A TEST DEVELOPMENT STUDY TO MEASURE COMPUTATIONAL THINKING SKILLS OF 2ND GRADE PRIMARY SCHOOL STUDENTS

SENA ÖZTÜRK SÖYLEMEZ

BOĞAZİÇİ UNIVERSITY

2023

A TEST DEVELOPMENT STUDY TO MEASURE COMPUTATIONAL THINKING SKILLS OF 2ND GRADE STUDENTS

Thesis submitted to the

Institute for Graduate Studies in Social Sciences

In partial fulfillment of the requirements for the degree of

Master of Arts

in

Educational Technology

by

Sena Öztürk Söylemez

Boğaziçi University

2023

DECLARATION OF ORIGINALITY

I, Sena Öztürk Söylemez, certify that,

- I am the sole author of this thesis and that I have fully acknowledged and documented in my thesis all source of ideas and words, including digital resources, which have been produced or published by another person or institution;

- this thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;

- this is a true copy of the thesis approved by my advisor and thesis committee at Boğaziçi University, including final revisions required by them.

Signature: ______________________
Date: ___________________________

ABSTRACT

A Test Development Study to Measure Computational Thinking Skills of 2nd Grade Primary School Students

The study aimed to develop an assessment tool for measuring computational thinking (CT) skills of 2nd grade Turkish students and to investigate its factor structure. The test was based on the three cognitive dimensions of CT, namely algorithmic thinking, pattern recognition, and debugging, which were measured using content dimensions of CT (sequencing, loops, and conditionals). A 30-item item pool was created and subjected to expert and student reviews before being piloted twice. The first pilot test was implemented with 291 students and consisted of 24 items. The second pilot test consisted of 18 items and was implemented with 377 participants. The final test included 13 items that were identified only with their cognitive dimensions, eliminating the content dimensions. It was implemented with 210 students from 10 different schools. Confirmatory factor analysis showed that the proposed three-factor structure did not fit the data. And exploratory factor analysis revealed a two-factor structure based on CT concepts and CT practices of the CT Framework (Brennan & Resnick, 2012). The study concluded that a more effective assessment of CT skills for 2nd grade students may involve focusing on the two-factor structure and differentiating items based on the CT concepts and CT practices from Brennan and Resnick’s CT Framework (2012). The findings suggest that participants' lack of explicit instruction in CT at their schools and the utilization of Scratch visuals and block-code-based structure may have contributed to the structure's deviation from the proposed three-factor model.

ÖZET

İlkokul İkinci Sınıf Öğrencilerine Yönelik Bilgi-İşlemsel Düşünme Becerisi Testi Geliştirme Çalışması

Bu çalışma, 2. sınıf ilkokul öğrencilerinin bilgi-işlemsel düşünme (BD) becerilerini ölçmek için bir değerlendirme aracı geliştirmeyi ve bu aracın faktör yapısını incelemeyi amaçlamaktadır. Test, BD'nin üç bilişsel boyutu olan algoritmik düşünme, örüntü tanıma ve hata ayıklama üzerine kurulmuştur ve bu boyutlar, BD'nin içerik boyutları olan sıralama, döngüler ve koşullar kullanılarak ölçülmüştür. 30 maddelik bir madde havuzu oluşturulmuş ve iki kez pilot uygulama yapılmadan önce uzman ve öğrenci incelemelerine tabi tutulmuştur. 24 madde içeren ilk pilot test 291 öğrenciyle uygulanmıştır. İkinci pilot test 18 madde içermektedir ve 377 öğrenci ile gerçekleştirilmiştir. İkinci pilot testte sadece BD'nin bilişsel boyutları olan algoritmik düşünme, örüntü tanıma ve hata ayıklamaya odaklanılmıştır. Son test, 13 maddeden oluşmaktadır ve 10 farklı okuldan 210 öğrenciyle uygulanmıştır. Önerilen üç faktörlü yapı, doğrulayıcı faktör analizine göre verilere uyum sağlamamıştır. Açımlayıcı faktör analizi ise iki faktörlü bir yapı ortaya koymuştur. Çalışma, 2. sınıf öğrencilerinin BD becerilerinin daha etkili bir şekilde değerlendirilmesinin, iki faktörlü yapıya odaklanmayı ve maddeleri Brennan ve Resnick (2012)'in BD çerçevesindeki BD konseptleri ve BD uygulamaları olarak farklılaştırmayı gerektirebileceği sonucuna varmıştır. Bulgulara göre, katılımcıların okullarında BD konusunda ders sonrası etkinliklere katılmaları ve Scratch görselleri ve blok kod yapısının kullanımına aşina olmamaları, yapılan üç faktörlü modelden sapmaya neden olmuş olabilir.

