Olympiads in Informatics, 2023, Vol. 17, 19–31

DOI: 10.15388/ioi.2023.02

Giochi di Fibonacci: Competitive Programming for Young Students

Giorgio AUDRITO[1], Madalina CIOBANU[2], Luigi LAURA[3]
[1]*Department of Computer Science, University of Torino, Italy*
[2]*University of Molise, Italy*
[3]*Uninettuno University, Rome, Italy*
e-mail: giorgio.audrito@unito.it, madalina.ciobanu@unimol.it, luigi.laura@uninettunouniversity.net

Abstract. In this paper, we share the experience gained by organizing and running a programming contest for upper primary and lower secondary schools students; contestants compete in their own age division. This contest, called *Giochi di Fibonacci* (Fibonacci's games), is organized in three phases, where the first one is based only on logical and algorithmical quizzes, whilst the other two deal with coding, either in Scratch or in a simplified pseudo-code programming environment developed for this scope.

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

1. Introduction

The integration of computational thinking skills and computer programming in primary and lower secondary education has become increasingly important in recent years, as also witnessed by the large success of Bebras[1] (Dagienė, 2008); see also the recent work of Dagienė *et al.* (2022) for a picture of the worldwide diffusion of Computational Thinking teaching in primary schools, whilst some considerations about the curriculum and teachers' perspectives related to the introduction of informatics in primary education are discussed in (Dagienė *et al.*, 2019).

Indeed, as also observed in (Dolinsky, 2022), there is a plethora of approaches: unplugged education (Vegt, 2016; Pluhár, 2021), the use of Scratch (Fagerlund *et al.*, 2020), robot programming (Kanemune *et al.*, 2017), LEGO robotics (Souza *et al.*, 2018), and gamification (Combéfis *et al.*, 2016).

In this context, programming contests can serve as an effective tool to motivate students and expose them to problem-solving challenges. In this paper, we describe the introduction

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

of 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. The competition, named ”Giochi di Fibonacci” (Fibonacci's games), is comprised of 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.

In the literature, there are reports about similar experiences (in Belarus (Dolinsky, 2022)) and also about training students for this events (Vegt, 2016; Kiryukhin *et al.*, 2022).

This paper is organized as follows: in the next section we provide a general overview of the *Giochi di Fibonacci*, whilst the three sections that follow detail the three phases of the competition, providing info about both the structure and the results obtained. Then, in Section 6, we discuss the lessons we learnt in this first experimental edition, together with some changes we plan to implement in the next year. Finally, Section 7 addresses final remarks and conclusions.

2. Giochi di Fibonacci

Our competition mimics the structure of the Italian Olympiads in Informatics (Audrito *et al.*, 2021), divided into three phases as well. 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.

Why Fibonacci. Leonardo Pisano, more commonly known as Fibonacci, is a mathematician from the 13th century. He learned the Hindu-Arabic numeral system during his travels to North Africa with his father, a customs agent, and described them in the Liber Abaci, or “Book of Calculation”, which revolutionized the way commerce was conducted. It enabled average individuals to buy and sell goods, convert currencies, and maintain accurate records of their possessions more easily than ever before. Its publication led to extensive international commerce and contributed to the scientific and artistic advancements of the Renaissance.

Fibonacci is most famously known for the Fibonacci sequence, which is a recurring pattern of numbers that can be generated using a simple recursive algorithm. This algorithm is based on the idea that each number in the sequence is the sum of the previous two numbers:

$$fib(n) = fib(n-1) + fib(n-2)$$

where the first two numbers in the sequence are 0 and 1, and thus the sequence begins with 0, 1, 1, 2, 3, 5, 8, 13, 21, etc. In his book (Devlin, 2011), Devlin points out the remarkable similarities between the computing revolution that took place in Tuscany during the 13th century, under the guidance of Fibonacci, and the one that began in California's

Silicon Valley more recently, with the personal computing revolution of the 1980s started by Steve Jobs, the founder of Apple computers, with the introduction of the mouse and a graphical interface. Devlin offers a unique perspective, showing how history repeated itself.

The structure of the competition The competition has been organized in three distinct phases. Due to the young age of the participants all the phases took place in their own schools, under the supervision of their own teachers.

