

Concept of Algorithmic Problems for Younger Students Olympiads in Informatics

Marina S. TSVETKOVA¹, Vladimir M. KIRYUKHIN²

¹*Academy of Improvement of Professional Skill and Professional Retraining of Educators,
8 Golovinskoe Shosse, Moscow 125212, Russian Federation*

²*Dept. of Informatics and Control Processes, National Research Nuclear University "MEPhI"
31 Kashirskoe Shosse, Moscow 115409, Russian Federation
e-mail: ms-tsv@mail.ru, vkiryukh@gmail.com, vkiryukhin@mephi.ru*

Abstract. The concept of algorithmic tasks for younger students olympiads in informatics is presented in this paper. This concept is based on development of various types of algorithms with help of interactive environment of computer decision modeling based on control commands of executor. This environment creates a range of visual forms of information objects – command executors and immerses the child in the problem situation defined by executor behaviour.

Keywords. Informatics, computer science, younger students olympiads in informatics, preparation for olympiads in informatics, methods of work with talented children, developmental teaching.

1. Introduction

The concept of algorithmic problems for 5–9 year old children is based on development of various types of algorithms with help of interactive environment of computer decision modeling based on control commands of executor. The environment creates a visual imagery of vivid images of information objects – executors of commands – and immerses the child into a problematic situation determined by behavior of executors. Software implementation of solutions of such problems is carried out with help of a set of control commands of executor offered to the child. Own set of control commands for each executor is built in the interactive environment. It can be called “executor command language”. This language is not programming language, but allows to accomplish linear and selection algorithmic constructs, and loops for a particular computer executor of commands.

The child selects commands using the mouse cursor and sees the resultant reaction on the screen. The decision is saved in a personal file on the computer in the form of a command set. Wrong step solution appears with a message on the computer screen and allows to enter a correct step (reflection of the solution step). The control command set

of each executor is embedded in the interactive environment of the laboratory. It can be referred to as the “executor command language (hereon ‘ECL’)”. This is not a programming language, but enables the realisation of algorithmic constructs: linear, conditional and cyclic for a specific computer executor command.

The “Virtual Informatics Laboratory” (Fig. 1) interactive learning environment includes six types of algorithmic problems. Each laboratory has its own executors, which are described using “living” prototypes from the surrounding world. The resource is publicly available on the BINOM methodical service portal (BINOM, 2015) – official website of the publisher BINOM with a free suggested materials in the Russian: an archive file of the product, manual for working with the product and examples screen for solution steps (BINOM, 2008), video lecture for the teacher, the methodical book for teachers (Tsvetkova and Kuris, 2008). Program implementation of solutions to these tasks is carried out by the child’s choice of executor control commands on the computer screen.

Each laboratory is geared towards a particular algorithmic problem:

1. Overflows (simple run-through).
2. River crossings (conditional selection).
3. Weightings (run-through with choice).
4. Black Boxes (determination of an algorithm according to known input and output data).
5. Rearrangements (run-through with limitations).
6. Passings (work with stacks).

Each group of problems comes in three levels of complexity set by the input data. Methodical description of all tasks in all the laboratories is found in the book (BINOM, 2015). The “Virtual Informatics Laboratory” learning environment can be used in various plans:

- Demonstrating the solution to a few of the problems.
- Individual training work with a selection of problems in a chosen group.



Fig. 1. The “Virtual Informatics Laboratory” interactive learning environment.

- A competition to put together a packet of problems (one from each group), against time.

The learning environment informs the student of:

- A correct solution.
- An error in the solution.
- The optimal solution.

The system awards points to students for their solutions, depending on how they carry out the tasks:

- 0 points – erroneous or absent solution.
- 1 point – correct solution, but not using the optimal number of operations.
- 2 points – optimal solution.

2. Tasks Group “Overflows”

Overflow tasks help children virtually use concepts like “container capacity”, “units of measuring volume” and “part and whole”; and to model solutions using several containers. Solving problems of this type facilitates more intensive logical thinking while running through possible options.

The “overflow” group provides several types of tasks:

- **Type 1 tasks.** Open system. Unlimited supply of water. The source (S) and terminus (T) are active in this kind of task. There are 6 levels of complexity.
- **Type 2 tasks.** Locked system. Dividing a liquid into container(s) using additional containers. Source (S) and terminus (T) are not used in this task.