ACKNOWLEDGEMENTS

I would like to express my deep and heartfelt gratitude to all the individuals who have contributed to the completion of this thesis. Foremost, I extend my sincere thanks to my thesis advisor, Assist. Prof. Duygu Umutlu for her guidance, expertise, and belief in me during my research. Her constructive feedback and constant encouragement have motivated me to strive for excellence. I am also thankful for my thesis co-advisor, Assoc. Prof. Serkan Arıkan, for his contributions and support. I am genuinely grateful for the opportunity to work under their mentorship.

To my family, I owe a debt of gratitude for their love and patience. I especially want to thank my mother, Sema Öztürk, and my father, Mehmet Öztürk, who have been shaping the person I am today. They are my luck, and I am forever indebted to them. I am thankful to my dear sister, Zeynep Öztürk, for sharing my dreams, and to my dear brother, Hüseyin Mete Öztürk. To my husband, Hasan Emre Söylemez, my partner in life, I am profoundly grateful to him for being a source of strength for me during the challenges of this journey. Their support and understanding have meant the world to me.

Finally, I would like to acknowledge my committee members, Assist. Prof. Mutlu Şen-Akbulut, Assoc. Prof. Sevda Yerdelen Damar, and Prof. Dr. Filiz Kalelioğlu, for their time, effort, and constructive feedback. To everyone mentioned and to all those who have supported me behind the scenes, my BFF Feyza Kırcalar, my angel of help İlke Topal, and CET teachers, thank you for being part of this endeavor. Your contributions have left an indelible mark on my academic and personal growth, and I am honored to have had your unwavering support.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Computational thinking (CT) is a critical ability that consists of a group of key skills in today's digital world (Denning, 2010; Doleck, Bazelais, Lemay, Saxena, & Basnet, 2017; Grover & Pea, 2013; Wing, 2006; Wing, 2010; Wing, 2017). These skills involve breaking down complex problems into smaller, more manageable parts, identifying patterns and relationships, developing algorithms and step-by-step procedures, and testing and refining solutions (Denning, 2010; Grover & Pea, 2013; Wing, 2006). CT skills are essential for success in a wide range of fields, from computer science and engineering to medicine and finance (Barr & Stephenson, 2011; Bers, Gonzalez-Gonzalez, & Armas-Torres, 2019; Denning, 2010; Grover & Pea, 2013). CT skills promote innovation and creativity, data literacy, and problem-solving skills (Barr & Stephenson, 2011; Denning, 2010; Grover & Pea, 2013; Voogt, Fisser, Good, Mishra, & Yadav, 2015; Wing, 2006). As such, it is important for educators and policymakers to prioritize the development of CT skills in students from a young age (Barr & Stephenson, 2011; Grover & Pea, 2013).

According to Kanaki and Kalogiannakis (2019), it is recommended that students be introduced to CT during their early years. Several studies have been conducted to develop effective CT lessons for young learners (Çakır et al., 2021; Kalelioğlu, Gülbahar, & Kukul, 2016; Relkin, de Ruiter, & Bers, 2020). Bers, Flannery, Kazakoff, and Sullivan (2014) and Relkin et al. (2020) are examples of those studies. Given the importance of CT skills in the modern era, it is now being incorporated into national curricula across the globe (Babazadeh & Negrini, 2022; Weintrop, Wise Rutstein, Bienkowski, & McGee, 2021). In Turkey, for example, CT courses are compulsory for fifth and sixth grades (Ministry of National Education [MoNE], 2018).

As CT continues to gain importance, the need for assessing this skill is becoming increasingly important (Gülbahar et al., 2019; Weintrop et al., 2021). Assessing CT skills is also essential in promoting their development. The use of assessments can help educators determine the effectiveness of teaching strategies and identify areas where students need additional support for effective CT (Voogt et al., 2015). Assessments can also help identify students who may need additional challenges and opportunities to apply their CT skills in real-world settings (Grover & Pea, 2013).

Measuring a thinking skill like CT is challenging, but it is possible through observable indirect methods and approaches (Gülbahar, Kert, & Kalelioğlu, 2019). Although there are some research studies that aim to assess CT, the number of tests for CT is limited compared to assessments of other fundamental skills like math and literacy (Ruiter & Bers, 2021). Moreover, assessing CT in primary schools has been rarely investigated so far (Zhang & Nouri, 2019). In this area, the most relevant and recent study is the Beginner CT Test (BCTt), developed for first-grade primary school students (Zapata Cáceres, Roman-Gonzalez, & Martín-Barroso, 2021) and Computational Thinking Test (CTt) was developed for middle school students (Román-González, Pérez-González, & Jiménez-Fernández, 2017). As seen, a test for second-grade students is not developed, highlighting the need for a valid and reliable test for this age group (Demir & Seferoğlu, 2019).

