Olympiads in Informatics, 2024, Vol. 18, 1–24

DOI: 10.15388/ioi.2024.01

Giochi di Fibonacci Year II: Competitive Blocks Programming for Young Students

Giorgio AUDRITO[1], Sara CAPECCHI[1], Madalina G. CIOBANU[2], Luigi LAURA[3]

[1]*Department of Computer Science, University of Torino, Italy*
[2]*University of Salerno, Italy*
[3]*Uninettuno University, Rome, Italy*
e-mail: {giorgio.audrito,sara.capecchi}@unito.it, mciobanu@unisa.it, luigi.laura@uninettunouniversity.net

Abstract. We organized the second edition of *Giochi di Fibonacci* (Fibonacci's games), a programming contest for upper primary and lower secondary schools students; contestants compete in their own age division. The contest is organized in three phases, where the first one is based only on logical and algorithmical quizzes, whilst the other two deal with coding using a Blockly environment integrated in our contest's platform. In this paper we report our experience and analyze the feedback collected from both students and teachers.

Keywords: programming contest, Olympiads in Informatics, peer education, programming training.

1. Introduction

The scientific-cultural side of computer science, also referred to as computational thinking, helps to develop logical skills and the ability to solve problems creatively and efficiently, qualities that are important for all future citizens.

The significance of incorporating computational thinking and programming into the curriculum of primary and lower secondary education cannot be overstated, a trend underscored by the notable success of initiatives like Bebras[1] (Dagienė, 2008). Dagienė *et al.* (Dagienė *et al.*, 2022) provide an insightful overview of how computational thinking is being adopted globally in primary education; the challenges and considerations surrounding the integration of informatics into primary education, from curriculum design to teachers' perspectives, are thoroughly examined in (Dagienė *et al.*, 2019).

[1] `https://www.bebras.org/`

A wide array of methodologies has been explored to facilitate this integration, including unplugged education (Pluhár, 2021; der Vegt, 2016), employing platforms like Scratch (Fagerlund *et al.*, 2020), and introducing robot programming (Kanemune *et al.*, 2017) and LEGO robotics (Souza *et al.*, 2018). Additionally, gamification strategies have been effectively utilized to engage students (Combéfis *et al.*, 2016), as highlighted by Dolinsky (Dolinsky, 2022).

In Italy, according to the current national curricular recommendations, computer science related topics are included in broad areas of cross-disciplinary key citizenship digital competence area or general technology subject area and *"whenever possible, students can be introduced to simple and flexible programming languages in order to develop a taste for creation and for the accomplishment of projects [...] and in order to understand the relationships between source code and resulting behavior."*[2] As a consequence, the implementation of the curricular recommendations is delegated to self-motivated teachers who propose valuable initiatives also in informatics education. In this context, initiatives bringing students and teachers closer to programming play a very important role.

The "Giochi di Fibonacci" (Fibonacci's games) (Audrito *et al.*, 2023) are a programming contest in upper primary and lower secondary education, with each age division competing separately, aimed at enhancing students' computational thinking and programming skills. In the first edition (last year)"Giochi di Fibonacci" were divided in three distinct stages, where the initial stage is solely based on logical and algorithmic assessments, similar to Bebras, while the other two phases involve the use of coding, either via Scratch or a specially designed simplified pseudo-code programming environment catered to this competition. The lessons learned in the first year edition (Audrito *et al.*, 2023) have been the base for the design of this year's edition, that we discuss in this paper.

On the bright side, as we decided, we improved the competition by focusing only on a Block programming language (based on Blockly[3]); on the dark side, we experienced a huge bug in the newly designed but not enough tested new platform for the second phase. It would be nice to say that we corrected the bug immediately and it not affected the competition but, to be fair, it had definitely a non neglectable impact.

Thus, in this paper we report our experience in running (a buggy version of) an improved Giochi di Fibonacci competition. In Section 2, we recall the structure of the competition and report the emergence of the bug and the countermeasures taken. Then, in Section 3 we discuss the results of the feedback obtained, using questionnaires, from students and teachers. Finally, Section 4 addresses final remarks and conclusions.

[2] `https://www.miur.gov.it/documents/20182/51310/DM+254_2012.pdf`

[3] The choice of relying on Blockly instead of the more well-known similar platform Scratch is due to its easier integration with our website. Blockly can be tried online at: `https://blockly.games`

2. Giochi di Fibonacci

A detailed description of the (first edition of the) *Giochi di Fibonacci* is discussed in (Audrito *et al.*, 2023). In this section, we briefly present the three phases, that mimic the structure of the Italian Olympiads in Informatics (Audrito *et al.*, 2021). The first phase does not involve coding, and problems proposed are similar to the ones of Bebras, thus aiming to involve students from that competition to participate. The first edition phases were organised in the following way:

- **First phase:** logical, algorithmic and program comprehension quizzes, similar to Bebras but with more weight on “program comprehension” skills.
- **Second phase:** simple programming tasks (in Scratch or conventional languages).
- **Third phase:** more difficult programming tasks.

In the following sections we discuss each of the phases of the second edition, describing mainly the changes from the first edition, according to the *lessons learned* from (Audrito *et al.*, 2023).

2.1. *First Phase*

The first phase aimed to address logical and algorithmic thinking. Since the first phase last year was successful and well-received, we maintained the same structure for the test, with only a minor difference: presenting program comprehension tasks as block-based code, instead of as flowcharts.

We designed the phase acknowledging the diversity in school resources, that is, not all educational institutions are equipped with either fixed or mobile computer labs. Consequently, we offered schools the flexibility to select their preferred competition method: either via traditional pen-and-paper format or through an online platform. This approach ensured equal opportunity for all participants, regardless of their access to technolo-

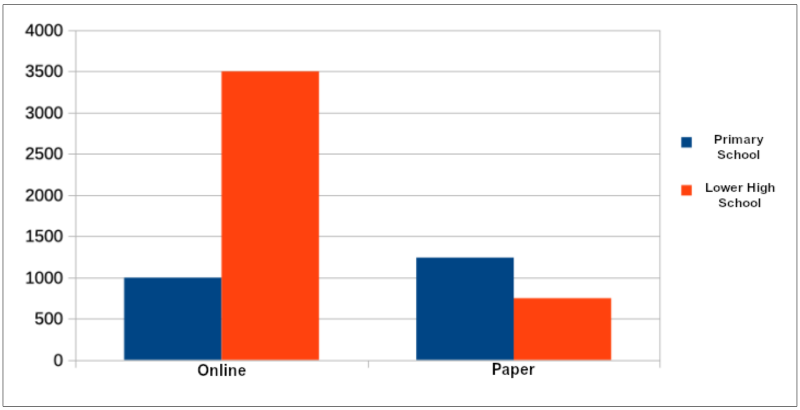


Fig. 1. Participating students by type of test.

