Measuring the Algorithmic Skills of Students Working with Low- and High-Mathability Programming Approaches

Katalin Sebestyén[a], Gábor Csapó[a], Mária Csernoch[b], Kálmán Abari[c]

[a]University of Debrecen, Doctoral School of Informatics
Kassai út 26, H-4028 Debrecen, Hungary
{sebestyén.katalin, csapo.gabor}@inf.unideb.hu

[b]University of Debrecen, Faculty of Informatics
Kassai út 26, H-4028 Debrecen, Hungary
csernoch.maria@inf.unideb.hu

[c]University of Debrecen, Faculty of Arts and Humanities
Egyetem tér 1, H-4032 Debrecen, Hungary
abari.kalman@arts.unideb.hu

Abstract: In spite of the fact that teaching programming is obligatory in Hungarian public ICT education, the low number of lessons accompanied by the lack of students' knowledge and interest marginalizes the topic. Research and experience show that even when working with experienced teachers, students have a hard time mastering imperative and object-oriented programming languages. These languages usually approach problem-solving from a mathematical perspective, with a minimal design IDE (Integrated Development Environment) and output screen. As an alternative solution, schools and courses apply various visual programming environments that make it possible to create colorful motivating games and animations even in one lesson. In our research, we compared two visual programming environments: Scratch (control group), developed for education, and Construct 3 (experimental group), developed for game and software development. We conducted measurement of the efficacy of the two environments for teaching programming in two grade-8 groups. The students learned the topic by solving traditional algorithmic tasks but taking advantage of the visual interfaces. The results, in accordance with previous findings, show that in developing the students' algorithmic skills there is no difference between these visual programming languages. Furthermore, we found proof that the selected teaching methods play a crucial role in the development of said skills of the students.

Keywords: visual programming; algorithmic skills; Bebras; ICT education; programming education; Construct 3; Scratch

1 Introduction

The process of developing students' computational thinking and algorithmic skills is restricted to the programming topic of ICT (Information and Communications Technology) education, according to the Hungarian Base Curricula [1] and Frame Curricula [2]. The low number of lessons, the ambiguous requirements of the Frame Curricula [3], and the outdated programming knowledge of ICT teachers marginalize the programming topic. This not only means that programming receives much less emphasis on the subject than was originally intended, but in several cases the topic is completely ignored or the focus is strictly on the tools, such as Scratch and robots instead of programming.

To teach programming in ICT education, several programming approaches and environments are widely accepted. The modern object-oriented languages (for example C++, C#, Python and Java) are present, but students can also encounter procedural languages which are outdated and are rarely used in the industry (like Pascal). Learning programming is difficult and challenging for beginners, considering both the problem-solving aspect and the syntactical rules of these high-level programming languages [3] [5] [6]. To make programming education more effective and more easily understandable for students, several educational programming languages (EPLs) have been introduced and have become widely accepted in the meantime. These languages usually take a different approach to create the code compared to text-based languages. [7] [8] [9] [10] [11] [12]. However, despite these attempts, students at the end of their secondary education do not possess the required level of algorithmic skills even for solving simple problems [12].

2 Visual Programming

In visual programming, students use pre-defined graphical language elements to construct the code of the program. This process emphasizes the building of algorithms without the burden of syntactical rules. The widespread use of this programming method in the industry is supported by several programming environments and game engines alongside the text-based options [14] [15] [16]. The visual representation of codes and the rapid developmental experience make visual programming languages compelling choices for the educational field, as well [17] [18].

It is worth noting that the various visual languages are not compatible with each other: a code created in one environment cannot be transferred directly to other environments. Analyzing the visual programming languages present in education and industry, we can define four categories based on their concepts [19] [20]:

– behavior-based,

- event-action based,
- block-based,
- node-based.

2.1 Scratch

The well-known Scratch, block-based visual programming environment is present at all levels of ICT education [11] [21]. It is primarily designed for beginner programmers, especially for students who could not imagine themselves as programmers before working with Scratch [11]. The environment includes several components designed for education (for example the ability to share projects or the public availability of project source codes). Despite the fact that students find this environment easy to use, several studies have encountered problems in terms of its effectiveness. Testing grade-5 students in a primary school, Kalelioglu & Gülbahar found that focusing on the spatial-visual aspect of Scratch did not increase problem-solving skills compared to traditional approaches [22]. Students tend to develop and follow bad programming habits while working with Scratch. For example, they include all blocks in a program code that might be needed or might be connected to the problem, without analyzing the original problem and the tools available. Additionally, students tend to over-deconstruct problems without logical coherence between the elements [23]. It is also important to emphasize that Scratch does not reinitialize variables upon re-executing a project. This leads to bad initialization habits and makes knowledge-transfer to further programming environments more difficult [24].