Thus, the purpose of this study is to develop a test for assessing CT skills of 2nd-grade students in Turkey. The findings of this study are expected to contribute to the existing literature on CT assessment and provide valuable insights for educators and policymakers in enhancing CT education in primary schools.

1.1 Significance of the study

In recent years, the development of CT has become an important goal for education systems around the world (Brennan & Resnick, 2012). There are so many materials to implement CT into classrooms. Also, many researchers work on CT skills, try to explain it, and establish methods to improve it. Yet, unfortunately, there is a small number of studies that focus on the assessment of CT (Babazadeh & Negrini, 2022; Demir & Seferoğlu, 2019; Eryılmaz & Deniz, 2019). As the use of CT at primary schools increases, so does the demand for appropriate assessment tools for CT in early education context (Weintrop et al., 2021).

Babazadeh and Negrini (2022) examined 26 studies conducted in Europe from 2016 to 2020. They found that tests/tasks, questionnaires, observations, interviews, and product analyses are the most frequently used methods to assess CT skills of learners. As for tests or tasks, CT Test (Roman-Gonzalez et al., 2017) research and Bebras tasks published on the website https://www.bebras.org, each of which have been used in 5 research studies. Researchers are more inclined to use the tests or tasks for CT assessment. Out of 26 studies, only 3 were for the assessment for 2nd graders' CT. Two of them were task-based, and one of them was product analysis (Babazadeh & Negrini, 2022). Eryılmaz and Deniz (2019) investigated the studies about CT. They found that semi-structured observations and interviews are the most used data collection tools in Turkey. They also found that studies are mostly quantitative studies. Tang, Yin, Lin, Hadad, and Zhai (2020) found that in their systematic review study, most research studies were about improving CT skills of primary and secondary school students. And because of students’ low literacy skill level, it is hard to assess CT skills of young students.

Assessment of students' CT skills in primary schools is essential for the reason that it helps to examine their growing cognitive abilities (Babazadeh & Negrini, 2022). Children in this age group are at a critical stage of cognitive development, where their problem-solving skills are rapidly evolving. CT assessments can help educators understand how students approach problems, identify patterns, and think systematically. Additionally, CT assessments can help educators identify students who may be struggling with these skills and provide targeted support to encourage them to catch up with their peers (Barr & Stephenson, 2011).

Moreover, CT assessments in primary schools can help identify future science, technology, engineering, and mathematics (STEM) leaders (Kafai & Proctor, 2021). Early assessment of CT skills can help educators identify students who show potential for future success in STEM. These assessments can identify students who possess the essential skills for success in these fields, including problem-solving, critical thinking, and creativity (Grover & Pea, 2013). By identifying these students early on, educators can provide targeted support to help them develop their CT skills further, setting them on a path to success in STEM fields.

Assessing CT skills of second-grade students can also help with the development of CT curricula. Assessing students’ skills can help educators identify areas where students may need additional support (Mullis, Martin, Ruddock, O’Sullivan, & Preuschoff, 2009). These assessments can help shape the development of CT curricula, ensuring that it addresses the needs of students and providing them with the necessary skills to succeed in the 21st century (Grover & Pea, 2013).

As Relkin, Ruiter, and Bers (2020) stated, various instruments have been developed to assess CT in recent years but very few of them focused on primary school students, especially first and second graders. According to Demir and Seferoğlu (2019), there is a need for a valid and reliable achievement test to assess CT in Turkish language. There is no available test to measure 2nd graders' CT skills in Turkish.

The results of current study may have practical implications for teachers, curriculum developers, and policymakers, who can use the test results to identify areas where students need additional support, adapt teaching strategies to better align with students' learning needs, and design educational interventions to foster CT skills (Tang et al., 2020). As such, this study's findings would contribute to the overall goal of enhancing the quality of computer science education in primary schools in Turkey. The Ministry of National Education (MoNE) of Turkey has recognized the importance of CT skills and integrated them into the national curriculum for primary and secondary education (MoNE, 2018). However, there is a lack of available tests to assess CT skills among primary school students, particularly those in the early grades (Demir & Seferoğlu, 2019). The current study was conducted to fill this gap.

1.2 Purpose of the study

In the current study, second graders' CT skills were assessed. The proposed CT dimensions were algorithmic thinking, pattern recognition, and debugging. The purpose of current study was to develop a test for assessing CT skills of 2nd grade students in Turkey.