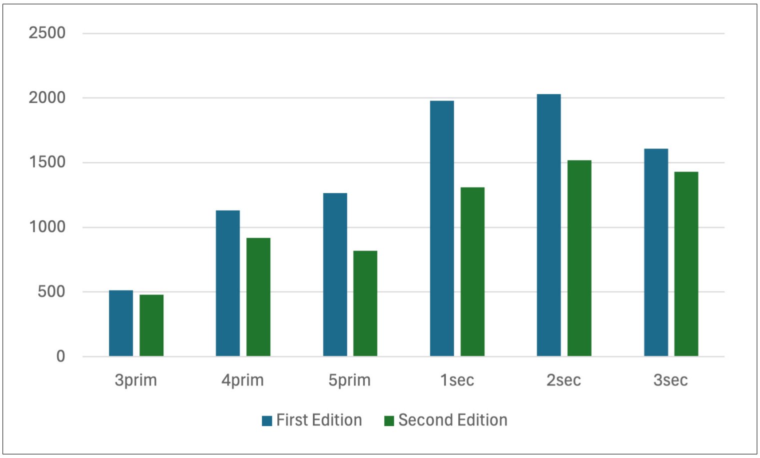


Fig. 2. Participating students by years, comparing the two editions.

gy. By analyzing the choice of testing methods across different educational levels (see Fig. 1), we see that in lower high schools there was a greater preference for the online test over the traditional pen-and-paper format. Conversely, in primary schools, the opposite trend was observed, with a higher prevalence of the pen-and-paper method over online testing. This may suggest a more widespread availability of computer labs in lower high schools w.r.t. primary schools.

6616 students participated in the first phase: 2352 primary school students and 4264 lower secondary school students. 2216 took the test on paper, while 4400 took it online. A total of 18 lower secondary school students obtained a full score of 50/50, while only one primary school student obtained a full score of 45/45. The scores were overall a bit lower that what we aimed: the average score was 14.7 for primary school and 17.8 for secondary school, while the median score was 15 for both primary and secondary school.

Fig. 2 shows the number of participating students divided by year of study, for both editions: 3prim, 4prim and 5prim are the last 3 years of primary school, while 1sec, 2sec and 3sec represent the 3 years of lower high school. We can see that the participation has been similarly distributed in the two editions, but lower overall for the second edition. This may have been a consequence of the exercises being too difficult last year, especially in the second phase.

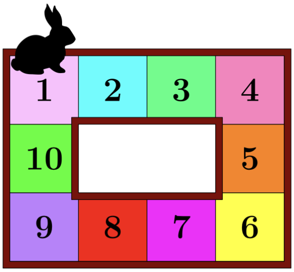
2.1.1. *Primary School*

The test for primary school contained 9 questions to be solved in 50 minutes, divided into three parts as follows:

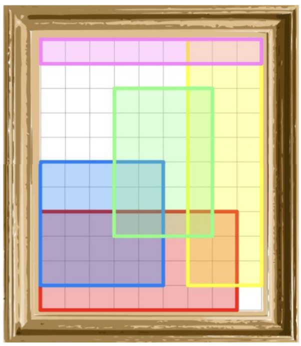
- Logical thinking questions (4 multiple-choice questions).
- Algorithmic thinking questions (2 open-ended numeric questions).
- Questions on interpretation of block programs. (3 multiple-choice questions).

In all three parts, the questions were ordered by increasing difficulty. Few sample questions follow.

Question 2. *The rabbit game board contains 10 squares, numbered from 1 to 10, with square 10 adjacent to square 1. Tip-Tap starts from square 1 and, in 4 subsequent turns, advances by 9, by 4, by 8 and finally by 7 boxes. In the end, which square is it on?* Multiple Choice Answers: *A) 4, B) 9, C) 1, D) 5, E) 3*

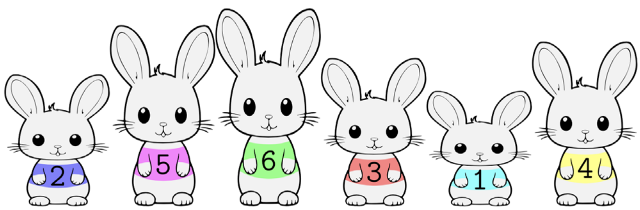


Question 3. *Tip-Tap received 5 rectangular postcards, measuring (in cm) 8×4, 5×5, 3×10, 9×1 and 4×6. Now he wants to buy a rectangular bulletin board in which to put the postcards, possibly overlapping but not rotated. For example, this is a possible bulletin board of area 9×11=99 that contains postcards.*



To save money, Tip-Tap would like to purchase the smallest bulletin board possible: how much is the smallest area (in cm^2) of a bulletin board that can contain all postcards? Multiple Choice Answers: *A) 80, B) 90, C) 0, D) 10, E) 85*

Question 5.1. *Tip-Tap friends all lined up for the count! Each of them has a different height, written on their t-shirt.*



In one move, Tip-Tap can choose two consecutive friends and remove the shorter of the two from the line. What is the minimum height of a friend who can remain in line after 5 moves?

Question 5.2. *What is the minimum height of a friend who can remain in line after 4 moves?*

Question 6. *In what order should these instructions be placed to obtain the number 6 in the variable x?* [We show the english translation on the right side of the original image of the contest.]

1: il contenuto di x diventa 2	1: the content of x becomes 2
2: il contenuto di x viene moltiplicato per 2	2: the content of x gets doubled
3: il contenuto di x aumenta di 1	3: the content of x gets increased by one

Multiple Choice Answers: *A) 3,2,1; B) 2,3,1; C) 2,1,3; D) 1,2,3; E) 1,3,2*

Fig. 3 reports the response distribution for the different questions. The second question had the highest number of correct answers. Conversely, question 5.2 recorded the highest count of incorrect responses. Moreover, question 3 and question 6 stood out as the ones with the highest frequency of unanswered responses, possibly suggesting that their statement was hard to understand.

2.1.2. *Lower High School*

The test for lower high school students consisted of 10 questions to be solved in 50 minutes. The questions were multiple choice or numerical open response, and were divided into three parts:

- Logical thinking questions (3 multiple-choice questions).

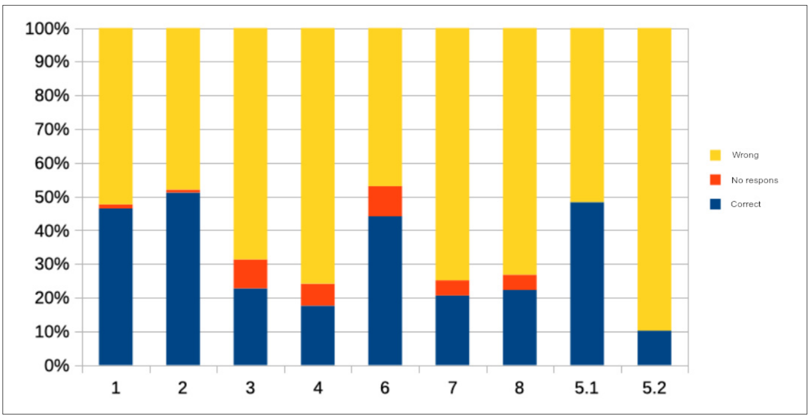
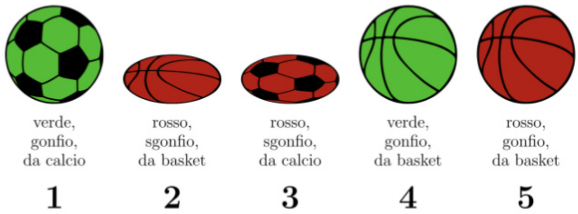


Fig. 3. Response distribution in primary school.