- *First phase*: logical and algorithmic quizzes, similar to Bebras but with more weight on "program reading" quizzes.
- *Second phase*: pseudo-code or Scratch programming.
- *Third phase*: pseudo-code or Scratch programming, with more difficult problems.

In the following sections we discuss each of the phases, describing in detail the types of exercises proposed and the overall feedback received after the conclusion of this first experimental edition.

3. First Phase

During the first phase, students could access the administered exercises through appropriate accessible online tools both via computer and via tablet or smartphone[2]. The exercises provided were intended to measure logic and basic mathematics thinking, ability to identify problem-solving algorithms and ability to understand descriptions of simple procedures. In this phase, no knowledge of programming languages was required to carry out and understand the exercises.

Questions required either multiple-choice or numeric answers. All multiple choice questions had 5 options, of which only one was correct. The score assigned for these questions was:

- 5 points for a correct answer.
- 1 point for a blank answer.
- 0 points for an incorrect answer.

Each numerical open-ended question required an integer (possibly negative) number as an answer. The score assigned for these questions was:

- 5 points for a correct answer.
- 0 points for an incorrect or blank answer.

The phase was managed and carried out independently by every single educational institution, at the times most suited to them, giving 50 minutes to students to complete the test. Further details follow, divided between primary and lower secondary schools. Overall, we had 9 questions for primary school and 10 questions for lower secondary school, and the response distributions are shown, respectively, in Fig. 1 and Fig. 2.

[2] An interactive training version of the exercises is available online at: `https://scolastiche.olinfo.it`

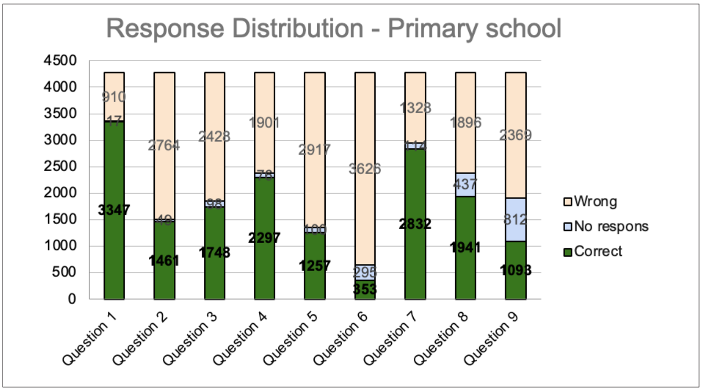

Fig. 1. Response distribution in primary school.

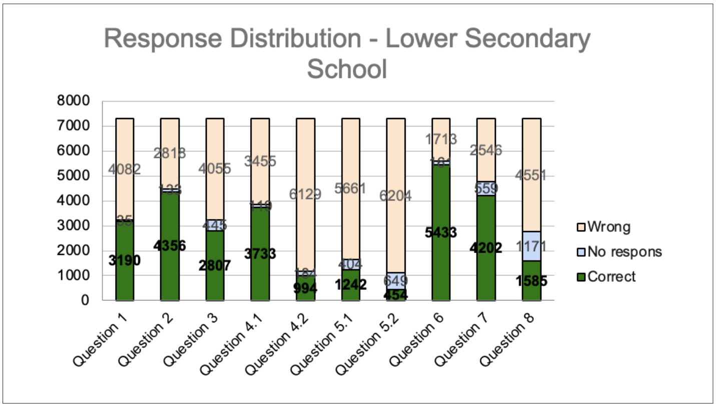

Fig. 2. Response distribution in lower secondary school.

3.1. *Primary School*

The test for primary school contained 9 questions, divided into three parts as follows:

- Logical thinking (4 multiple-choice questions).
- Algorithmic thinking (2 open-ended numeric questions).
- Program reading (3 multiple-choice questions).

In each of the three parts, the questions were roughly ordered by increasing difficulty. A sample of the questions follows.