CHAPTER 2

REVIEW OF THE LITERATURE

2.1 CT

CT has a history in computer science and mathematics, although it is a comparatively recent concept and pedagogical framework (Papert, 1980). One of Seymour Papert's (1980) strongest theories is allowing students to program a computer and think like a programmer. According to Papert (1980, p.19), "Thinking about thinking makes kids into epistemologists." Students take an active role as learners as a result of this. Papert (1980) also investigated how skills in programming affect pupils' perspectives on life. Papert defined computer programming skills as playing with codes, monitoring the process, working in steps, and solving problems analytically. Papert did not label this skill despite his many studies on teaching programming to children and the cognitive benefits of studying programming.

Wing (2006) was the first to use the term CT. It is one of the key abilities of this century's generation, according to her (Wing, 2006). It is programming thinking, but it is one of the major skills that everyone, not just computer programmers, must learn (Wing, 2006). It is also required not only for programming but also for reading, writing, and arithmetic. The expanding emphasis on science, technology, engineering, and mathematics (STEM) education in the United States and other countries has contributed to a growing interest in computational thinking as a framework for teaching and learning (Guzdial, 2010). Several frameworks for teaching computational thinking, including the K-12 Computer Science Framework, the Computational Thinking Framework for K-12 Education, and the Exploring Computer Science curriculum, have been developed in recent years.

Wing (2010, 2017) states that computer science, mainly CT, is as fundamental as arithmetic (mathematical reasoning) and reading.
And programming and CT are abilities that should be gained gradually similar to teaching numbers in kindergarten, algebra in primary school, and calculus in college (Wing, 2017). According to Brennan and Resnick (2012), CT has become an increasingly popular subject in recent years. With computers getting into all aspects of life, thinking like a programmer, and applying this type of thinking skill in other areas of life is becoming an essential ability. CT provides numerous advantages, including improved problem-solving abilities, creativity, and digital literacy (Sengupta, 2016; Voogt et al., 2015). Therefore, CT becomes a talent that everyone should acquire at a young age (Barr & Stephenson, 2011). Every student, according to Voogt et al. (2015), must have competency in CT. CT is an essential skill for the twenty-first century, according to educational technology specialists and educators (Voogt et al., 2015).

CT is defined by Wing (2006) as "thinking in terms of prevention, protection, and recovery from worst-case scenarios through redundancy, damage containment, and error correction." (p.34). CT is the use of computer science visions for problem-solving, designing systems, and understanding how people think. It refers to a set of logical thinking processes that include mathematical, scientific, and engineering reasoning (Wing, 2008). There is no consensus on how to define CT (Zapata- Cáceres et al., 2021). Doleck and his colleagues (2017, p.2) define CT as "an umbrella term" that encompasses some cognitive abilities and computational concepts. CT is explained by International Society for Technology in Education

(ISTE) (2015) as a process that includes algorithmic and analytical thinking, imaginative, problem-solving, and cooperation abilities. CT, according to Voogt et al. (2015), is more about the process of finding, generalizing, and transferring a problem solution. CT appears to be algorithmic thinking, but it also incorporates mathematics, engineering, and scientific reasoning (Bers et al., 2019). Because CT is a broad term, it includes many components.

2.2 Computational thinking components

CT is the major skill that every child must gain in the 21st century (Wing, 2006). Being a computational thinker is not just being a programmer. It is the way of thinking like a programmer through the processes of abstraction, reasoning, error correction, problem-solving, and combining multiple ways of thinking: engineering, and mathematics. Abstraction is a core element of CT (Wing, 2008). Wing (2008) defines abstraction as an algorithm that takes an input and makes it a wanted output in a way that implements it step by step. It also explains patterns and generalization the algorithm (Wing, 2010). CT is related to thinking about the process of problem-solving, investigating the way that is used to solve the problem, making a formula from it, and using that formula in any other problem (Wing, 2017).

Brennan and Resnick (2012) put forward that CT skills include creativity, critical thinking, debugging, problem-solving, decomposition, heuristic reasoning, abstraction, algorithmic thinking, cooperativity, data analysis, and recursive thinking. They categorized CT as computational concepts, computational practices, and computational perspectives in CT framework. Computational concepts are sequences, loops, events, conditionals, operators. Those are the basics of the programming. Computational practices are the skills that children develop while implementing computational concepts. They include being incremental and iterative, testing and debugging, reusing, and remixing, and abstracting and modularizing. Computational perspectives are expressing, connecting, and questioning.

According to Weintrop, Beheshti, Horn, Orton, Jona, Trouille, and Wilensky (2016), there is a need for defining CT and integrating it into Math and Science lessons. Therefore, Weintrop et al. (2016) offered a taxonomy for school Math and Science lessons. It consists of four main categories: data practices, modeling and simulation practices, computational problem-solving practices, and systems thinking practices. Based on this taxonomy, CT includes these four main categories.