Typical ECL commands:

```
< fill > : [X], [n]
< transfer > : [X] , [Y] , [n]
< drain > : [X], [n]
```

Explanation:

X, Y – denote the capacity of (A, B...) specified capacity in liters

n – the number of litres Example the screen for solution steps “Overflows”

Example task “Overflows”

You have two watering cans: one empty with a capacity of 3 litres, and one full with a capacity of 5 litres. How can you obtain 7 litres of water from the tap?

The solution as an algorithm.

start :

1. < transfer >: [B], [A], [3]
2. < drain >: [A], [T], [3]
3. < transfer >: [B], [A], [2]
4. < fill >: [S], [B], [5]

End

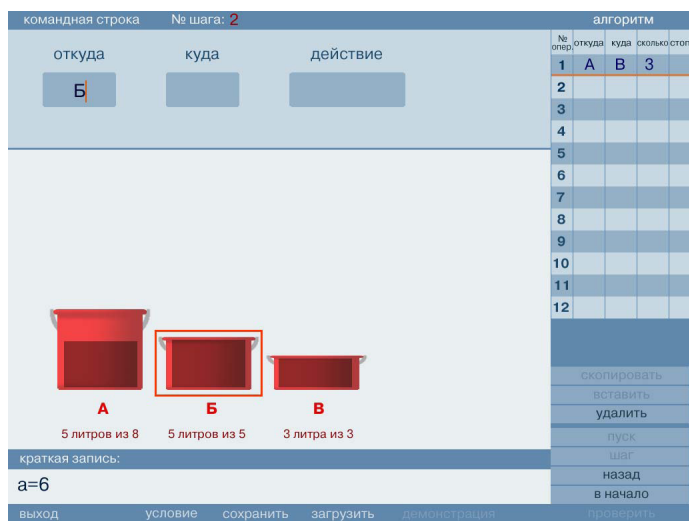


Fig. 2. Examples screen for solution steps “Overflows”.

3. Tasks Group “River crossings”

Performing river-crossing tasks enables a child, working using common sense (the relationship between various crossing participants), including spatial awareness and intuitive reasoning, to plan a sequence of actions and write an algorithm, taking into account the complexity of the task. The difficulty of river-crossing tasks lies in the load-bearing limitations of the raft, and the quantity and peculiarities of the passengers (their possible combinations).

Typical ECL commands:

- < get on > : [X]
- < get on > : [empty]
- < cross > [A], [B]
- < get off > : [Y, ...]
- < repeat > Point1, Point2, n

Explanation:

- X – the passengers for boarding
- Y – the passengers for the landing
- A, B – the pointers of the banks
- Point1, Point2 – the number of commands to repeat
- n – the number of repeats

Example task “The goat and her seven kids”

A goat and her seven kids are going to grandma’s house on the other side of the river, on the right-hand bank. The goat does not want her kids to cross without her, but the raft can carry no more than 2 passengers. How can they all cross the river in the most rational manner?

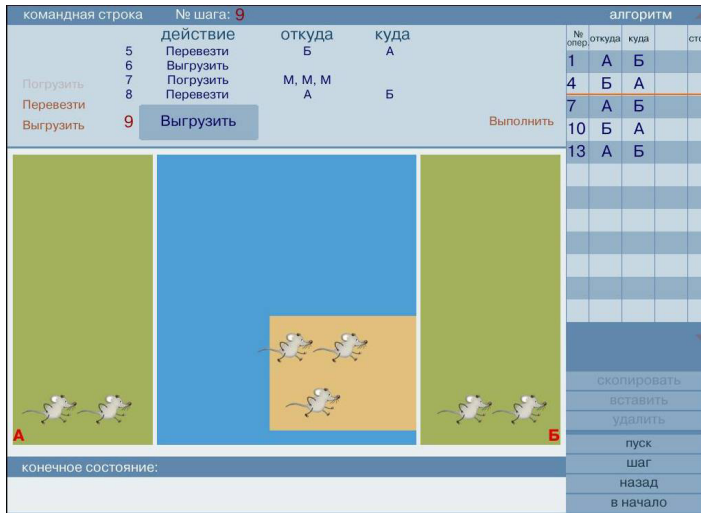


Fig. 3. Example the screen for solution steps “River crossings”.