2.2 Construct 3

Our research group selected the HTML5 based general-purpose Construct 3 environment [25] which uses an implementation of the event-action-based and behavior-based visual programming approaches. With Construct 3, students can create simple 2D games and multimedia web applications quickly and easily. Consequently, the environment fulfills all the requirements of teaching fundamental programming concepts.

Construct 3 provides a complete graphical interface for creating projects. The users work with objects that are placed on layouts representing how the application will look visually on screen. These objects are pre-made elements that cover different functionalities of a project (like displaying a sprite or playing a sound). The environment includes various pre-programmed behaviors (for example different movement algorithms) which can be attached to objects. For creating custom logic, the users can use event-sheets to build up the visual code connecting actions to events referencing the objects. Further details on the workflow of the Construct environment are presented in our previous paper [19].

Construct 3 is well-documented, which aids its integration into classes [26] [27] [28] [29] [30]. To further support the education processes, the environment provides an option, similarly to Scratch, to publicly share students' work online. The free version of the environment can only be used with limitations. However, the capabilities of the free version are sufficient to teach programming in elementary and secondary ICT education.

2.3 Helping Materials and Tasks

Similar to other topics in ICT education [31] [32] [33], programming is also exposed to erroneous tasks, algorithms, and source codes that are built up without deeper understanding or logic. In general, low-mathability approaches are applied in the teaching-learning process, where the focus is on the tools instead of the problem [34].

Even the environments designed for educational contexts are burdened with such tasks. Figure 1 presents the source code of a Scratch task [34] which writes out the numbers divisible by five in the range of 100-150. The source code is based on several unusual solutions: for no apparent reason, the program only runs when the user presses the "o" character, and it uses an infinite loop (forever command) to write out the numbers which is only stopped by a separate if statement. This could be interpreted as a do-while loop; however, the task description in [34] includes no information on this knowledge item whatsoever.

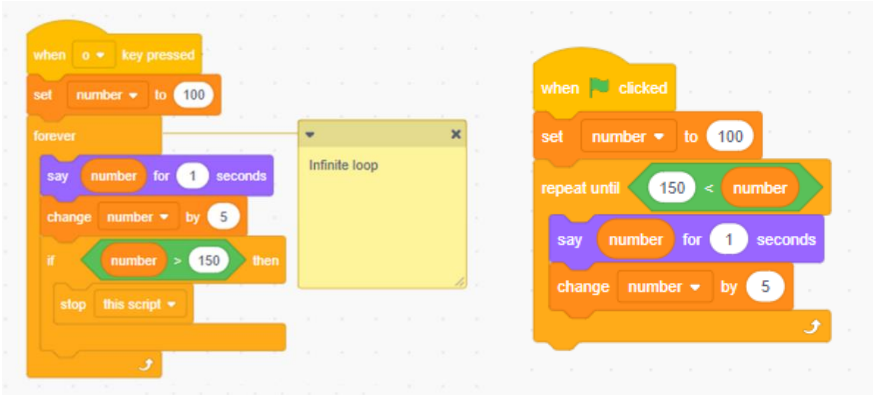


Figure 1

An example task in the Scratch environment for writing outnumbers divisible by five in the range 100–150 [34] (left) The modified version of the original source code (right) (The source codes are the English translation of the original Hungarian version)

We selected this task to highlight the erroneous problem-solving strategies present in ICT education. Note that the task presented and the errors included in it do not represent the whole world of ICT, as one can find educational materials and tasks of outstanding quality, just as there are ICT teachers who are accurate and

professional (expert teachers) **Error! Reference source not found.** [36]. In general, we can conclude that it is time to rethink, correct, and transform ineffective practices, and that this is in the interest of all parties involved in education.

2.4 Our Goals

In this paper, we present an analysis of the effectiveness of the Scratch and Construct 3 environments' development of students' algorithmic skills. Our goal was to examine how different visual programming approaches affect the development of said skills by solving tasks with methods that focus on problem-solving – high mathability [38] – instead of on the graphical interfaces and visual capabilities of the environments – low mathability [34] [38] [39] [41].

3 Applied Methods, Tools, and Strategies

3.1 Research Environment

We analyzed the effectiveness of the methods used in programming education in two grade-8 groups in a local high school (experimental and control groups). The students took part in the 6-year training program of the school and during their previous education, they had not encountered the programming topic. Both groups progressed and learned the knowledge items at the same pace. The experimental group used Construct 3, while the control group learned with the Scratch environment. During the teaching period, we aimed to minimalize the differences between the two environments by developing customized tasks. Therefore, both groups followed the same schedule and worked on the same tasks optimized for their environment. The experimental group studied the topic for 18 lessons, while the control group for 17 lessons, one lesson per week.

3.2 Applied Methods

The main roles of the first tasks were that the students could finish simple projects and get feedback on their work, and additionally, learn the fundamentals of creating projects in their environment. After the groups grasped the essence of the interface, and it did not hinder the real problem-solving process, the complexity of the tasks was increased.