Question 1. *Every Monday, Tap-Tap picks 10 carrots from his garden. Every day of the week, Tap-Tap eats a carrot. How many carrots does he have left over each week?* (see Fig. 3)

Question 6. *Tip-Tap has gone on a trip to Turing's farm, and he wants to bring back lots of carrots to his farm mates. On Turing's farm, the Carrot Market takes place every* *week, where it is possible to buy many boxes of carrots, each at a cost of 10 carrots. Carrots are good for your eyesight, and Tip-Tap eats so many he can see through the boxes! These are the numbers of carrots contained in each box [. . .] how many carrots can Tip-Tap earn at most, by buying a set of boxes of his choice?* (see Fig. 4)

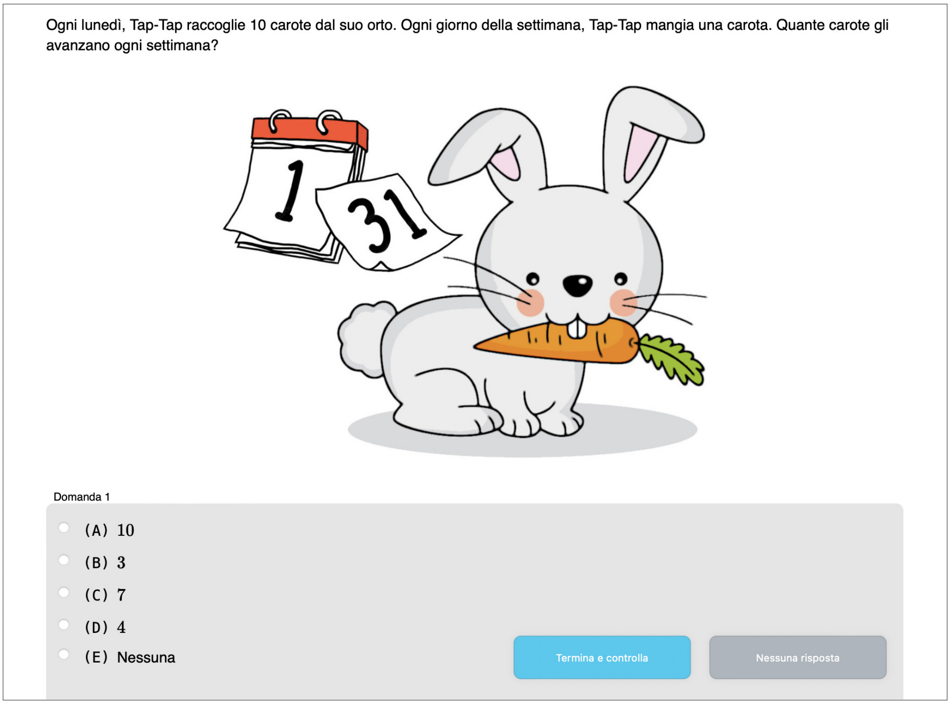

Fig. 3. User interface for question 1 (multiple-choice, logic thinking).

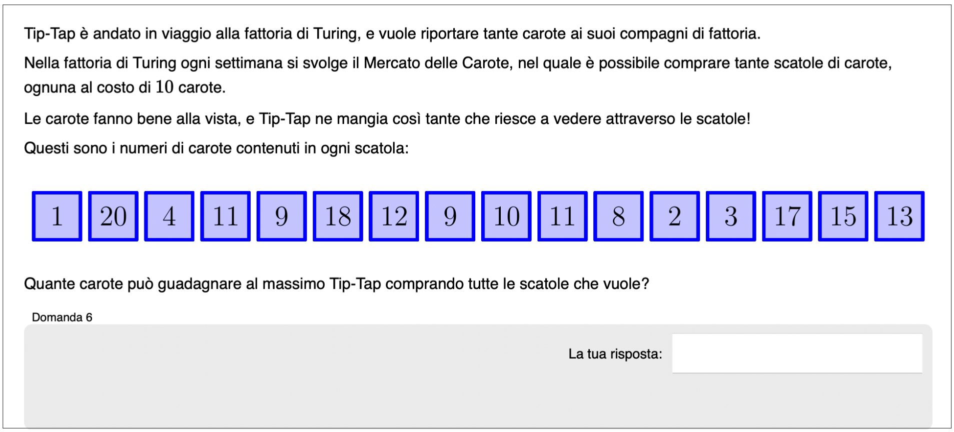

Fig. 4. User interface for question 6 (open-ended, algorithmic thinking).