Representation of the solution as an algorithm.

start:

1. < get on > : [Kid]
 2. < cross > : [1] , [2]
 3. < get off > : [Kid]
 4. < cross > : [2] , [1]
 5. < repeat > : commands 1,2,3,4 while > 1 kid remains on the left-hand bank.
 6. < get on > : [Kid]
 7. < cross > : [1] , [2]
 8. < get off > : [Goat].
- end

4. Tasks Group “Weightings”

Weightings tasks teach children to select efficient sequences of actions using the relationships “lighter” and “heavier”. Comparison is a logical approach used to establish the similarities or differences between objects. The result of a comparison is described by the terms “equal”, “larger” or “smaller”, depending on the relationship of objects for comparison.

In tasks of this type, not only single objects are compared but also groups of objects. This forces the child to analyse his or her actions and “think through” all possible combinations to find a solution.

Typical ECL commands:

< compare >: [x] , [y] ,

```

If [x] = [y] ,
Elseif [x] > [y] ,
Endif ,
< choose > : [X]

```

Explanation:

X, Y – objects for weighing

Example task “Fake coins 4”.

A merchant has 6 coins, 1 one of which is false and lighter than the others. Suggest a way of finding the fake coin with the minimum number of weighs on a set of scales. The real coins are all of equal weight.

Representation of the solution as an algorithm:

start:

```

1. <compare>: [1,2] , [3,4]
2. if [1,2] = [3,4]
3. then < choose > [5,6]
4. <compare>: [5] , [6]
5. if [5]>[6]
6. then < choose > [6]
7. else < choose > [5]
endif

```

else

```

2. if [1,2] > [3,4]
3. then < choose > [3] , [4]
4. <compare>: [3] , [4]
5. if [3]>[4]
6. then < choose > [4]

```

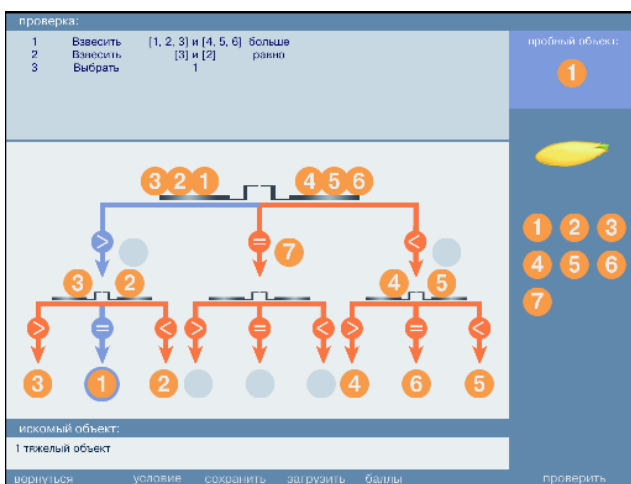


Fig. 4. Example the screen for solution steps “Weightings”.

```

7. else <choose> [3]
endif

else
2. if [1,2] < [3,4]
3. then < choose > [1], [2]
4. <compare>: [1], [2]
5. if [1] > [2]
6. then <choose> [2]
7. else <choose> [1]
endif
    
```

5. Tasks Group “Black Boxes” (Mathematical Operations with Numbers)

The reinforcement of mathematical principles in relation to the educational standard:

- . Calculating operations
- . Comparison operations
- . Order of actions
- . Number composition

Primary recording of arithmetical and logical expressions, order of calculations in arithmetical expressions, modelling mathematical records of number composition tasks.

ECL description.

take <x> – take some value of X

<operation> – perform an operation according to the formula present

Brackets: ()

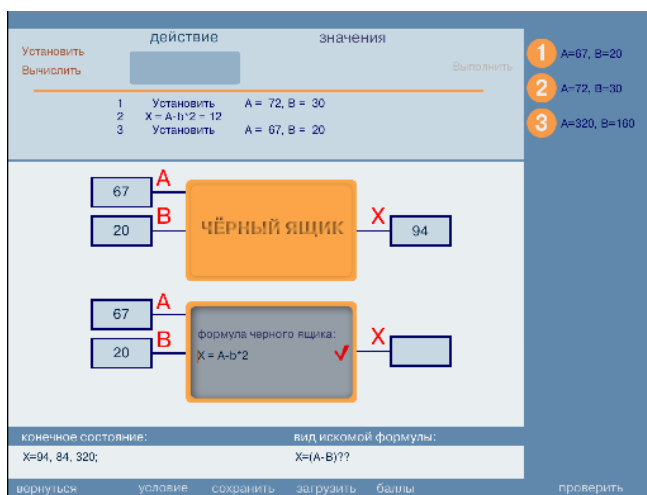


Fig.5. Example the screen for solution steps “Black Boxes”.