To solve the tasks, the students were guided by the coaching method [42], based on Pólya's [42] concept-based problem-solving approach:

1) Presenting the task.

2) Understanding the problem: Analyzing the complete task and decomposing it into subtasks.
3) Setting the goal of the subtasks. Highlighting the input and output values of the subtasks.
4) Building the algorithm.
5) Precoding in natural language: While following the structure of the visual programming language, phrasing the conditions and statements required to code.
6) Coding.
7) Testing: Running the code and discussing the outputs.
8) Debugging and correcting the errors.
9) Combining the subtasks: debugging, discussion, abstraction.

Note, that designing tasks for visual programming environments involves additional factors that teachers must take into consideration. Besides the creation of the project and the algorithms included in it, the visual design of the project also plays an important role in the success of each task. During our work, besides minimalizing the differences originating from the two environments, we also focused on avoiding error prone approaches). The teaching of both the experimental and control groups was carried out by the same teacher from our research group.

The problems, the students worked on were presented in visually engaging smaller projects. At the beginning of these projects, the teacher presented the complete work and started a discussion following the aforementioned approach. The projects were then decomposed into smaller subtasks which the groups analyzed further, before starting the development process. For example, in one of the projects the groups created a game with randomly appearing targets and a sling that could shoot projectiles. The students analyzed the problem both from the point of the required objects and assets and from the point of the algorithms behind the behavior of these elements. During the teaching period, the following projects were created:

- crossing the street (simple movement, collision detection, handling variables, outputting text referencing variables, and random number generation),
- target practice (handling user input, creating advanced logic),
- UFO attack (exercise task).

3.3 Data Collection

We collected data in pre- and post-tests, using paper-based test sheets. The pre-test was carried out before the first programming lesson to avoid affecting the prior knowledge of the students. The post-test was administered during the lesson following the last programming class.

Table 1
The number of students in the experimental and control groups

	experimental group	**control group**
number of students	14	14
pre-test	10	13
post-test	12	13
paired tests	7	12

Based on the Frame Curricula and the local curricula of the school, the programming topic only appears in grade-8 and grade-10 classes. In the school year of the measurement, only one grade-8 class was enrolled, and the class was divided into two groups.

The low number of students in both groups made it possible to ensure a similar progression in the topic (Table 1). The fluctuation in the number of students (especially considering the paired tests in the experimental group) can be explained by the students' absences and various school activities.

3.4 Tests

To measure the algorithmic skills of the students, we composed a test that relies on Bebras tasks [44] [45], making it independent of the programming environment. We analyzed and selected the Bebras tasks we included to make them cover several aspects of the algorithmic skills, requiring differing thinking processes and strategies to solve. The selected tasks were not focused directly on the knowledge items present in the problems solved. Instead, our goal was to measure the algorithmic and problem-solving skills developed during the teaching-learning period. It is important to note that we found differences between the original tasks [46] and the Hungarian versions we used [44]. However, these differences are minimal and have no effect on the skills measured. We changed several multiple-choice tasks to open-answer tasks so that students did not have predefined options from which they could select one randomly if they could not solve the problem. Following these conditions, the testing process included these altered tasks (Appendix, section 0).

3.5 Analyzing the Data

To store the data collected with the tests, we created a database where the tasks were decomposed into items. Due to the particularities of the tasks, the student answers were on a narrow scale as several tasks required a character or number as the answer, despite the open questions. Therefore, the decomposition of the students' answers into items was limited.

POPULARITY (2015-CA-01)

Seven beavers are in an online social network called Instadam. Instadam only allows them to see the photos on their own and their friends' page. In this diagram, if two beavers are friends they are joined by a line. After the supper holiday everybody posts a picture of themselves on all of their friends' page.

Which beavers' picture will be seen the most?

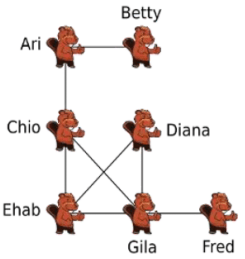


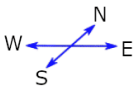
Figure 2
The popularity task (2015-CA-01) [45] [46]

During the analysis of the data, we stored additional information on the students' answers where the tasks justified it. The *popularity* task is an example of this, where a frequently chosen answer was the character who seemingly has the most connections (Figure 2).

In this case, we not only marked the answer incorrect but grouped it into a separate category to make it possible to observe the differences in how students applied fast-thinking [47]. Similarly, in the *spherical robot* task (Figure 3), we also separated those answers which guided the ball into the goal with unnecessary instructions from those that completed the task perfectly.

SPHERICAL ROBOT (2016-JP-03)

The BeaverBall is a toy that can be operated by remote control, and understands each of four direction commands.



If the BeaverBall moves to a white square, it drops down one level. The BeaverBall ignores commands that cause it to move outside the borders.