This question is the one that received the most wrong answers as we can see from the graph in Fig. 1. That was somewhat expected, as the topic of the question (algorithmic thinking) is mostly novel for this age group. In primary schools in Italy, competitions already exist that develop logical thinking, but not developing algorithmic thinking.

Question 8. *Consider this process, represented as a flowchart. The procedure refers to three numerical variables, represented by letters a, b and c. This program runs twice. The first time the variables are assigned values $a = 7$, $b = 4$ and $c = 6$. The second time the values assigned are instead $a = 5$, $b = 7$, $c = 9$. What numbers does the procedure write in the two runs?* (see Fig. 5)

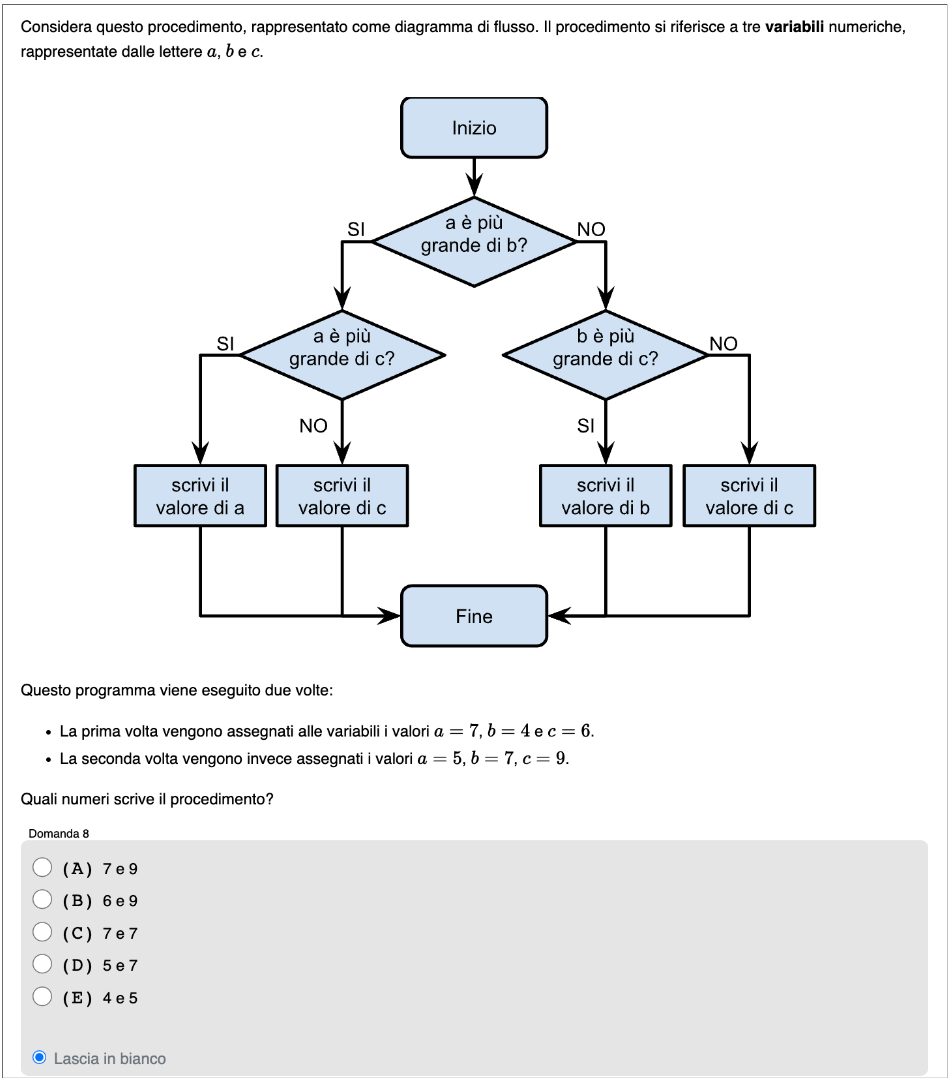

Fig. 5. User interface for question 8 (multiple choice, program reading).