- Algorithmic thinking questions (4 open-ended numeric questions).
- Questions on interpretation of block programs. (3 multiple-choice questions).

In all three parts, the questions were ordered by increasing difficulty. Few sample questions follow.

Question 7. *Tip-Tap has to decide whether to throw away some of these balloons:*



To do this, he follows this procedure: [Pseudocode in english is shown to the right side of the original image of the contest]


```
procedure SelectBalls
    for each ball b do
        if b is basketball then
            if b is red then
                keep b
            else
                throw b
            end if
        else
            if b is deflated then
                throw b
            else
                keep b
            end if
        end if
    end for
end procedure
```

Which balls does Tip-Tap throw? Multiple Choice Answers: *A) 2,3,4; B) 1,4; C) 3,4; D) 1,5; E) 1,2,5*

Question 4.1 was the question with the greatest number of correct answers, and question 4.2 was the question with the greatest number of incorrect answers. Those questions were identical to questions 5.1 and 5.2 reported previously for primary schools. Question 7 was the question left blank by the greatest number of students. Question 7 might have been left blank because it was a question that required some competence on block programming. However, question 8 is also a question on block programming and was the question with the second highest number of correct answers, so other factors might be into play.

2.2. *Second Phase*

In this second edition, we significantly changed the second phase, leading to what we evaluated as a considerable improvement. This change stems directly from the valuable feedback received from teachers last year. In particular, in the first edition, the second phase was perceived too difficult and the web system hard to use. Many students lacked even rudimentary computer programming skills, including familiarity with block-based programming concepts. This deficiency is largely attributed to the minimal exposure to computer science in the standard education curriculum in Italy, particularly up to the age of 14. Addressing this disparity is one of our primary objectives: to introduce students to the world of information technology at an earlier stage, fostering a deeper and more meaningful engagement with the subject matter.

In the previous edition, we allowed the use of traditional languages or Scratch. For each supported language (including Scratch), they were provided with a starter file containing the code handling input and output, and the students had to implement only the main procedure. It was up to the students to edit and run the program with external software and tools. The obstacle in solving the exercises was not only in the difficulty of the exercises but, as emerged from the feedback, also in the method provided to perform the exercises. In particular with Scratch, the devised system required to interact back-and-forth with two separate websites (the scratch website and the contest website), and this increased the barrier of accessibility to the contest.

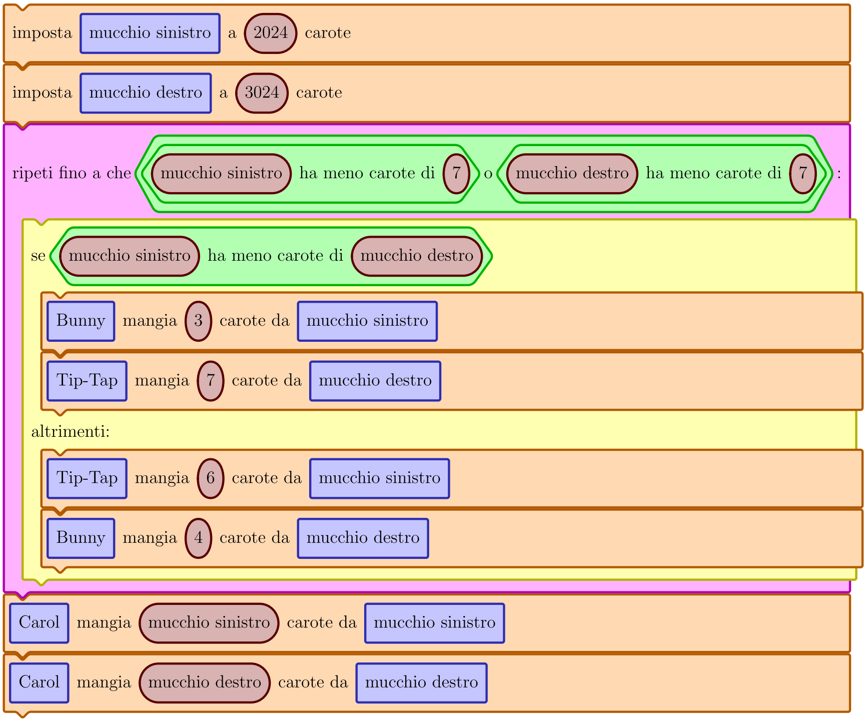
This year we added an extension to our platform in order to integrate Blockly and avoid the above mentioned issues. The test had the same structure for both primary and secondary schools, and consisted of 6 questions to be solved in 100 minutes. The questions were divided into two parts:

- 3 questions on interpretation of block programs.
- 3 block programming interactive questions.

Each block programming question required to write a single blockly program, which was then evaluated on three levels of increasing difficulty. All questions were accessible from a same page in the contest website, which was based on the same QuizMS platform used on the first phase, but with an extension providing Blockly support. In both parts, the questions were ordered by increasing difficulty. The set of questions was different for primary and secondary schools, as detailed in the following.

2.2.1. *Primary School*

Question 3. *The rabbits at the Fibonacci farm have prepared two huge piles of carrots. At the beginning the left pile contains 2024 carrots, while the right pile contains 3024. Bunny, Tip-Tap and Carol eat them following this procedure:* [Pseudocode in english is shown below the original image of the contest]




```
Set left pile to 2024 carrots
Set right pile to 3024 carrots
repeat
    if left pile has fewer than 7 carrots or right pile has fewer than 7 carrots then
        if left pile has fewer carrots than right pile then
            Bunny eats 3 carrots from left pile
            Tip-Tap eats 7 carrots from right pile
        else
            Tip-Tap eats 6 carrots from left pile
            Bunny eats 4 carrots from right pile
        end if
    end if
until left pile has fewer than 7 carrots or right pile has fewer than 7 carrots
Carol eats left pile carrots from left pile
Carol eats right pile carrots from right pile
```

How many carrots does Carol eat? Multiple Choice Answers: *A) 0, B) 4, C) 1, D) 2, E) 8*

Question 4.1. *Tip-Tap needs to sort out his old collection of N footballs. Since he doesn't have room for all of them, he decided to keep all the inflated soccer balls and basketballs, while throwing away the deflated basketballs. To do this, Tip-Tap can perform the following actions:*

- *Keep: put the next ball on the shelf.*

- *Throw: throw away the next ball in the bin.*
- *Soccer ball: true if the next ball is a soccer ball.*
- *Inflated balloon: true if the next balloon is inflated.*
- *Finish: finish putting the balloons away.*

Write a program that allows Tip-Tap to sort all his balloons! (see Fig. 4)

Fig. 5 reports the response distribution for the different questions. Question 4.1 had the highest number of correct answers, while question 3 had the highest number of blank answers. On the other hand, questions 5.3, 6.2 and 6.3 were not solved by any student.