Look at the position of the BeaverBall in the picture above. Which of the following lists of directions will cause the BeaverBall to reach the GOAL?

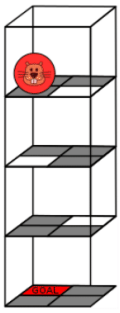


Figure 3
The spherical robot task (2016-JP-03) [44] [46]

4 Analyzing the Effectiveness of Programming Tools

Table 1
The results (%) of the pre- and post-tests in the experimental (exp.) and control groups by task, along with the p-values of the differences between the groups

Tasks	Pre-test		p	Post-test		p
	exp.	control		exp.	control	
rotating puzzle	90.00	69.23	0.226	83.33	84.62	0.934
popularity	70.00	38.46	0.030	70.83	53.85	0.059
beaver code	100.00	100.00		100.00	100.00	
party guests	70.00	63.08	0.718	83.33	87.69	0.752
hierarchy	80.00	84.62	0.784	91.67	100.00	0.339
spherical robot	50.00	76.92	0.195	91.67	84.62	0.606
deactivatin	80.00	23.08	0.005	66.67	69.26	0.896
concurrent directions	80.00	53.85	0.197	91.67	86.92	0.329
four errands	65.00	84.62	0.254	100.00	86.54	0.047
kix code	22.50	42.31	0.292	89.58	65.38	0.130
blossom	46.00	81.54	0.074	91.67	98.46	0.413
total	59.61	65.97	0.429	88.46	83.13	0.396

To measure the effectiveness of the two programming interfaces (experimental group: Construct 3, control group: Scratch) in developing the algorithmic skills of students, we analyzed the results of the pre- and post-tests (Table 1). At this step of the research, we included the results of all students, regardless of whether they were present at both the pre- and the post-tests. For analyzing the data and the difference between the groups, we used the SPSS software package [48].

Considering the results of the pre-test, both groups provided a high proportion of correct answers. This meant that no significant differences could be found between the experimental and control groups ($p = 0.429$), except for two tasks: *popularity* ($p = 0.030$) and *deactivatin* ($p = 0.005$).

Popularity: The experimental group completed the task more successfully, with 70%, while the control group achieved 38.46%. A high proportion of students marked the incorrect option, i.e. “Gila”, who has the most direct connections. This implies that they answered the task by relying on their first impression without further analyzing the problem in hyper-attention mode [49] [50] (Figure 2).

Deactivatin: Students giving incorrect answers mostly marked jars D or E, which can be explained by a shallow interpretation of the task, again applying the hyper-attention mode [49] [50] (Figure 4).

DEACTIVATIN (2016-HU-06)

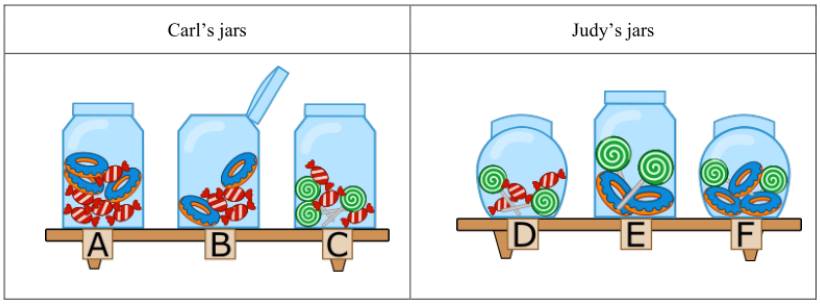
Beavers Carl and Judy have 3 jars of candies each.

Each jar can have several of the following 10 properties:

- It's either open (1) or closed (2).
- It either contains red candies with white stripes (3) or not (4).
- It either contains blue sugar donuts (5) or not (6).
- It either contains green spiral lollypops (7) or not (8).
- Its form is either round (9) or angular (10).

For example Carl's "A" jar has the following properties: 2, 3, 5, 8 and 10.

You can notice that Carl's jars have got some common properties just like Judy's jars.



Which jar has both the properties that are common among Carl's jars and Judy's jars separately?

Figure 4
The deactivatin task (2016-HU-06) [44] [46]

Considering the results of the post-test, the groups completed the test with similar rates of success, without significant differences ($p = 0.396$). Analyzing the tasks, only the *four errands* task shows a significant difference between the experimental and control groups ($p = 0.047$).

Examining the results of both the pre- and post-tests, both groups show significant development in the topic (experimental group: $p = 0.002$; control group: $p = 0.022$). Furthermore, in the post-test the experimental and control groups showed no significant difference in solving the Bebras tasks [44] [46]. This allows us to conclude that the development of the students' algorithmic skills is independent of the visual programming environment.

Conclusion

In this paper, we presented research on how an algorithm-driven teaching method in different visual programming environments and languages develops the students' algorithmic skills, and what differences can be observed between the groups. The sample included two grade-8 student groups who studied the programming topic at a similar pace. The experimental group studied with Construct 3, while the control group with Scratch. Both groups had the same teacher and worked on the same programming problems using a concept-based [42], coaching method [42].