Gülbahar et al. (2019) specified CT strategies as abstraction, data collection, pattern recognition, algorithm when they created items of the scale, they developed to measure self-efficacy in CT. More specifically, they had 5 groups of items: (1) designing algorithms, (2) problem-solving, (3) data processing, (4) basic programming, and (5) self-efficacy. To define CT, they used components which are data, abstraction, and modeling. Data consists of data analysis, data presentation, and parsing. Abstraction includes automation, algorithm design, and concurrent operation. And modeling involves pattern recognition and pattern generalization. Kalelioğlu et al. (2016) reviewed literature to define CT by setting the core components. They included 125 studies on CT in their review to identify its most common features. As a result, they indicated abstraction, problem solving, algorithmic thinking, pattern recognition, and design-based thinking as the core components of CT.

Zapata Cáceres et al. (2021) developed a valid and reliable instrument for measuring CT skills of early age students. Items were created based on several computational components which are sequences, loops (simple and nested loop), and conditionals (if-then, if-then-else, and while). The results showed that the test items are usable for first graders and there is no significant result for older students. After CT entered K-12 curricula, elementary and secondary schools needed modules to implement and evaluate CT. To address this need, Yadav, Zhou, Mayfield, Hambrusch, and Korb (2011) prepared a module that includes definitions of CT and its five basic components. Basic components of CT in this module were abstraction, logical thinking, algorithms, debugging, and problem identification and decomposition.

Tsai, Wang, and Hsu (2019) developed a scale to measure computer programming self-efficacy of middle school students in computer literacy education. They described CT under five subscales. Those are Logical Thinking, Cooperation, Algorithm, Control, and Debug. In another study, Tsai, Liang, and Hsu (2021) investigated computer literacy and CT. They created a CT scale for computer literacy education by examining CT in two aspects. Tsai et al. (2021) divided CT definitions into two groups. First one was related to programming and computing concepts. The other one was related to the competencies for both general and specific domains. The first one was named as domain specific and the second one was named as domain general. Moreover, Tsai, Chien, Wen-Yu Lee, Hsu, and Liang (2022) developed a CT test for elementary students. In that study, they described CT within domain specific and domain general definition groups, and they focused on domain general type of CT in a non-programming context. The test measures domain general CT through decomposition, algorithm, abstraction, generalization, and evaluation subskills.

An international mathematics and science ability assessed test, Trends in International Mathematics and Science Study (TIMSS), used a mathematics/science assessment framework that divided the assessed concept into two dimensions content dimension and cognitive dimension (Mullis & Hooper, 2015; Mullis et al., 2009). The mathematical and scientific subjects were the content domain, and mental skills were the cognitive domain. Plus, the cognitive domain was divided into three categories: knowing, applying, and reasoning. Likewise, Tang et al. (2020) also used a two-group structure and divided CT into two groups: first group was representation of CT skills which are coding concepts, and the second group was cognitive CT skills that were used in coding.

Current study maintains CT as a two-dimensional structure. As it is in CT framework, CT concepts and CT practices are used. CT practices are thinking skills, and the CT concepts are programming skills. Thinking skills that CT covers are named cognitive dimensions of CT, and programming skills are content dimensions of CT (Tsai et al., 2022). The main aim of this test to measure the cognitive dimensions of CT. To test those, CT content dimensions are used as the content of the test.

2.3 Definitions of cognitive dimensions of CT

2.3.1 Algorithmic thinking

According to Mehri (2017), the word “algorithm” was first used by an Arab mathematician Al-Khwarizmi. It means a procedure that involves a bunch of steps in order to find a solution to a specific problem. It is one of the bases of programming. Futschek (2006) defined an algorithm as a problem-solving method that includes all the steps and instructions needed precisely. Algorithmic thinking consists of the abilities such as analyzing and defining problems, stating the steps that are adequate for the given problem, constructing an algorithm, considering all possible ways for solving the problem, and selecting an efficient one. The term algorithm is shown as a gradually designed procedure to accomplish a task not only in Computer Science but also in other disciplines (Selby, 2014). Algorithm design describes creating a pathway with all the steps of instructions to solve similar problems or overcome a task (ISTE, 2021). Also, Wing (2008) stated that an algorithm is designing a simple step-by-step process to reach a goal at the end. As Barr and Stephenson (2011) indicated, K-12 students already know how to solve problems but by using algorithms, they can identify where to compute and manipulate data like computers. In this test, the term "algorithmic thinking" is regarded as an ability to develop a pathway, an algorithm, with step-by-step commands to solve problems. If a programmer wants to compute a computer with one exact goal, he/she should specifically explain how to do it in detail. For example, when the programmer wants a player in a computer program to go to the ball, he/she should identify the way that player goes step by step until it reaches the ball. This identification process of the pathway is called an algorithm of the program (reaching-the-ball program). Algorithmic thinking is thinking based on these pathways, finding efficient pathways, and selecting the best one to solve the problem.