< Number composition >

A <comparison operation> B

Only two-way comparisons are used in this group of operations, but there could be formulae on either side of the comparison.

Explanation:

X – expression

A, B,.. – operands

Examples of task conditions.

«Scattered numbers»

At the entrance of the Black Box is the number a, comprised of 7 tens, 3 hundreds and 5 units.

Complete the formula using the information of the Black Box, the number 10 and multiplication and addition.

a	=												
---	---	--	--	--	--	--	--	--	--	--	--	--	--

Some cells may remain empty if they are not necessary for the result, and this will not affect the accuracy of the answer.

The Black Box will calculate the number according to the formula and say whether it is correct or not.

“Compare”

At the entrance to the Black Box there are 3 mathematical expressions.

It compares their results and completes the table using comparison operations.

Complete the table for the mathematical expressions provided by the Black Box. Put “>”, “<” (change “?”)

$$25 \times 5 \ ? \ 100 - 25 \ ? \ 4 \times 25$$

$$14 \times 14 \ ? \ 14 + 14 \ ? \ 2 + 14$$

$$18 \times 3 \ ? \ 9 \times 6 \ ? \ 3 \times 18$$

The Black Box will check it and say whether the task has been solved correctly.

“The secret of the brackets»

The magic Black Box can calculate quickly. It inserts brackets in different places in identical mathematical expressions to get different answers.

Put the brackets in grey cell. Using the results the Black Box shows you, correctly insert brackets into the mathematical expressions.

13	=	5	×	(9	-	7)	+	3	
----	---	---	---	---	---	---	---	---	---	---	--

25	=	5	×	(9	-	7		+	3)
----	---	---	---	---	---	---	---	--	---	---	---

35	=	5	×		9	-	(7		+	3)
----	---	---	---	--	---	---	---	---	--	---	---	---

6. Tasks Group “Rearrangement”

The idea behind rearrangement tasks is that their solutions require a clear order of actions limited by strict conditions. The tedious task of gradually moving objects teaches a child to think about consequences, analyse every action and build a strategy. It involves the transfer of objects from one position to another. Only the uppermost object is ever accessible. A rearrangement is only performed if the movement conforms to the conditions set regarding the size of the objects.

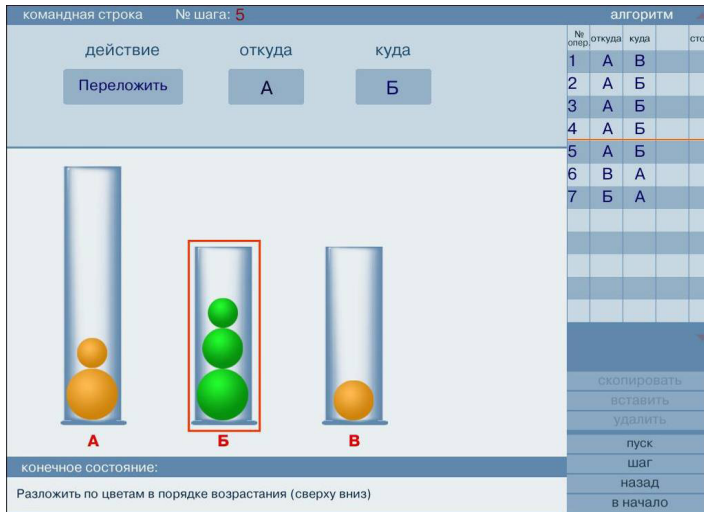


Fig. 6. Example the screen for solution steps “Rearrangement”.

Typical ECL commands:

< move > [A] , [B]

Explanation:

A – tube (rod) to take,
B – tube (rod) to put.

Example task “Coloured balls 2”.