The data collection was carried out applying printed test sheets composed of Bebras tasks. Considering the analysis of the data (Table 1) both groups completed both tests at similar levels, without significant differences.

Based on the results, the Construct 3 environment created for software production does not develop the students' algorithmic skills more effectively than the Scratch environment designed for educational purposes. However, both environments develop the algorithmic skills of students significantly when the tasks focus on problem-solving, instead of on the graphical, drawing functions of the programs. The results show that while the used software environment does not have an effect on the development of the students' algorithmic skills, the method and/or the teaching approach applied to **Error! Reference source not found.**.

Acknowledgements

This work was supported by the construction EFOP-3.6.3-VEKOP-16-2017-00002. The project was supported by the European Union, co-financed by the European Social Fund.

References

[1] NAT 2012: 110/2012 (VI. 4.) Government regulation on the publication, introduction and application of the Base Curricula, 2012, In Hungarian: Nemzeti Alaptanterv, 2012 [Online] Available: http://ofi.hu/sites/default/files/attachments/mk_nat_20121.pdf [Accessed: 04-Nov-2019]

[2] OFI: Frame Curricula. In Hungarian: Kerettanterv. 51/2012. (XII. 21.) számú EMMI rendelet – a kerettantervek kiadásának és jóváhagyásának rendjéről, 2012 [Online] Available: https://www.oktatas.hu/kozneveles/kerettantervek/2012_nat [Accessed: 04-Febr-2020]

[3] Nagy T. K.: The paradox of the hungarian frame curricula in informatics. The Turkish Online Journal of Educational Technology, INTE 2018, pp. 910-922

[4] Booth, S.: Learning to program: A phenomenographic perspective. Acta Universitatis Gothoburgensis, Gothenburg, Sweden, 1992

[5] Soloway, E.: Should we teach students to program? Communications of the ACM, 1993, 36(10), pp. 21-25

[6] Ben-Ari, M. (2011). Non-myths about programming. Communications of the ACM, 2011, 54(7), pp. 35-37

[7] Fincher, S. et al.: Comparing Alice, Greenfoot & Scratch. 41st ACM technical symposium on Computer science education, ACM, 2010, pp. 192-193

[8] Fowler, A., Fristce, T. & MacLauren, M.: Kodu Game Lab: a programming environment. The Computer Games Journal, 2012, 1(1), pp. 17-28

[9] Klassner, F. & Anderson, S. D.: Lego MindStorms: Not just for K-12 anymore. IEEE Robotics & Automation Magazine, 2003, 10(2), pp. 12-18

[10] Papadakis, S. et al.: Novice Programming Environments. Scratch & App Inventor: a first comparison. Workshop on Interaction Design in Educational Environments, ACM, 2014

[11] Resnick, M. et al.: Scratch: programming for all. Communications of the ACM, 2009, 52(11), pp. 60-67

[12] Chmielewska, K. & Gilanyi, A.: Mathability and computer aided mathematical education. 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2015, pp. 473-477

[13] Csernoch, M., Biró, P., Máth, J. & Abari, K.: Testing Algorithmic Skills in Traditional and Non-Traditional Programming Environments. Informatics in Education, 2015, 14(2), pp. 175-197

[14] "Blueprints Visual Scripting" Epic Games. [Online] Available: https://docs.unrealengine.com/en-US/Engine/Blueprints/index.html [Accessed: 10-Oct-2019]

[15] Hutong Games: PlayMaker – Visual Scripting for Unit. [Online] Available: https://hutonggames.com/ [Accessed: 10-Jan-2020]

[16] YoYo Games: Drag And Drop Overview [Online] Available: https://docs2.yoyogames.com/ [Accessed: 10-Jan-2020]

[17] Fesakis, G. & Serafeim, K.: Influence of the familiarization with scratch on future teachers' opinions and attitudes about programming and ICT in education. ACM SIGCSE Bulletin, ACM, 2009, 41(3), pp. 258-262

[18] Kim, H., Choi, H., Han, J. & So, H. J.: Enhancing teachers' ICT capacity for the 21st Century learning environment: Three cases of teacher education in Korea. Australasian Journal of Educational Technology. 2012, 28(6) pp. 965-982

[19] Csapó, G.: Placing event-action based visual programming in the process of computer science education. Acta Polytechnica Hungarica, 2019, 16(2) pp. 35-57

[20] Csapó, G. & Sebestyén, K.: Educational Software for the Sprego Method. The Turkish Online Journal of Educational Technology, INTE 2017 October, pp. 986-999

[21] Lifelong Kindergarten Group:Scratch – Imagine, Program, Share [Online] Available: https://scratch.mit.edu [Accessed: 10-Nov-2019]