2.3.2 Pattern recognition

Some algorithms produce repetitions due to their shared objectives and requirements. Algorithms that are used repeatedly to solve problems are known as algorithmic patterns. Because programmers share knowledge of solutions and reuse best practices, pattern recognition is crucial (Muller, 2005). Sometimes programmers find a solution for a specific problem but the way used to deal with that problem can be helpful for solving other similar problems. Pattern recognition consists of recognizing this and integrating that code into different situations. According to Selby (2014), pattern recognition is the term used for identifying patterns in algorithm and problem-solving processes. It is a generalization of the solution method to other problem-solution processes. ISTE (2021) defined pattern recognition as observing similarities, trends, regularities in data as patterns. In this study, “pattern recognition” is used as the identification of the similarities in problem-solving steps of different or same problems, solution processes, and using those solution steps for different relevant problems. The programmer can use similar code commands when going two different places (for example, using two times of "move forward and turn right" commands). The programmer may also recognize repeated commands in the same program. For example, a character may follow these steps: turn right - move forward - turn right - move forward. This means use of the "turn right - move forward" command twice. Noticing and identifying these similar commands is called pattern recognition.

2.3.3 Debugging

Learning computer programming inherently requires debugging. It is the process of locating the bugs and fixing the bugs that keep a program from operating as expected. (Morales-Navarro, Fields, & Kafai, 2021). In programming, debugging is the commonly used strategy while writing codes. According to Araki, Furukawa, and Cheng (1991), programmers use debug statements to observe how a program works step by step. This helps them to catch errors. Here, the error means the unexpected behavior that the program exhibits. And bug means the cause of the error. Debugging is the process for fixing these bugs (Araki et al., 1991). Debugging is an iterative process of applying different types of knowledge, such as domain knowledge, system knowledge, and strategic knowledge, to form and test hypotheses, and to reduce the problem space until the bug is located (Li, Chan, Denny, Luxton-Reilly, & Tempero, 2009). In education, according to Papert (1980), students can learn from their mistakes by correcting them like programmers. Correcting bugs makes children not afraid of being wrong and gives them the power to fix their mistakes. Debugging is the strategy to think about the failures. It is the process that students think what happens, what is wrong, and understand the problem and fix it (Papert, 1980). According to Kim, Yuan, Vasconcelos, Shin, and Hill (2018), debugging is the basis of programming and is essential for computer science as a part of problem-solving. Kafai, DeLiema, Fields, Lewandowski, and Lewis (2019) defined debugging as using failure to be productive. Debugging consists of the ability to think about the overall system that programmers are building. The idea behind the programming process is that bugs are a necessary element of productive failure. These may help us gain understanding of the issue at hand and find better, more appropriate solutions.

Another important process in computer programming is testing. Testing is the process that tests the program to find bugs and ensure its proper functioning. As Brennan and Resnick (2012) explain, things rarely work just as imagined in programming; hence, designers must develop strategies for dealing with problems and anticipating them. Programmers employ testing techniques to assess the behavior and performance of their programs. This involves running the program with different inputs, evaluating the output, and comparing it to the expected results. By systematically testing the program, developers can identify and locate bugs, ensuring that the program functions as intended. Testing is an iterative process that helps programmers refine their code and eliminate errors. Through trial and error, programmers can observe how their program behaves and identify areas that require improvement or correction.

The term "debugging" is defined as the ability to test the code and fix errors in the problem-solving process in this test development study. Programmers may test the program or trace the algorithm step by step, recognize the errors in it, establish cause-effect relationships while identifying the error, and correct the error properly in the debugging process. Debugging is a process that involves locating errors in codes and fixing them. For example, when a bee goes to rocks instead of flowers, programmers examine the code of the bee and find where the mistake is. In the program, for instance, the bee goes right instead of going left and reaches rocks. Learners will find the wrong turn block (turn right block) and replace it with the correct block (turn left) through debugging. By systematically testing their programs and comparing the observed behavior with the expected results, programmers can refine their code and make sure that it works properly. This process is called debugging in this study.

2.4 Definitions of CT's content dimensions

2.4.1 Sequencing

Sequencing is purposefully constructing step-by-step algorithms. It is the core element of programming (Brennan & Resnick, 2012). Building a sequential problem-solving method is what sequencing entails in order to complete a certain activity (Threekunprapa & Yasri, 2020). Within the scope of this study, sequencing includes putting a series of individual command blocks in order (Parker, Kao, Satio-Stehberger, Franklin, Krause, Richardson, & Warschauer, 2021). Creating an algorithm by putting five "move forwards" blocks successively to move character 5 steps is an example of sequencing.