3.2. Lower Secondary School

The test administered to lower secondary school students consisted of 10 questions, divided into three parts as follows:

- Logical thinking (3 multiple-choice questions).
- Algorithmic thinking (4 open-ended numeric questions).
- Program reading (3 multiple-choice questions).

In each of the three parts, the questions were roughly ordered by increasing difficulty. Some of the easier questions were shared with the test for primary school: 2 logical thinking questions, 2 algorithmic thinking questions, and all program reading questions. The two additional algorithmic questions were increased difficulty follow-ups of the two questions shared with primary schools. A sample of the questions follows.

Question 1. *Bunny found three piles of books in the library of the Fibonacci farm! Bunny would like to read any two gardening books, and he knows that the books on this subject are the ones with a yellow cover. To get a book Bunny has to move all the books above it. Bunny is very lazy, so he wants to be able to grab any two of the books he's interested in by moving as few books as possible! What's the minimum number of books Bunny has to move, counting the gardening books he takes?* (see Fig. 6)

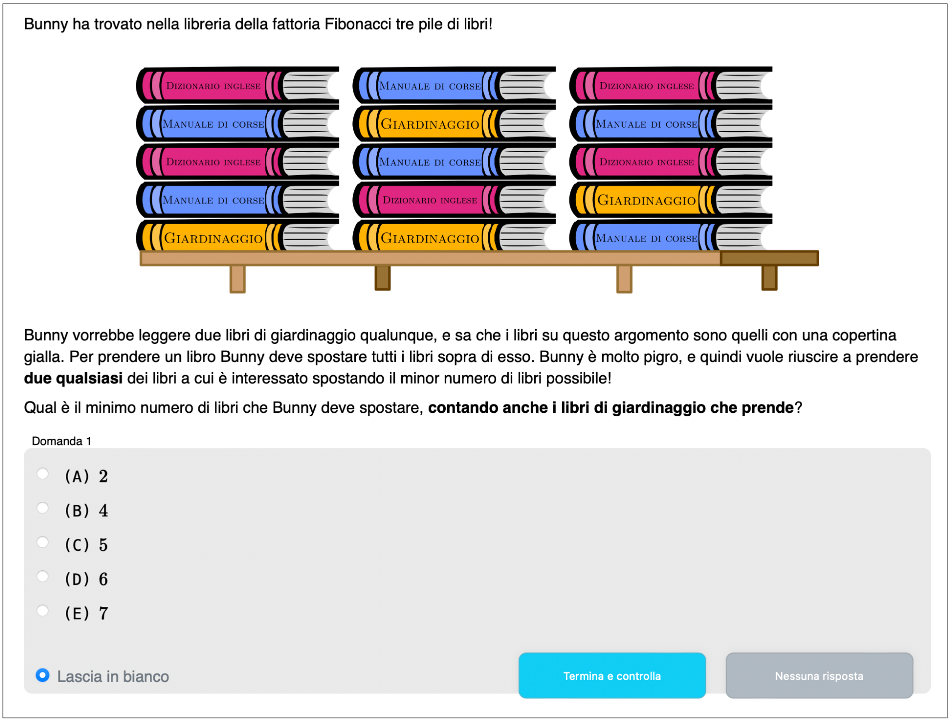

Fig. 6. User interface for question 1 (multiple-choice, logic thinking).

Question 4.1. *Bunny found these five slips of paper with numbers written on them: [. . .] Bunny wants to know which numbers can obtain by aligning the slips vertically and reading a column. For instance, aligning slips as in the picture, he can obtain 14518: [. . .] What is the largest number he can obtain in that way?*

Question 4.2. *Actually, Bunny would also like to figure out what is the biggest number he can get if he can change the order of the slips. For example, exchanging the first sheet with the last one, the order of the sheets would become: [. . .] What is the largest number he can obtain in that way?* (see Fig. 7)

Question 4.2 is the one with the most percentage of incorrect answers (see Fig. 2). Even though lower secondary school students performed better on the two algorithmic questions shared with primary school students, they still had issues on the more complex follow-ups of them, showing that algorithmic thinking is indeed a skill that needs to be further encouraged and developed also in their age group.