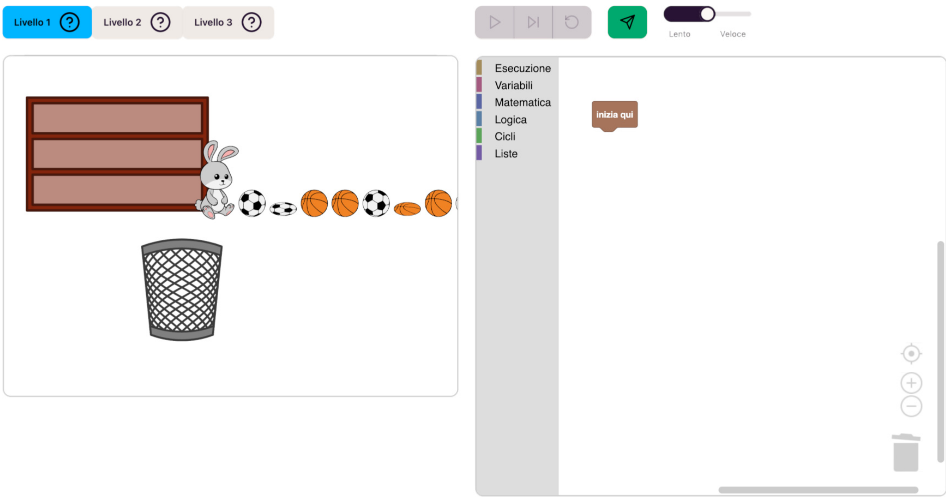


Fig. 4. Second Phase – Primary School – Question 4.1.

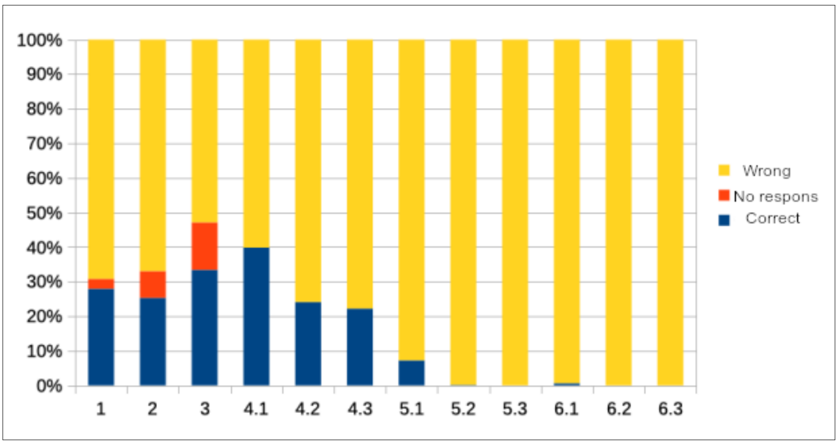
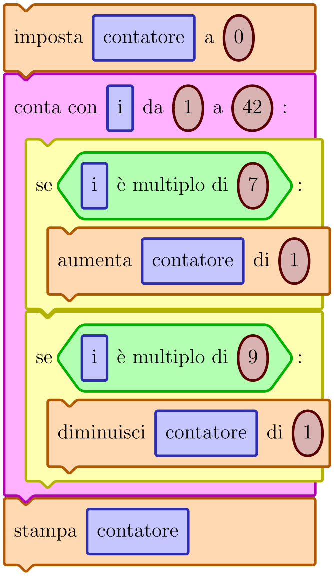


Fig. 5. Second Phase – Response distribution in primary school.

2.2.2. *Lower High School*

Question 3 *While fixing up his attic, Tip-Tap came across a very old programming book. On the first page he found the following procedure:* [Pseudocode in english is shown to the right side of the original image of the contest]




```
Set counter to 0
for i from 1 to 42 do
    if i is a multiple of 7 then
        Increase counter by 1
    end if
    if i is a multiple of 9 then
        Decrease counter by 1
    end if
end for
Print counter
```

Unfortunately the next page is ruined so Tip-Tap can't understand which number will be printed at the end... help him! What number is printed from the last block? Multiple Choice Answers *A) 2, B) 6, C) 10, D) 4, E) 0*

Question 6.3 *Tip-Tap wants to build a new shed for his farm! First, he needs to build the two supporting columns: one on the left S centimeters high, and one on the right D centimeters high. To do this he plans to stack some blocks taken from a construction set, composed of a single block for every possible height between a minimum of 1 centimeter and a maximum of M centimeters, and which in total reach exactly the total height of the two columns. Now you can do these operations:*

- right column height: the current height of the right column.
- *left column height: the current height of the left column.*
- *stack block i on the right: adds the block i centimeters high to the right column, if it has not already been used.*
- *stack block i on the left: adds the block i centimeters high to the left column, if it has not already been used.*
- finish: complete the columns and build the canopy.

Help Tip-Tap complete the shed as planned! (see Fig. 6)

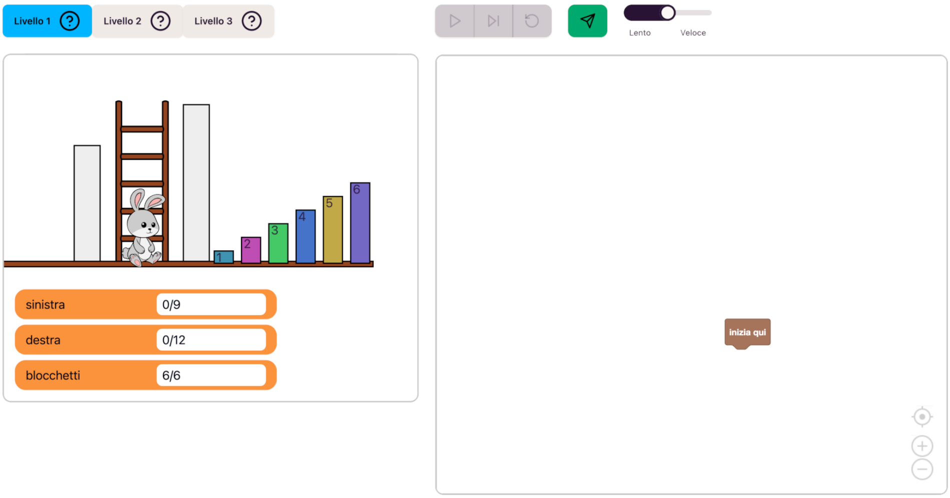


Fig. 6. Second Phase – Lower High School -Question 6.3.

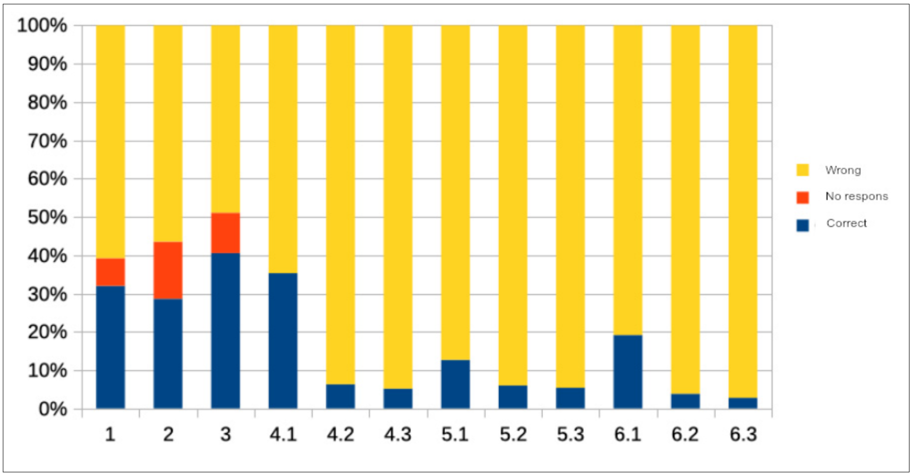


Fig. 7. Second Phase – Response distribution in lower high school.