2.4.2 Loops

When some code blocks are repeated multiple times, there is a mechanism to write it once and repeat the same blocks multiple times instead of writing several lines of commands. This mechanism is called loops; and it is one of the key elements of programming (Brennan & Resnick, 2012). Loops help programmers to make codes shorter. This is a more practical way to represent problem-solving steps by repeating one command multiple times (Threekunprapa & Yasri, 2020). In this study, loops refer to repeating the same action multiple times by not writing lines of commands (Parker et al., 2021). Two types of loops will be used in this study: (1) simple loops and (2) conditional loops. Simple loops are repeating some code blocks inside the loop function multiple times (Bers et al., 2019). With this simple loop function, a programmer can show how many repeats will be done in the same sequence of codes. Conditional loops are the repetition of an action depending on a condition which can be true or false (Threekunprapa & Yasri, 2020). For example, a while loop repeats an action when the stated condition is true.

2.4.3 Conditionals

Another fundamental concept in programming is conditionals. It is used for decision-making under specific circumstances. And based on conditions, actions and commands are programmed (Brennan & Resnick, 2012). It is a function that decides to run or not to run a command depending on the absence or presence of the condition (Threekunprapa & Yasri, 2020). If statement is for running a sequence of codes if the stated condition is true. When there is an if statement, the program checks the condition before running the written code, and if the condition is met, then it starts to run the code. If there are extra conditions in the situation when the condition in the if statement is not true, then extra if is needed for those additional conditions. This other if is called else statement. If the "if" condition is not true, the program looks for an "else" condition.

2.5 Studies based on CT assessment

As the significance of CT grows, so does the demand for measuring this skill (Gülbahar et al., 2019). Even though it is difficult to measure a thinking skill, it is achievable through observable indirect methods and approaches (Gülbahar et al., 2019). Because of this, a credible and accurate test is required to assess CT skills (Demir & Seferoğlu, 2019). There are some multiple-choice tests to measure the CT skills of students of various ages.

Roman-Gonzalez et al. (2017) developed the Computational Thinking Test (CTt). It is the measurement of CT for middle school students. The test has 28 items and takes 45 minutes. They used Brennan & Resnick's (2012) CT framework. They used computer science concepts to assess the CT skills of students.

Zapata-Caceres, Martin-Barroso, & Roman-Gonzalez (2020) developed a valid and reliable instrument for measuring the CT of early-age students. They called the measurement the Beginners' Computational Thinking Test (BCTt). They used CT Framework while developing items as computer concepts (Zapata-Caceres et al., 2020). The test includes 25 items. They created it as 40 items with experts and did a pilot test. After the pilot study, there are 25 items in the second version of the test. Items were created according to computational concepts which are sequences, loops (simple and nested loop), and conditionals (if-then, if-then-else, and while). BCTt is an assessment instrument that is independent of any programming environment. Its statistical analysis and content validation showed the consistency of the test to assess CT in primary schools. Their population was students between age 5 to 12 but the research showed that it is too easy for age 9 to 12. The exam questions were significant for first graders, according to the results, but not significant for later children (Zapata Cáceres et al., 2021). BCTt's cronbach's alpha value is .82 for overall. According to first graders' test results, α is equal to 0.83 and it is getting lower when the grades getting higher (Cronbach alpha value of 2nd grade is .79, 4th is .77, 5th is .66).

El-Hamamsy, Zapata Caceres, Martín, Mondada, Dehler Zufferey, and Bruno (2022) developed Competent Computational Thinking Test (cCTt). It is adapted from BCTt (Zapata-Caceres et al., 2020), unplugged test which is third and fourth grades students. It has 25 items. Cronbach's alpha value of cCTt is .85. The test implemented with 1519 students can be used in low and medium ability level students (El-Hamamsy, Zapata-Cáceres, Marcelino, Bruno, Dehler Zufferey, Martin-Barroso, & Roman-Gonzalez, 2022).

Tsai et al. (2022) create a Computational Thinking Test for Elementary School Students (CTT-ES). It consists of 16 items. CTT-ES is for fifth and sixth graders. CTT-ES evaluates students' CT competencies (skills or performance) in a non-programming environment. It divided CT into five contexts: algorithm, abstraction, generalization, decomposition and evaluation. For item analysis, Tsai et al. (2022) used IRT and CTT.

Relkin et al. (2020) developed TechCheck assessment to measure CT skills of early primary school students. It is a multiple-choice test, and it uses an unplugged format. They developed TechCheck for first and second graders. After the implementation, they found that while TechCheck's difficulty level is appropriate for first graders, and it is easy for second graders. After this study, they conducted another one to measure CT in kindergarten: TechCheck-K (Relkin & Bers, 2021). It is a new version of TechCheck and has 15 unplugged multiple-choice questions. It is for students who do not have coding experience. CT is assessed by logical problems with pictorial puzzles.