11,581 students took part in the first phase: 4,274 from primary school and 7,307 from lower secondary school. The school with the highest number of participants was the "I. C. L. Da Vinci/G. Carducci" school in Palermo, with 633 students. 86 students obtained a full score: 38 from primary school and 48 from lower secondary school. The school with the most full scores was the "I. C. Torgiano-Bettona" primary school, with 13 full scores. The mean score was 19.5 and the median score was 20, the same for both primary and lower secondary schools. The reported satisfaction was high: the students enjoyed learning and showing off their potential, through a test deemed adequate in all its sections, and effective administration tools despite some minor technical difficulties.

4. Second Phase

Students who achieved sufficient results in the first phase were invited to participate in a second phase dedicated to coding which involved carrying out the questions using the computer. Also in this case, the test was prepared at a national level by the technical-didactic operational unit of the Italian Informatics Olympics committee. The competition consisted in solving algorithmic problems by writing computer programs. The programming language used was a choice of Python, Scratch, or Pseudo-code based on suggestions from the school teachers. The test consisted in three tasks to be completed in two hours, different for the two school levels but with an overlap: primary schools had tasks *mele*, *dadi*, *monologo*; while lower secondary schools had tasks *dadi*, *monologo*, *soldatini*.[3] The students who obtained the best results in this second phase were invited to carry out the tests of the third phase.

1,320 students took part in the second phase: 295 from primary school and 1,025 from lower secondary school. Unfortunately, the test turned out to be very difficult and only half of these managed to score points: 142 from primary school and 442 from

[3] An interactive training version of the exercises is available at: `https://demo.fibonacci.olinfo.it`