Fig. 7 reports the response distribution for the different questions. Question 3 had the highest number of correct answers. Conversely, question 2 (which was the same as question 3 for primary school) recorded the highest count of wrong answers. Question 6.3 was the level solved by the lowest number of students.

2.2.3. *The Hunt for the Bug*

The second phase was scheduled to be held during March 13th, 2024. Each school could choose the two-hour window for the test at any time during the day, to accommodate their needs. Few schools asked to have the test on the 12th or 14th due to logistic reasons and were granted this possibility. The online platform used for the test allowed participants to directly register for the test, using a passphrase given by their teacher. Their answers in the test were locally saved in the browser's cache, with Blockly questions being visualized and scored directly in the browser, so that unreliable connection with the back-end could not impair the students' experience. The website would still try to regularly synchronize the students' answers with a back-end, leveraging the Google Firebase development platform. The teacher had also access to a managing dashboard, were he could see the results of all of its students in real-time, as they were saved in the Firebase back-end.

The contest operations seemed to proceed smoothly both on the 12th and on the 13th morning. Around noon, a teacher wrote us to report that the results she was seeing in the teachers' dashboard did not match the results the participants were seeing in their webpage, and were actually consistently lower. We first thought that could be due to errors from the teachers' side, or network problems from the school: but with some interaction we the teacher, thanks to her cooperation, we realised that indeed the result synchronisation was not working properly. Quickly we realised that this was due to a misconfiguration in Firebase, that rejected updates above a certain maximum size. That maximum size was not a problem for classic questions, that can be encoded with few bytes, but it was relevant for Blockly tasks, for which the whole blockly program was saved and easily reached the maximum allowed size.

By 1 PM the allowed size was raised, preventing the problem from happening for the 15% of schools that still had to start the test. However, many schools were affected severely, as the students' answers on Blockly tasks were mostly not saved. We immediately gave instruction to the teachers on how they might still recover results through the browsers' cache: 30% of schools managed to reconstruct the students' scores by this means, and 13% of schools declared that their students were not affected by the bug (probably because of low results on Blockly tasks). The remaining 42% of schools were affected by the bug and unable to reconstruct the scores.

This bug was a very unfortunate experience, making it hard to properly select the best students for the final phase. In order to mitigate its consequences, we opted for a selection criteria heavily grounded on a per-school basis, since students in a same school were identically affected by the bug. We admitted the first student of each school, regardless of his score, and the second student provided he reached a given minimum score. Given the very good results we obtained in the final phase, we believe that this mitigation strategy worked reasonably well.

Of course, the bug had an impact on the impression of the contest to the affected parties. However, the bug did not affect the test experience per se, during which the student would obtain the correct feedback from the system and could enjoy the challenge regardless. From the feedback gathered, we believe that only few of the schools were significantly unhappy of the experience, with most of them being instead supportive and understanding.

2.3. *Third Phase*

Since last year the third phase seemed too difficult for primary school students, only lower high school students participated this year in the third phase. The test consisted of 4 block programming questions to be solved in 3 hours. Questions were sorted by increasing difficulty. The problems were the following:

Question 1. *Tip-Tap loves chocolate, so he bought himself a chocolate bar made of $N \times M$ squares. His K farm-mates would also like to eat chocolate, and Tip-Tap is too good to not give them some! Then, for K times he breaks the tablet into two rectangular parts, not necessarily equal, and gives one of the two to one of his K companions. At the end of the process, he will keep the remaining last piece for himself.*

The tablet can only be broken along the edges of the squares, horizontally or vertically, so as not to divide any square in two. Furthermore, once a part is broken, it is immediately taken and eaten by a friend without giving him the opportunity to break it further. Tip-Tap would like to know how to break the bar K times so that he can keep the most possible pieces at the end.

You can use these blocks:

- width: the current width of the tablet.
- *height: the current height of the tablet.*
- *companions: the number of companions who still ask for chocolate.*
- *break x squares horizontally: break the tablet horizontally, leaving x rows for a partner.*
- *break x squares vertically: break the tablet vertically, leaving x columns for a partner.*
- finish: eat the remaining chocolate.

Help him break the bar K times while keeping as many squares as possible! (see Fig. 8)

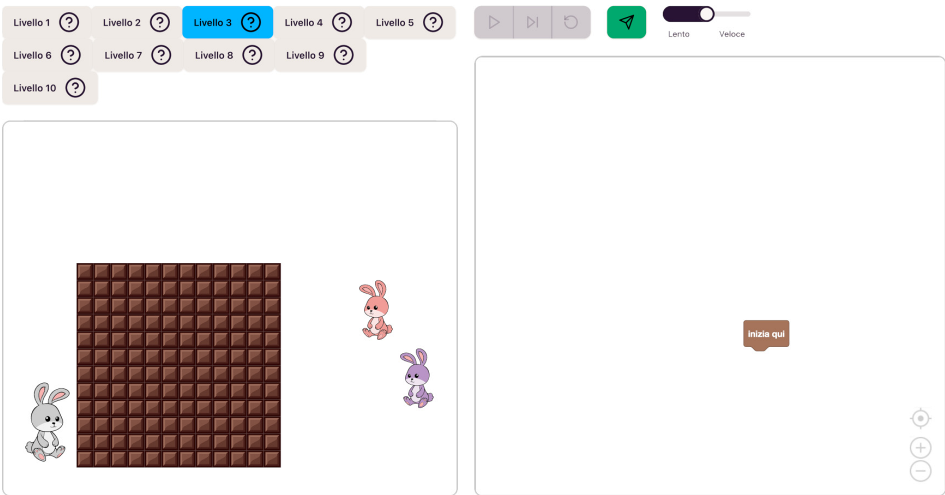


Fig. 8. Third Phase – Question 1.

Question 2. *The rabbits of the Fibonacci farm have just bought a new very efficient electric car! They can't wait to try it, so they organize a test trip to the nearby mountains. The route they want to take is made up of uphill and downhill sections. Along the route there are N charging stations where you can stop, at different heights. The car uses one unit of energy to climb 1 meter in altitude, while it gains one unit of energy by descending 1 meter in altitude, and it does not need energy to advance on flat terrain. Unfortunately, the machine starts without energy, and to recharge it can wait a minute at one of the charging stations for each unit of energy it wants to obtain at that point. You can use these blocks:*

- *N: The length N of the path.*
- *energy: the current amount of energy.*
- *altitude of charging station i: the altitude of the i-th charging station on the route.*
- *advance: continue your journey to the next column, if you have enough energy.*
- *recharge for x minutes: wait x minutes at a charging station to recharge x units of energy.*
- *finish: turn off the machine.*

The rabbits start from charger 1, and must arrive at charger N. Plan the trip to the mountains, ensuring that the car does not stop before arriving! (see Fig. 9)

Question 3. *Carol dropped her calculator, and now it doesn't work as it should! The only working keys are* −, ×, 1 *and* 2. *To use the calculator she is forced to start from either number 1 or number 2 (by pressing the corresponding key), and apply one of the 4 possible operations that still work, zero or more times:*