[22] Kalelioglu, F. & Gülbahar, Y.: The effects of teaching programming via Scratch on problem solving skills: a discussion from learners' perspective. Informatics in Education, 2014, 13(1), pp. 33-50

[23] Meerbaum-Salant, O., Armoni, M. & Ben-Ari, M.: Habits of programming in scratch. 16th annual joint conference on Innovation and technology in computer science education. ACM, 2011, pp. 168-172

[24] Franklin, D., Hill, C., Dwyer, H. A., Hansen, A. K., Iveland, A. & Harlow, D. B.: Initialization in scratch: Seeking knowledge transfer. 47th ACM Technical Symposium on Computing Science Education, ACM, 2016, pp. 217-222

[25] "Game Making Software – Construct 3" [Online] Available: https://www.construct.net/en [Accessed: 21-Sep-2019]

[26] Alexander, J.: Construct 2 – From Beginner to Advanced - Ultimate Course! | Udemy. 2016 [Online] Available: https://www.udemy.com/construct-2-from-beginner-to-advanced-build-10-games [Accessed: 06-Dec-2019]

[27] Scirra: Welcome to the Construct 3 Manual. [Online] Available: https://www.construct.net/en/make-games/manuals/construct-3 [Accessed: 21-Sep-2019]

[28] Scirra: Game Development Tutorials. [Online] Available: https://www.construct.net/en/tutorials?flang=1 [Accessed: 21-Oct-2019]

[29] Scirra: Construct Forum [Online] Available: https://www.construct.net/en/forum [Accessed: 21-Sep-2019]

[30] Scirra: ScirraVideos – YouTube [Online] Available: https://www.youtube.com/user/ScirraVideos [Accessed: 20-Nov-2019]

[31] Papp, P.: Texting or text management? Teaching text management in ICT textbooks. In Hungarian: Szövegelés vagy szövegszerkesztés? Szövegkezelés tanítása az informatika tankönyvekben. Debreceni Egyetem Informatikai Kar Tudományos Diákköri Konferencia 2019

[32] Erdélyi, A.: Down the yellow paved road of ICT textbooks. In Hungarian: Végig az informatika-tankönyvek sárga köves útján – Avagy Pöttömföldről eljutunk-e így az igazi varázslathoz? Debreceni Egyetem Informatikai Kar Tudományos Diákköri Konferencia 2019

[33] Nagy, R. K.: Learning spreadsheet management from textbooks: do we only manage the spreadsheets or do we solve problems? In Hungarian: Táblázatkezelés elsajátítása tankönyvekből: a táblázatot csak kezeljük vagy a problémát is megoldjuk? Debreceni Egyetem Informatikai Kar Tudományos Diákköri Konferencia 2019

[34] Baranyi, P. & Gilányi, A.: Mathability: Emulating and Enhancing Human Mathematical Capabilities, IEEE 4th International Conference on Cognitive Infocommunications (CogInfoCom), 2013, pp. 555-558

[35] Burcsi: Scratch programming. In Hungarian: Scratch programozás [Online] Available: http://www.burcsi.hu/_inf/Scratch/ [Accessed: 12-Nov-2019]

[36] Hattie, J.: Visible Learning for Teachers: Maximizing Impact on Learning, Routledge, 2012

[37] Csernoch, M.: Thinking Fast and Slow in Computer Problem Solving, Journal of Software Engineering and Applications, 2017, 10(1), pp. 11-40

[38] Chmielewska, K. & Gilányi, A.: Computer Assisted Activating Methods in Education. 10th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2019, pp. 241-246

[39] Biró, P. & Csernoch, M.: The mathability of computer problem solving approaches. 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2015, pp. 111-114

[40] Biró, P. & Csernoch, M.: The mathability of spreadsheet tools. 6th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2015, pp. 105-110

[41] Chmielewska, K., Gilányi, A. & Łukasiewicz, A.: Mathability and Mathematical Cognition. 7th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2016, pp. 245-250

[42] Chmielewska, K. & Matuszak, D.: Mathability and coaching. 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2017, pp. 427-432

[43] Pólya, G.: How To Solve It. A New Aspect of Mathematical Method. Second edition (1957) Princeton University Press, Princeton, New Jersey, 1954

[44] ELTE IK T@T Labor: Archivum | e-Hód. [Online] Available: http://e-hod.elte.hu/archivum/ [Accessed: 09-Aug-2019]

[45] Pohl, W. & Dagienė, V.: What is Bebras | www.bebras.org [Online] Available: https://www.bebras.org/ [Accessed: 15-Jan-2020]

[46] Bebras: International Challenge on Informatics and Computational Thinking [Online] Available: https://www.bebras.org [Accessed: 15-Jan-2020]

[47] Kahneman, D.: Thinking, Fast and Slow. Farrar, Straus and Giroux, New York, 2011