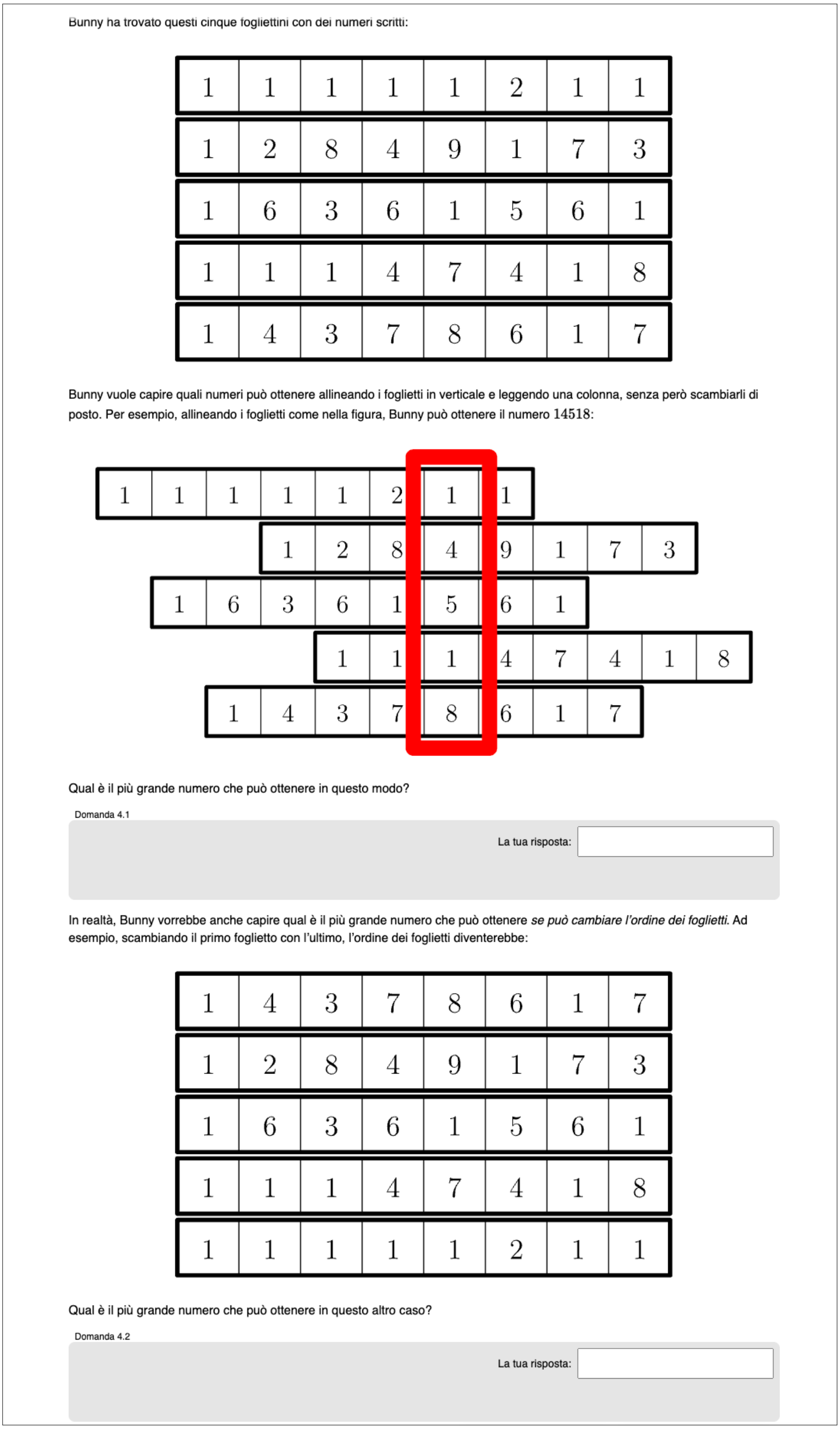

Fig. 7. User interface for questions 4.1 and 4.2 (open-ended, algorithmic thinking).

lower secondary school. There was only one full score for primary school, and six full scores for secondary school. 25 primary school students with a score of at least 55 points, and 57 lower secondary school students with a score of at least 100 points were selected for the national final. The feedback gathered from teacher was varied, but leaning on the negative side overall, as the test was discouraging most students. Particularly negative was the interaction with the test system for Scratch: as we weren't able to integrate Scratch within our platform, students were required to download and re-upload multiple files between two sites, resulting in a cumbersome and confusing interaction.

5. Third Phase

67 students participated in the third and final phase: 21 from 7 primary schools and 46 from 18 lower secondary schools. Of these, 18 primary school students and 36 secondary school students managed to get points. Among these, 32 medals were awarded, following the assignments used in other scientific competitions: 14 bronzes, 12 silvers, and 6 golds. The test consisted in four programming tasks to be solved in three hours. Three tasks were borrowed from the regional selections of the Italian Informatics Olympiads (*rettangolo*, *newlines* and *muro*), while the fourth easier task was specific to the competition (*formiche*).[4] Every task was solved by at least a contestant, but no contestant solved every task, so that the maximum score was of 118 points out of 200. This was a satisfactory result for the contest itself, but quite far from the level that higher secondary school students achieved on the common tasks. The students selected for the national Italian Informatics Olympiads all scored at least 110 points on the three common tasks, with all of them fully solving the easier of the three tasks (*rettangolo*). As the additional *formiche* task was even easier, this projects their likely cutoff score to be at about 160 points out of 200. Since even the best scoring student from lower secondary school was quite far from this cutoff, we decided to not invite any student from this competition to join the Italian Informatics Olympiads, as we feared that would not be a constructive experience for them.

The feedback for this phase was positive overall: even though the test had the same format of the second phase, which was not well received, the more selected pool of students was able to handle the system effectively. This is not surprising as only the students performing well on the second phase were selected for the third, which are students that were already able to work successfully with the competition format previously. On the other hand, it is an indication that the format and level of the third phase was indeed appropriate for a national-level final competition for lower secondary schools, as sufficiently many students nation-wide were able to score well. Unfortunately, the same can not be said for elementary schools: only one student scored well (obtaining a lower gold medal with 104 points out of 200), while the other 17 participating students scored at most 11 points out of 200 (which were not enough for any medal).

[4] An interactive training version of the exercises is available at: `https://demo.fibonacci.olinfo.it`

6. Lessons Learned

Based on the feedback and results gathered, we concluded that the competition should live for the following years, but with several necessary changes. As the first phase was the most successful, we plan to leave it mostly untouched, only reducing slightly the weight of logical quizzes in favor of algorithmic and program interpretation. We also plan to propose the programs to be interpreted in a block-like format, in order to be more preparatory for the following phases; and making our first phase more of a “further step” after Bebras. In this way, we plan to perform slightly more selection before the second round, to ensure that most of the students proceeding in the competition have the skills needed to take profit from it.