- *subtract 1;*
- *subtract 2;*
- *multiply by 1;*
- *multiply by 2.*

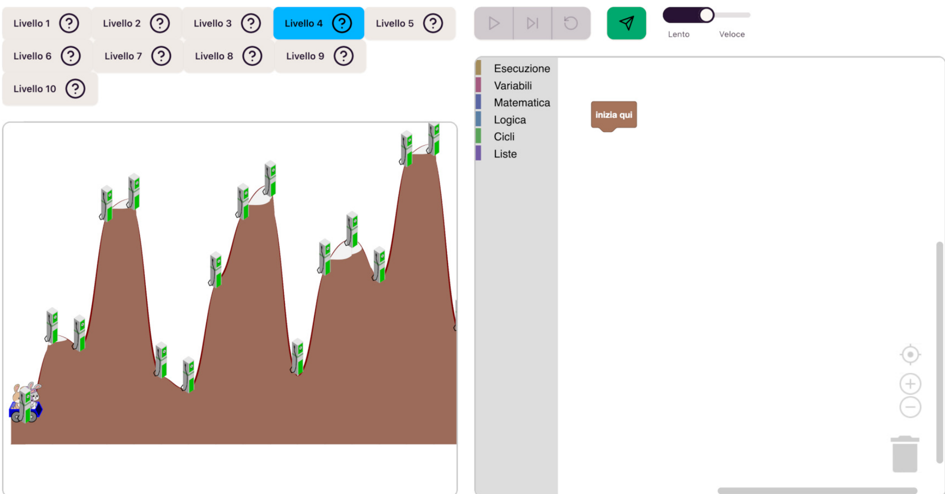


Fig. 9. Third Phase – Question 2.

To prank his friends, she would like to reach number N on the calculator. How many operations must be performed at least to achieve this? You can use these blocks:

- *N: the number N it wants to reach.*
- *ends in x operations: reports that it is possible to reach the number N in x operations.*

Note that you are not asked to reconstruct the operations to be performed: just calculate the necessary number of operations! (see Fig. 10)

Question 4. *The brand new SuperBunny video game is finally on the market! Bunny, the protagonist of the video game, in each level must overcome N obstacles numbered from 1 to N. On each obstacle there are two platforms (at different heights) on which Bunny can jump: the obstacle number i is made up of a higher platform which is at a height of A_i meters, and of a lower platform at a height of B_i meters.*

Bunny starts from the ground at height 0 and must first jump onto obstacle number 1 by choosing one of the two platforms. Once he reaches obstacle 1, he will choose one of the two platforms of the next obstacle, 2, and jump onto it. The objective of the game is to overcome all the obstacles in order up to obstacle number N. Even though Bunny can choose which obstacle platform to jump onto each time, not all jumps are the same! In fact, the bigger the jump, the longer it takes to do it. To jump from the platform at height h to a platform on the next obstacle at height k, Bunny will take an amount of seconds equal to the absolute difference between h and k. The total time taken to complete a level is the sum of the times taken in each jump. How many seconds does it take for Bunny to complete the level? You can use these blocks:

- *N: the number N of obstacles.*
- *high platform i: the height A_i of the highest i-th platform.*
- *low platform i: the height B_i of the lowest i-th platform.*

Fig. 10. Third Phase – Question 3.

- *absolute difference between x and y: the absolute difference* $|x - y|$ *between* x *and* y*.*
- *minimum between x and y: the minimum value between two numbers* x *and* y*.*
- *finish in x time: reports that it is possible to reach the N-th obstacle in* x *time.*

Furthermore, if you need it, you will have the possibility to write down a value of your choice on each platform (see Fig. 11) with these blocks:

- *high platform i value: the value written on the* i*-th high platform.*
- *low platform i value: the value written on the* i*-th low platform.*
- *set value of high platform i to x: write the value* x *on the* i*-th high platform.*
- *set value of lower platform i to x: write the value* x *on the* i*-th lower platform.*

79 students from 55 lower high schools participated in the third (and final) phase, of which 75 obtained a non-zero score. 39 students were awarded: 20 bronze, 11 silver and 8 gold medals. Fig. 12 reports the response distribution for the different questions. The

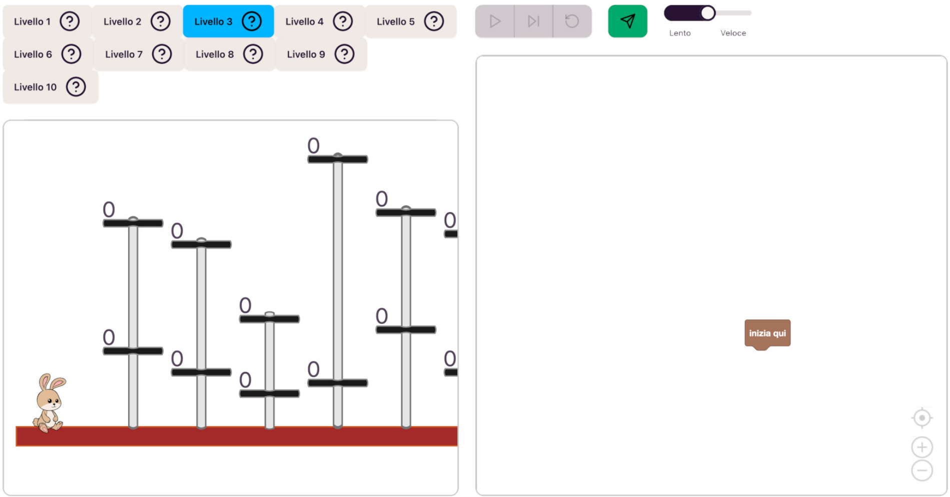


Fig. 11. Third Phase – Question 4.

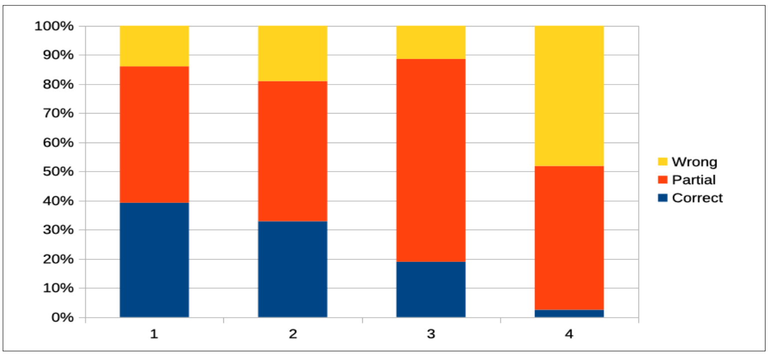


Fig. 12. Third Phase – Response distribution.

question ordering reflected the number of fully correct answers (10/10 levels done). Questions 1-2-3 all had a similarly low number of wrong answers, with most students completing at least some levels. Question 4 was much more difficult, as it required dynamic programming for a full solution, which was achieved only by two students.

