (by mouse, by keyboard, by hotkeys, etc.) and the examination program, certainly, cannot check correctness of all of them. But the file is composed in such a way that the result of *transformation* defines *information.*

*A simple example:*

**Extended task 18.** (Deleting symbols). The file Task18.txt contains

E214F5F325H43H54345G5G56H7G8G9

"Open the file Task18.txt, delete all $N$ = 1 .. 5 and input the position of $M$ = 6 .. 9." The answer is $24 - N + 2\,(M - 6)$.

*For example:* "Open the file Task18.txt, delete all "3" and input the position of "7". The answer is 23.

### 4.4. *Tasks with Creating Files*

**General task 19.** *(Creating images). Given (verbal) description of an (random) object. "Find its (continuous quantitative characteristic rounded to integer) or (discrete quantitative characteristic)."*

[Calculation of such characteristic is absent in standard software.]

The contestant is to guess that the only possible way to solve the task in given time is creating an image of the object by means of standard software and looking at it.

*Examples:*

**Task 20.** How many zeros has the function $F(X) = \sin X + 0.2\,X - 0.01\,X^2$, $0.8 \le X \le 5.7$?

**Task 21.** What is the value of the least local maximum of (such function) rounded to integer*?*

The examination program itself solves such tasks by means of (written in advance) procedure where random values are input.

## 5. Non-Standard Tasks in Programming

In contrast to subsections 4.2, 4.3 and 4.4, tasks below are intended to be solved without existing software. One of the techniques is involving too large numbers.

Tasks are given in a conventional Pascal-like language, the contestant is to guess the (very simple) syntax of this language.

**General task 22.** (Analyzing a program). Given a program. "What Input will yield given Output?"

*A simple example:*

**Task 23.** Given the program

```
M := 1000^1000; Input X; X := 2 ∗ X + M;
If X < 102 + M then X := X + 50 − M else X := X + 80 − M;
Output X.
```

What Input will yield Output 146? [answer: 48]

**General task 24.** (Graphical operations). Given some graphical operations involving objects that do not fit on the screen. How many pixels will be coloured?

*A simple example:*

**Task 25.** Given a piece of program:

```
Line (20, 2000)–(20, 5000), red.
Line (15, 2100)–(15, 2200), green.
Line (15, 2100)–(15, 2150), red.
```

How many pixels will be red? How many pixels will be green?

The following types are well-known. We mention them for completeness.

**General task 26.** (Manual execution). Given a program without input and with large intermediate results. What will the Output be?

**General task 27.** (Acceleration of program). Given a program without input working too slowly (with some embedded cycles which can be eliminated). What will the Output be?

## 6. Interactive Tasks with "Black Box"

For reactive tasks, the contestant is to write a program which will call given procedure(s) and find the answer by their responses. We propose the contestant to make queries him/herself.

*The simplest example is:*

**Task 28.** You are to detect an unknown integer between 120 and 1100. For this purpose you may guess numbers and obtain responses "Guess higher", "Guess lower", "Exactly!" If you solve the task in less than 11 queries you will get the full score.

Of interest for students waere:

**General task 29*.** You are to reach the goal in a (may be unusual, see Borubaev *et al.*, 2003) space. For this purpose you may choose the directions of your steps and obtain distances to the goal as responses.

## 7. Output-Only Tasks – Approximate Calculations

**General task 30*.** A *random* geometrical object is described. To get full score calculate its (*quantitative characteristic*) with the accuracy …

*Remark 1.* Such tasks are more interesting than logically equivalent "calculate the definite integral".

*Remark 2.* A task about the length of an arc of parabola gave the paradoxical, although predictable result. All students who knew the formula for arc length got into a tangle, but most of students, who did not know this formula, have solved the task correctly.

## 8. Output-Only and Open-Ended Tasks – Optimization

A *random* function to be optimized may be constructed in such a way that the exact answer is known but it is practically impossible to find it. Then the task is "Find the *maximum/minimum* of the function ... with the accuracy..." or "as precisely as possible".

For example, the function $F(X) = A(X - C)^4 + B(X - C)^2 + D$ after opening the brackets by the examination program where random (individual) real numbers

$A \in [5.1, 5.2]$; $B \in [2.1, 2.2]$; $C \in [5, 10]$; $D \in [10, 30]$ (argmin $F(X) = C$;
min $F(X) = D$).

Tasks with an unknown exact answer should be used in the final competition only.

Difficulty of obtaining results for different *random* functions must be equal, as in the example.

If the contestant submits an approximate answer $M < D$ then it is incorrect; else the smaller the difference $(M - D)$, the better is the answer.

(We recall the traditional procedure.) During the competition the examination program checks correctness of submitted answers only. After the competition the submitter of the best result earns the full score and other contestants who submitted correct answers earn corresponding parts of the full score.

## 8. Conclusion

We hope that implementation of tasks of types listed above will enrich competitions like *Bebras,* will distinguish young persons who have good command in informatics as whole, but are not experts in programming, and will attract more young people to informatics. Developing software for such competitions will be of interest for programmers.

## References

Pankov, P.S., Janalieva, J.R. (1995). Experience and perspectives of using UNIQTEST complex of unique tests in the learning process. *Theses of Reports of Scientific-Practical Conference Education and Science in New Geopolitical Space*. Bishkek, International University of Kyrgyzstan. (In Russian).

Borubaev, A.A., Pankov, P.S., Chekeev, A.A. (2003). Chapter 4. Constructive and computer presentations of uniform spaces. In: Borubaev, A.A. *et al.*, *Spaces Uniformed by Coverings*. Budapest. Hungarian-Kyrgyz Friendship Society.

Janalieva, J.R. (2009). Development of software for the examination on the basis of extended tasks. In: *Proceedings of the VI International Scientific-Practical Conference Intellectual Technologies in Education, Economics, Management.* Voronezh, Russia, 330–335. (In Russian).

Pankov, P.S. (2010). Real processes as sources for tasks in informatics. *Olympiads in Informatics*, 4, 95–103.

Janalieva, J.R. (2012). Conducting offline informatics olympiads with individual tasks. *Olympiads in Informatics*, 6, 170–177.

**P.S. Pankov** (1950), doctor of physical-math. sciences, prof., corr. member of Kyrgyzstan National Academy of Sciences (KR NAS), was the chairman of the jury of Bishkek City OIs (1985–2013), of Republican OIs (1987–2012), the leader of Kyrgyzstan teams at IOIs (2002–2013). Graduated from the Kyrgyz State University in 1969, is a main research worker of Institute of theoretical and applied mathematics of KR NAS, a professor of the International University of Kyrgyzstan.

**J.R. Janalieva** (1969), candidate of pedagogical sciences, conducts various competitions on mathematics, informatics and languages, including collective ones for students of Bishkek and Internet ones for students of Kyrgyzstan. Graduated from the Kyrgyz State University in 1991, she works as a docent of the International University of Kyrgyzstan.