As the second phase was the most unsuccessful, we decided to completely rethink it. Instead of having it with an identical format as the *third* phase (but with easier tasks), we plan to have it with a similar format as the *first* phase, having it hosted on the same quiz-based web platform, with the goal of making the second phase somewhat more of an intermediate step between the first and the third. The second phase will differ from the first in two main aspects:

- The removal of the section on logical quizzes, thus only focusing on algorithmic thinking and program reading.
- The algorithmic questions will be solvable by writing a program in Blockly[5] computing their answers.

More precisely, each algorithmic question will feature a Blockly editor integrated with the website, and multiple answer boxes for inputs of increasing complexity (inspired by the two-step questions asked in the first phase for lower secondary schools). By composing a simple program solving the question and pressing a “run” button, it will be possible to automatically fill in all answers and see whether they are correct. Only the first input will be small enough to be solved by hand, so that being able to correctly write a block-based solution should result in an higher score.

Finally, as the third phase was successful for lower secondary schools, we plan to leave it mostly untouched, but propose it only to lower secondary school students. Few selected primary school students may be invited as well if they score well enough, with their teacher’s consent, but by being an exception rather than the rule we hope that having very few primary school students at the third phase in this way should not detract from the sentiment of the (many) primary school students that will not be selected. In fact, we feel that we cannot propose a full-fledged programming contest to primary school students in Italy at the time of this writing (except for very few exceptional students).

[5] 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`

7. Conclusions

In this paper we described our experience in the organization of a programming competition devoted to students of upper primary and lower secondary education.

Being a first edition, we clearly communicated that it was an experimental edition, and that we would collect feedback from both teachers and students.

Overall, the competition has been a success, with some issues that we plan to fix in the following editions: besides minor fixes, the main change will be the in the second phase, that will follow the format of the first phase, thus being some sort of a more difficult first phase instead of an easier third phase. Also, we plan to have only the first two phases for the upper lower education, with only few exceptions for some very talented students that will be invited with their teacher's consent.

Summing up, our experience with organizing and running the Giochi di Fibonacci programming contest has demonstrated, once again, the large need for computational thinking and programming skills in upper primary and lower secondary education. Our contest has shown, as observed in prior researches, that students of these ages are capable of engaging in complex logical reasoning and algorithmic problem-solving if provided with the appropriate tools and training. By introducing simplified programming environments like Scratch or our custom pseudo-code environment, students were able to apply their abstract reasoning to real-world programming challenges. We believe that initiatives like the Giochi di Fibonacci contest are crucial in preparing the next generation for an increasingly technology-driven society. We hope that our experience can be helpful for others to create similar programs that promote computational thinking and programming skills among young students.

References

Audrito, G., Luigi, W. di, 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., Beresnevičius, G., Dagienė, V. (2016). Learning programming through games and contests: overview, characterisation and discussion. *Olympiads in Informatics*, 10, 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.

Devlin, K. (2011). *Leonardo and Steve: The Young Genius Who Beat Apple to Market by 800 Years*. Independently published.

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

Fagerlund, J., Häkkinen, 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, S. (2017). Informatics and programming education at primary and secondary schools in Japan. *Olympiads in Informatics*, 11, 143–150.
Kiryukhin, V.M., Kinder, M.I., Cvetkova, M.S., Borisov, N.A., *et al.* (2022). Methods of tracks for training juniors in Olympiad Informatics: The ISIJ experience. *Olympiads of Informatics*, 16, 75–87.
Pluhár, Z. (2021). Extending computational thinking activities. *Olympiads in Informatics* 15, 83–89.
Souza, I.M.L, Andrade, W.L., Sampaio, L.M.R., 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. 1–9.
Vegt, W. van der. (2016). Bridging the gap between Bebras and Olympiad: Experiences from the Netherlands. *Olympiads in Informatics*, 10, 223–230.

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.

M. Ciobanu is involved in Italian Olympiads in Informatics since 2009. She got a Ph.D in Computer Science in the University of Salerno, and currently she is a teacher in a high school.

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.