[48] IBM SPSS software: IBM SPSS Statistics [Online] Available: https://www.ibm.com/analytics/spss-statistics-software [Accessed: 01-Jan-2020]

[49] Dani, E.: The HY-DE Model: An Interdisciplinary Attempt to Deal with the Phenomenon of Hyperattention. Journal of Systemics, Cybernetics and Informatics 2016, 13:(6) pp. 8-14

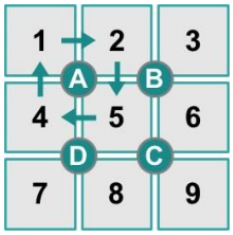
[50] Csernoch, M. & Dani, E.: Data-structure validator: an application of the HY-DE model, 8th IEEE International Conference on Cognitive Infocommunications (CogInfoCom), 2017, pp. 197-202

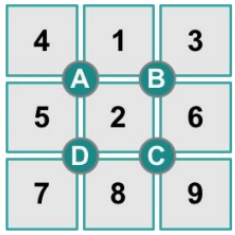
Appendix

1.1. The test sheet used for data-collection

ROTATING PUZZLE (2012-HU-01A)

Henry Beaver plays a new game. If he presses one of the buttons A, B, C or D in this game, numbers around the button will be rotated clockwise as shown on the pictures in left. The result of pressing the button A is shown to the picture in right. The initial state of the numbers is as seen on the left image.





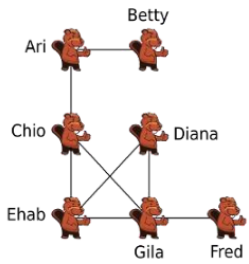
Henry Beaver pressed consecutively four buttons: D, C, B, B.

Where the number 4 happens to be after the pressing of the four buttons?

POPULARITY (2015-CA-01)

Seven beavers are in an online social network called Instadam. Instadam only allows them to see the photos on their own and their friends' page. In this diagram, if two beavers are friends they are joined by a line. After the supper holiday everybody posts a picture of themselves on all of their friends' page.

Which beavers' picture will be seen the most?



BEAVER CODE (2016-CH-12)

Barbara has been given two stamps. With one she can produce a little flower, with the other a little sun. Being a clever girl, she thinks of a way to write her own name by using the code.

Letter	B	A	R	E	Y
Code					

So Barbara becomes:

She then writes the name of her friend:

What is her friend's name?

PARTY GUESTS (2016-IS-02)

To arrange a dinner party Sara the beaver need to talk to five friends: Alicia, Beat, Caroline, David und Emil.

However, to talk to her other friends, there are a few points to consider:

1. Before she talks to David, she must first talk to Alicia.
2. Before she talks to Beat, she must first talk to Emil.
3. Before she talks to Caroline, she must first talk to Beat and David.
4. Before she talks to Alicia, she must first talk to Beat and Emil.

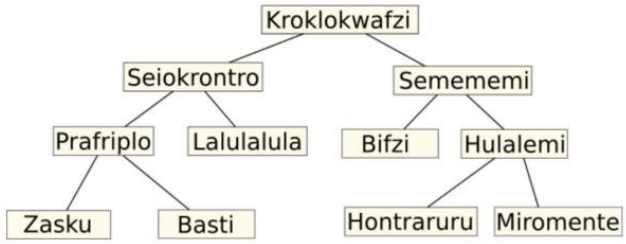
In what order should Sara talk to all of her friends if she wants to talk to all of them?

HIERARCHY (2016-CZ-03)

We have got a tree describing relationships between animals on Morgenstern planet. A link between two categories in the tree below means that all members of the lower category are also members of the upper category. Hence, some sentences can be formed looking at this tree, e. g. every "Hulalemi" is a type of "Semememi" and that some "Seiokrontro" are not a type of "Basti".

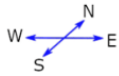
Only one of the following statements is true. Which one?

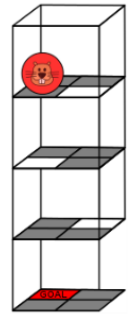
1. Every Basti is a Seiokrontro as well.
2. Some Hontraruru is not Semememi.
3. Every Zasku is Bifzi as well.
4. Every Prafriplo is Basti as well.



SPHERICAL ROBOT (2016-JP-03)

The BeaverBall is a toy that can be operated by remote control, and understands each of four direction commands.





If the BeaverBall moves to a white square, it drops down one level. The BeaverBall ignores commands that cause it to move outside the borders.

Look at the position of the BeaverBall in the picture above. With what commands can you make the BeaverBall to reach the GOAL?