There are three test-tubes and three green, three red and three yellow balls (Fig. 7). The balls are stacked one upon the other in the first test-tube. The balls enter the test-tube one-by-one from the top. With the help of the other two test-tubes, arrange the balls by colour.

Representation of the solution as an algorithm:

- ```

start :
1. < move > : [a] , [b]
2. < move > : [a] , [b]
3. < move > : [a] , [c]
4. < move > : [a] , [b]
5. < move > : [a] , [b]
6. < move > : [a] , [c]
7. < move > : [a] , [b]
8. < move > : [a] , [c]
9. < move > : [b] , [c]
10. < move > : [b] , [a]
11. < move > : [b] , [a]
12. < move > : [c] , [b]
end

```

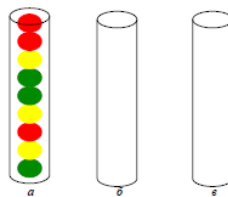


Fig. 7. Example task “Coloured balls 2”.

## 7. Tasks Group “Passings”

Passing tasks are about finding the optimal solution. Many things in the task can change: the motion trajectory, quantity, load-bearing capabilities and peculiarities of a raft, the quantity and quality of variable objects, the presence of additional conditions increasing the number of methods of solving them. In the proposed tasks there is a stack in the form of a ferry with an open entrance and open exit or a stack in the form of a barge with an open entrance and closed exit.

There is a spare stack or two stacks – a ferry or a barge in a river branch – with entrances and exits accordingly, but as this stack can stockpile a certain quantity of objects, we consider it unlimited in terms of objects stockpiled.

Several types of tasks can be provided for solving.

- **Type 1 tasks.** There are two piers situated on one bank. Traffic travels on a ferry from the source pier to the terminal pier. There is a spare ferry in the river branch.
- **Type 2 tasks.** There are three piers situated on one bank. Traffic travels on a ferry from the source pier to the terminal pier. There is a spare ferry in the river branch.
- **Type 3 tasks.** There are two piers situated on one bank. Traffic travels on a barge from the source pier to the terminal pier. There is a spare barge in the river branch.

### Typical executor ECL commands:

< load > : [n]

< cross > : [A] , [B] ([C])

< unload > : [n]

(n is the number of a car in the queue).

| командная строка | № шага: 14 | действие  | откуда | куда | алгоритм |        |      |      |
|------------------|------------|-----------|--------|------|----------|--------|------|------|
|                  | 10         | Перевезти | А      | Б    | №        | откуда | куда | стоп |
| Погрузить        | 11         | Выгрузить |        |      | 1        | А      | 3-1  | 1    |
| Перевезти        | 12         | Перевезти | Б      | А    | 5        | А      | 3-1  | 2    |
| Выгрузить        | 13         | Погрузить |        |      | 9        | А      | Б 3  |      |
|                  | 14         |           |        |      |          |        |      |      |

конечное состояние:  
Синяя, Зеленая, Белая, Синяя, Зеленая, Белая, Синяя, Зеленая, Белая

Fig.8. Example the screen for solution steps “Passings”.

**Explanation:**

[A] , [B] ([C]) – designations of the banks and emergency stationary ferry on the river,  
 n – the number of the car.

**Example task “Crossing 2: 5 cars”.**

A ferry travels from pier A to pier B, which are both situated on one side of the river. Using the ferry it is possible to transport one car at a time. The river also has a branch where a spare ferry Z is waiting. Cars can enter and exit both ferries from either side. Cars must be unloaded at pier B in a strict order: red – blue – white – red... Cars are waiting for the ferry at pier A in a queue of this order: blue – red – blue – red – white... Transport 5 cars from pier A to pier B using the ferry and meeting the conditions. The cars are numbers 1 to 5.

Representation of the solution as an algorithm.

```

start:
1. < load > : [1]
2. < cross > : [A] , [Z]
3. < unload > : [1]
4. < cross > : [Z] , [A]
5. < load > : [2]
6. < cross > : [A] , [B]
7. < unload > : [2]
8. < cross > : [B] , [A]
9. < load > : [3]
10. < cross > : [A] , [B]
11. < unload > : [3]
12. < cross > : [B] , [A]
13. < load > : [4]
14. < cross > : [A] , [Z]
15. < unload > : [4]
16. < cross > : [Z] , [A]
17. < load > : [5]
18. < cross > : [A] , [B]
19. < unload > : [5]
20. < cross > : [B] , [Z]
21. < load > : [4]
22. < cross > : [Z] , [B]
23. < unload > : [4]
24. < cross > : [B] , [Z]
25. < load > : [1]
26. < cross > : [Z] , [B]
27. < unload > : [1]
end