3. Students and Teachers' Feedback

After the first phase of this years' edition, we collected feedback from teachers without involving students to maximize participation: asking teachers to make a high number of students fill out questionnaires could have been a barrier to participation. For the second and third phases of this year's edition, we instead performed a more systematic evaluation of the students' and teachers' feedback. For the second phase, the test was split in the two parts of progra*m comprehension an*d coding, and we followed this split during feedback collection from students, thus asking the questions separately for the two parts. We adapted some of the question proposed by MEEGA+KIDS (Gresse von Wangenheim *et al.*, 2020; Petri *et al.*, 2019), a model to evaluate the quality of educational games used as instructional strategy for computing education. Participants (both teachers and students) were asked to assess their level of agreement with each of the following sentences using a 5-point Likert scale, ranging from "Strongly disagree" to "Strongly agree":

- S1 Questions were sufficiently clear.
- S2 The graphical presentation of the test was enticing.
- S3 Using the web platform to answer questions was not cumbersome.
- S4 The topics covered were interesting.
- S5 Questions were adequately challenging.
- S6 Questions were not repetitive and boring.
- S7 Students had fun working on the questions.
- S8 Solving the test made students feel satisfied.
- S9 The things learned from questions were satisfactory.
- S10 Students' results depended on their personal commitment and skill.
- S11 I'd recommend others to participate in the future.

Context information was also asked to teachers, to assess the backgrounds and initiatives the students had before the tests, including:

- C1 List the preparation activities the students had before the second phase test.
- C2 Detail whether the second phase bug impacted the scores of your students.

The information above described was collected in order to measure:

- RQ1 The impact of various preparatory activities on students' results.
- RQ2 The correlation between the teachers' impressions on their students and direct students' feedback.
- RQ3 The impact of the bug on students' satisfaction metrics.
- RQ4 The difference between the program comprehension and coding sections in satisfaction metrics.
- RQ5 Which factors are more decisive on the students' satisfaction.
- RQ6 Which factors are more decisive on the intention to recommend participation.

We focus our research on the second phase in lower high school, for which we have an almost complete feedback. Many students and teachers in primary schools did not give feedback, and that could sway the distribution of results. For questions considering students' results, we restrict only to the 58% of schools that were not affected by the bug or could reconstruct the scores, excluding the 42% of schools with unreliable scores.

3.1. *Overall Results*

Table 1 summarises the overall feedback gathered. We received feedback from 920 students out of the 1300 participating (71%), and 56 teachers out of the 66 participating (85%). Out of them, 445 students and 32 teachers come from schools able to handle the bug. For all questions, the standard deviation of responses was low, as most respondents gave intermediate agreement to most questions (neither agree nor disagree). The overall feeling was slightly more positive than negative for students (with an average score of about 3.2), and more decidedly positive for teachers (with an average score of about 3.6). Still, the results suggest that there is margin for improvement in future editions. It is worth mentioning that the best scores were seen in questions S6, S3, S10 for students, S2 and S4 for teachers. This suggests that the platform was effective, and that the novelty of tasks and their appropriateness to test students' skills were widely recognised. Indeed, also by looking at individual suggestions from participants, the main source for discontent was task difficulty, perceived too high by a good fraction of participants.

On the 14th of May we streamed, using YouTube, the Awards Cerimony[4] that included also some video contributions by participants.

Table 1

Summary of the feedback collected, by scoring agreement from 1 (strongly disagree) to 5 (strongly agree). Students' feedback is split for test part A (program comprehension) and B (coding)

	students (A)		students (B)		teachers	
	mean	*st.dev*	*mean*	*st.dev*	*mean*	*st.dev*
S1	2.76	1.01	2.61	1.12	3.14	0.86
S2	3.31	1.15	3.31	1.18	3.91	0.80
S3	3.41	1.27	3.41	1.27	3.92	0.88
S4	2.83	1.13	2.87	1.15	3.96	0.90
S5	3.22	1.08	3.15	1.12	3.00	1.22
S6	3.42	1.15	3.25	1.17	4.16	0.88
S7	3.04	1.19	3.01	1.20	3.27	1.14
S8	3.39	1.25	3.37	1.23	3.23	1.09
S9	3.28	1.15	3.23	1.13	3.78	0.98
S10	3.73	1.09	3.59	1.17	3.23	1.09
S11	3.17	1.22	3.17	1.22	3.71	1.04
TOT	3.23	0.72	3.18	0.79	3.57	0.73

[4] Available at: `https://www.youtube.com/watch?v=sNhQy4MW5zk`

3.2. *Research Questions*

We analysed the correlation matrix between all gathered questions, in order to answer RQ1–RQ6. A detailed report follows for each item.

3.2.1. *Impact of Preparatory Activities*

To assess RQ1, we measured the correlation between the main preparatory activities gathered in C1 and students' scores, for the 445 students with scores unaffected by the bug. The four main preparatory activities performed were:

1. Blockly-based training classes.
2. Scratch-based training classes.
3. Students' trying to solve the demo contest we prepared.
4. Teachers' explaining the solution of the demo contest we prepared.

Unfortunately, the correlation of scores with the preparatory activities turned out to be not statistically significant (ranging from −0.02 to 0.11). We believe, however, that this might be a consequence of the feedback gathering method used, and possibly of the bug swaying results. For the future, we plan to investigate further details about preparation activities (to estimate how long they prepared for and whether they had done other activities, courses, etc.).

3.2.2. *Correlation between Students and Teachers Impressions*

To assess RQ2, we considered all 920 feedbacks and looked at the correlation between questions S1–S11 asked to students and to teachers. These correlations were very low, ranging from −0.03 to 0.11: this suggests that teachers' feedback is not a good predictor of students' feedback.

3.2.3. *Impact of the Bug on Student Satisfaction*

The impact of the bug on students' satisfaction (RQ3) was also not particularly relevant: correlation with questions S1-S11 ranged from −0.02 to 0.11. This is probably due to the fact that feedback from students was gathered at the end of the test, and the students' experience during the contest was not affected by the bug. Most students learned about the bug much later, and that might have had an effect on feedback only if feedback was collected later in time.

3.2.4. *Difference between Program Comprehension and Coding*

The correlation between answers from students on test part A and B (RQ4) was, instead, very strong. It ranged from a minimum of 0.56 (for question S5) to a maximum of 0.69 (for questions S4, S8, S9, S10). Apparently, the students were not able to differentiate their experience on the two parts of the test, suggesting that they were balanced enough in their main qualities.

3.2.5. *Main Factors for Students' Satisfaction*

To assess RQ5, we considered as main satisfaction metrics the answers to questions S6, S7, S8, S9. As main candidate factors for affecting students' satisfaction, we considered their results, personal interest (S4) or presentation (S1, S2, S3). The correlation with results turned out to be unexpectedly low, ranging from 0.06 (S6 for part B) to 0.13 (S7-S8 for part A). The strongest correlation was with students' interest (S4), ranging from 0.27 to 0.54 for the various pairs for questions, with an average correlation of 0.43. A lower but still meaningful correlation was registered with presentation, ranging from 0.21 to 0.48 for the various pairs of questions, with an average correlation of 0.36.