DEACTIVATIN (2016-HU-06)

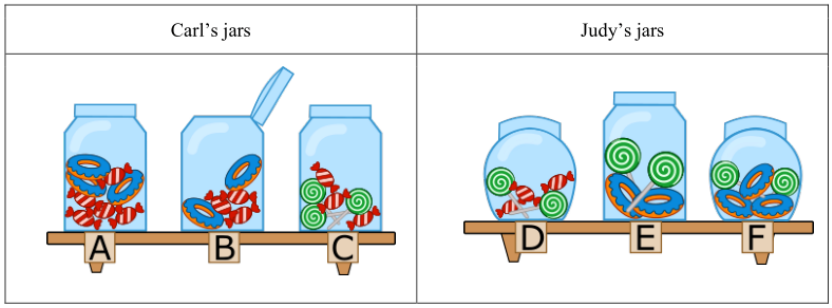
Beavers Carl and Judy have 3 jars of candies each.

Each jar can have several of the following 10 properties:

- It's either open (1) or closed (2).
- It either contains red candies with white stripes (3) or not (4).
- It either contains blue sugar donuts (5) or not (6).
- It either contains green spiral lollypops (7) or not (8).
- Its form is either round (9) or angular (10).

For example Carl's "A" jar has the following properties: 2, 3, 5, 8 and 10.

You can notice that Carl's jars have got some common properties just like Judy's jars.



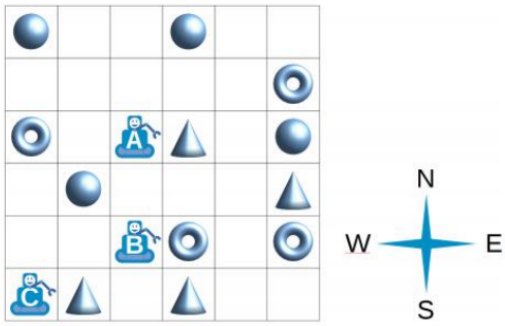
Which jar has both the properties that are common among Carl's jars and Judy's jars separately?

CONCURRENT DIRECTIONS (2016-IE-05)

In a warehouse, three robots always work as a team. When the team gets a direction instruction (N, S, E, W), all robots in the grid will move one square in that direction at the same time.

When the robots touch an item, they pick it up. You have to control them so that they will not pick up unnecessary items.

For example, if we give the list N, N, S, S, E to the team, then the robots will pick up two cones, and a ring.



What list of instructions can be sent to the robots so that the team picks up exactly a sphere, a cone, and a ring?

FOUR ERRANDS (2016-LT-03)

Beaver Alexandra wants to do the following tasks during her break (12:00 – 13:00):

- buy a book at a bookstore;
- buy a bottle of milk at a grocery;
- send the newly bought book by post;
- drink a cup of coffee in a cafeteria.

Alexandra estimated the time to complete each task. But these estimates are valid only outside of the busiest periods. So she is trying to avoid the busiest periods.

Place	Duration	Busiest periods
Bookstore	15 min	12:40 – 13:00
Grocery	10 min	12:00 – 12:40
Post office	15 min	12:00 – 12:30
Cafeteria	20 min	12:30 – 12:50

Help Alexandra order her tasks to make sure that she will avoid all of the busiest periods.

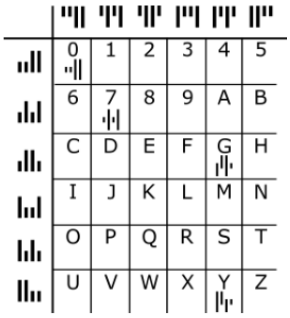
KIX CODE (2016-NL-04)

The Bebras Post Office uses postal codes that contain four characters. To make the postal codes readable by machines, they convert the postal codes into Kix codes.

In a Kix code, each character is represented by 4 vertical bars.

A code has 2 sections: upper and lower. The upper section contains only the middle and the top bars, while the lower section contains only the middle and the bottom bars. The middle bars overlap each other.

This table shows the codes for several characters:



Example: The Kix code for “G7Y0” is

Question: Another postal code has this Kix code.

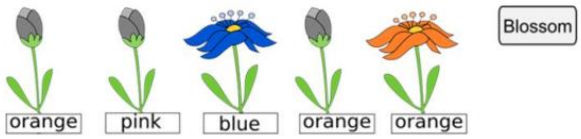
What is the postal code?

BLOSSOM (2016-SK-04)

Jane is playing a computer game. First the computer secretly chooses colours for five buds. The available colours for each flower are **blue**, **orange**, and **pink**. The chosen colours remain the same throughout the game. Jane has to guess which flower has which colour. She makes her first five guesses and presses the Blossom button.

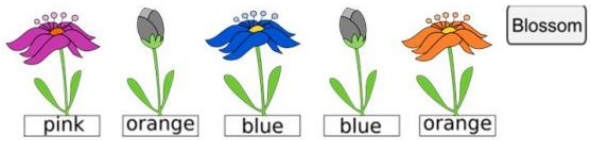
The buds, whose colours she guessed correctly, break into flowers. The others remain as buds.

Jane's first go:



Jane then has another go at guessing and presses the Blossom button again.

Jane's second go:



What colours did the computer choose for the flowers?