```

**8. Conclusion**

The virtual laboratory learning environment has a task editing plan for teachers, enabling them to edit their own tasks and use them in their chosen laboratory. Algorithmic problems are a part of informatics tuition at primary school (Tsvetkova and Kuris, 2008; Kuris and Tsvetkova, 2013) and form the basis for developing algorithmic thinking faculties in younger students. The concept the algorithmic problems Virtual Laboratories in Informatics as learning environment can be used for algorithm competitions and train to Olympiad for children interested in informatics at a school additional lessons (Kuris and Tsvetkova, 2013; Kiryukhin and Tsvetkova, 2014). It should be noted that these tasks are other examples of implementation: “Robotlandia” (ROBOTLANDIA, 2015), some tasks for students with grades 1–6 from the collection of the International contest in Informatics “Bebras” (Bebras, 2015).

## References

- BINOM (2015). *Educational Software Product "Virtual Laboratories in Informatics"*. (In Russian). Reference on 20.08.2015:  
<http://metodist.lbz.ru/authors/informatika/5/ep-4-umk3-4fgos.php>
- BINOM (2008). *Manual for Working with Virtual Laboratories in Informatics*. (In Russian). Reference on 20.08.2015: <http://metodist.lbz.ru/iukm/informatics/files/manual.pdf>
- Tsvetkova, M.S., Kuris, G.E. (2008). *Virtual Laboratories in Informatics at Primary School: methodological manual*. BINOM. Laboratory of knowledge, Moscow. (In Russian). Reference on 20.08.2015:  
<http://lbz.ru/books/264/5211/>
- Kuris, G.E., Tsvetkova, M.S. (2013). *Informatics. Teaching Materials for Primary School: Grades 3–4. Methodological Manual for a Teacher* (public electronic version). BINOM. Laboratory of knowledge, Moscow. (In Russian). Reference on 20.08.2015: <http://files.lbz.ru/pdf/mpMogilev3-4fgos.pdf>
- Kiryukhin, V.M., Tsvetkova, M.S. (2014). *Informatics Plans of Olympiads Training Students: Grades 5–11. Methodological Manual for a Teacher*. BINOM. Laboratory of knowledge, Moscow. (In Russian).
- ROBOTLANDIA (2015). *Educational Software Product "Robotlandia"* (In Russian). Reference on 20.08.2015:  
<http://www.botik.ru/~robot/sale/robot.htm>
- Bebras (2015). *International Contest in Informatics "Bebras"* (In Russian). Reference on 20.08.2015:  
[http://bebras.ru/bebras15/domain\\_contests](http://bebras.ru/bebras15/domain_contests)



**M.S. Tsvetkova**, professor of the Russian Academy of Natural Sciences, PhD in pedagogic science, prize-winner of competition “The Teacher of Year of Moscow” (1998), main expert of state projects of school education informatization in the Ministry of Education of the Russian Federation (2001–2005), the expert of the World bank project “Informatization of Education System”. Since 2002 she is a member of the Central methodical commission of the Russian Olympiad in informatics, the pedagogic coach of the Russian team on the IOI. She is the author of many papers and books in Russia on the informatization of education and methods of development of talented students. Since 2013 she is the Russian team leader.



**V.M. Kiryukhin** is professor of the Russian Academy of Natural Sciences. He is the chairman of the federal methodical commission on informatics which is responsible in Russia for carrying out the national Olympiads in informatics. He is the author of many papers and books in Russia on development of Olympiad movements in informatics and preparations for the Olympiads in informatics. He is the exclusive representative who took part at all IOI from 1989 as a member of the IOI International Committee (1989–1992, 1999–2002, 2013–2016) and as the Russian team leader (1989, 1993–1998, 2003–2012). He received the IOI Distinguished Service Award at IOI 2003, the IOI Distinguished Service Award at IOI 2008 as one of the founders of the IOI making his long term distinguished service to the IOI from 1989 to 2008 and the medal “20 Years since the First International Olympiad in Informatics” at the IOI 2009.