3.2.6. *Main Factors for Recommending Participation*

To assess RQ6, we first excluded students' scores and having experienced the bug as main factors, since those had very low correlations (from −0▷06to 0▷09) with the intention of recommending participation (S11). We then looked at the correlations with every other question asked, for both students' and teachers' (see Table 2). The correlations were all quite strong, but personal interest (S4) stands out as the best predictor for S11, with satisfaction (S7, S8, S9) following closely. Given that personal interest also has a very high correlation with satisfaction, it seems likely that it could be the main cause for S11 as well. It is worth mentioning that even though S6 (boringness) is also a measure of satisfaction, it had much lower correlation with S11. This may be caused by the fact that S6 had the highest scores overall, getting good agreement also for students which didn't engage with the test. We also remark that the correlation was lowest with S3 (probably also because of its higher scores overall), and with S1 and S2 for teachers.

3.3. *Lessons Learned*

The feedback gathered mostly confirmed the choice of the new test structure and web platform design, which we plan to keep for the upcoming year. The overall satisfaction scores could still be improved, with one of the main causes being the perceived difficulty of tasks. However, given that this is a second phase for a selected audience, we do not plan to lower the difficulty of the tasks: instead, we plan to instruct teachers to

Table 2

Correlation of other questions with recommending participation (S11)

	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
students (part A)	0.42	0.44	0.29	0.52	0.28	0.35	0.47	0.38	0.45	0.32
students (part B)	0.40	0.50	0.29	0.46	0.35	0.40	0.52	0.45	0.53	0.37
teachers	0.20	0.09	0.29	0.57	0.53	0.40	0.57	0.53	0.42	0.40

administer the test to fewer students, trying to restrict to only those truly interested and with the necessary skills to enjoy the test. In fact, this year the percentage of participants passed from phase 1 to phase 2 was particularly high (32%), with respect to the percentage from phase 2 to phase 3 (4%) and similar percentages last year (11% and 5%). Since participating schools handle the first round of selection autonomously, we can only suggest a course of action to teachers, but how to implement it is ultimately up to them. We reckoned that there is a widespread will from teachers to include as much people as possible in all phases of the competition. Next year, we will try to manage this impulse by clarifying that the second phase is only intended for the very best students of a school, and suggesting to use this phase as a guided class activity for the other students, with the teacher helping them in solving the tasks. We will also require a higher level of commitment for participation in the second phase, asking each student to register to our website in order to participate (this year, registration was required only for the third phase).

4. Conclusions

The *Giochi di Fibonacci* is a programming contest for upper primary and lower secondary students, structured to foster computational thinking and programming skills across separate age groups. In its inaugural edition last year, the contest was organized into three distinct stages. The first stage focused exclusively on logical and algorithmic quizzes akin to those found in Bebras, while the subsequent stages required participants to code, either in Scratch or in a simplified pseudo-code environment specifically developed for the competition.

This year, in the second edition, we improved the structure of the contest mainly by swapping Scratch for an integrated Blockly module, smoothing the user experience. Unfortunately, as detailed in Section 2.2.3, this also added a significant bug in the second stage.

Overall, the mostly positive feedback received largely supported our decision to adopt the new test format and web platform design, which we intend to maintain for the next year. While there is room to improve overall satisfaction ratings, the primary concern was the tasks' perceived difficulty. However, since this was a second phase intended for a selected group, we do not anticipate reducing the task difficulty. Instead, we aim to guide teachers to limit the number of students taking the test, focusing on those who are genuinely interested and possess the requisite skills to appreciate the challenge.

References

Audrito, G., Ciobanu, M., Laura, L. (2023). *Giochi di Fibonacci*: Competitive programming for young students. *Olympiads in Informatics*, 17, 19–31.

Audrito, G., Di Luigi, W., Laura, L., Morassutto, E., Ostuni, D., *et al.* (2021). The Italian Job: Moving (Massively) Online a National Olympiad". *Olympiads in Informatics*, 15, 3–12.

Combéfis, S., Gytautas Beresnevičius, G., Dagienė, V. (2016). Learning programming through games and contests: Overview, characterisation and discussion. *Olympiads in Informatics*, 10.1, 39–60.

Dagienė, V. (2008). The BEBRAS contest on informatics and computer literacy–students drive to science education. In: *Joint Open and Working IFIP Conference. ICT and Learning for the Net Generation, Kuala Lumpur*, pp. 214–223.

Dagienė, V., Jevsikova, T., Stupurienė, G. (2019). Introducing informatics in primary education: curriculum and teachers' perspectives. In: *Informatics in Schools. New Ideas in School Informatics: 12th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2019, Larnaca, Cyprus, November 18–20, 2019, Proceedings 12*. Springer, pp. 83–94.

Dagienė, V., Jevsikova, T., Stupurienė, G., Juškevičienė, A. (2022). Teaching computational thinking in primary schools: Worldwide trends and teachers' attitudes". In: *Computer Science and Information Systems* 19.1, pp. 1–24.

Dolinsky, M.S. (2022). Primary School Programming Olympiads in Gomel Region (Belarus). *Olympiads of Informatics*, 16, 107–123.

Fagerlund, J., H¨akkinen, P., Vesisenaho, M., Viiri, J. (2020). Assessing 4th grade students' computational thinking through Scratch programming projects. *Informatics in Education*, 19(4).

Kanemune, S., Shirai, S., Tani, T. (2017). Informatics and programming education at primary and secondary schools in Japan. *Olympiads in Informatics*, 11, 143–150.

Petri, G., Gresse von Wangenheim, C., Ferreti Borgatto, A. (2019). MEEGA+, Systematic Model to Evaluate Educational Games. In: *Encyclopedia of Computer Graphics and Games*. Ed. by Newton Lee. Springer.

Pluhár, Z. (2021). Extending computational thinking activities. *Olympiads in Informatics*, 15, 83–89.

Souza, I.ML, Andrade, W.L, Sampaio, L.MR, Souto O Araujo, A.L. (2018). A Systematic Review on the use of LEGO® Robotics in Education. In: *2018 IEEE Frontiers in Education Conference (FIE)*. IEEE, pp. 1–9.

der Vegt, W.V. (2016). Bridging the Gap between Bebras and Olympiad: Experiences from the Netherlands. *Olympiads in Informatics*, 10, 223–230.

Gresse von Wangenheim, C., Petri, G., Ferreti Borgatto, A. (2020). MEEGA+KIDS: A Model for the Evaluation of Games for Computing Education in Secondary School. In: *RENOTE* (2020). `https://api.semanticscholar.org/CorpusID:225504503`

G. Audrito is involved in the training of the Italian team for the IOI since 2006, and since 2013 is the team leader of the Italian team. Since 2014, he has been coordinating the scientific preparation of the OIS and of the first edition of the IIOT. He got a Ph.D. in Mathematics in the University of Turin, and currently works as a Junior Lecturer in the University of Turin.

S. Capecchi received the M.Sc. degree in Computer Science at the University of Florence in 2002 and a Ph.D. in Computer Science (2003-2006) at the Department of Computer Science, University of Florence. She currently works as a assistant professsor in the University of Turin. Her main research interests include computer science education.

M.G. Ciobanu is involved in Italian Olympiads in Informatics since 2009. She got a Ph.D in Computer Science in the University of Salerno, and in addition to being a high school teacher and also a research fellow in the University of Salerno.

L. Laura is currently the president of the organizing committee of the Italian Olympiads in Informatics that he joined in 2012; previously, since 2007, he was involved in the training of the Italian team for the IOI. He is Associate Professor of Theoretical Computer Science in Uninettuno university.