Zhang, Wong, and Pan (2021) designed Computational Thinking Test for Lower Primary (CTt-LP). They viewed CT through these concepts; sequences, directions, loops, and conditionals. The test is still in the development process. The first version of the test contains 30 items. The pilot study was conducted, and Cronbach's alpha value of the test was .74. The participants of the test are 72 second-grade students in China. After the item analysis, three items were eliminated.

Batı (2018) created the Computational Thinking Test (CTT) for middle school students in Turkey. The test is in Turkish. He aimed to measure CT by using science and mathematics questions. CTT is suitable for eighth-grade students. Demir and Seferoğlu (2019) also created a Scratch-based coding achievement test for university students in Turkey. The test is in Turkish. There are 18 items in the test, and Cronbach's alpha reliability coefficient of the test is .80. There were 186 participants who were willing to take the test from Hacettepe University Computer Education and Educational Technologies (CET) department.

Çetin, Otu, and Oktaç (2020) conducted an adaptation study on CTT (Román-González et al., 2017; Román-González, 2015). They adapted the test into Turkish. The test is for middle school students (5th to 8th grades). There were 28 items and 502 participants. As a result of the study, the test contains 24 items. The original test's Cronbach's alpha value is .79, and the adaptation test has .77 Cronbach's alpha coefficient. The adaptation test consists of the following computational concepts: sequencing, loops, loops with conditions, conditions, basic functions.

In terms of participant demographics, research has shown that CT assessments are not biased against certain groups, such as gender or socioeconomic status (Dagiene & Jevsikova, 2014). However, studies have found that younger children, such as those in K-2, may struggle with certain CT concepts, such as abstraction (Bers et al., 2014). Therefore, it is essential to use CT assessment tools that are appropriate for each age group to obtain accurate results. Assessment of CT in primary schools is one of the recent topics that needs further investigations. In addition, most of these CT tests developed for young learners are in English. Therefore, there is a need for a test in Turkish for young learners, such as 2nd grade students in Turkey.

2.6 Usage of scratch

Scratch is a visual programming environment that allows students to create projects such as animated graphics, video games, and music projects (Maloney, Resnick, Rusk, Silverman, & Eastmond, 2010). Sound and photos can be embedded into it. Students can program without any typing skills within the block-based structure of the environment (Aytekin, Sönmez Çakır, Yücel, & Kulaözü, 2018). Scratch is one of the most effective environments to teach computer programming concepts to primary school students and improve their CT skills as it is a block-based coding environment (João, Nuno, Fábio, & Ana, 2019). Because of these properties, it is the most popular environment in programming education for early grades (Demir &

Seferoğlu, 2019; Eryılmaz & Deniz, 2019). Moreover, primary school students found it easy to use (Kalelioğlu & Gülbahar, 2014). Because this measurement study is implemented in Turkey with Turkish students, Scratch is used in Turkish. It is important that the interface has Turkish language option so that screenshots can easily be used in test items. For this reason, it is getting ahead of its competitors in Turkey (Karabak & Güneş, 2013). Although it is used widely, there is a need for a valid and reliable assessment tool to measure CT skills of students by using Scratch (Demir & Seferoğlu, 2019).

For items in this test development study, the Scratch platform is used. There are screenshots and code representations from Scratch. Scratch is selected because it is visual, block-based, user-friendly, and appropriate for primary school students.

CHAPTER 3

METHOD

Creswell (2012) defines the test development process in four basic phases which are literature review, item writing, implementation of the test, and validating process. In current study, these steps were followed to develop a test for CT. In the first phase, the purpose of the test and target group were defined. The dimensions of the CT were identified after reviewing several researchers' (e.g., Brennan & Resnick, 2012; Weintrop et al., 2016; Wing, 2006; Zapata Caceres et al., 2021) definitions of CT in different contexts. Moreover, measuring the CT skills is a topic that still needs to be investigated especially for second graders. After an extensive literature review, three CT components and three CT concepts were selected as the dimensions of the present study. The aim of the study was to develop a test for assessing 2nd graders' CT skills through CT components. Cognitive dimensions of the test were CT components which were algorithm, pattern recognition, and debugging. Content dimensions of the test were CT concepts that were sequencing, loops, and conditionals.

3.1 Participants

The aim of current study was to develop a test to assess 2nd grade primary school students' CT skills. To achieve this aim, the data were collected from 2nd grade students. After taking permission from the MoNE (see Appendix A), the researcher took school principals' permissions to implement the study. Participants were from public schools which approved the study implementation.

In the piloting process (in the first and second pilot study), participants' prior knowledge of computer science is not the elimination criteria. All the participants in piloting process were selected by using convenience sampling. Participants were selected in this sampling method because they were available and willing to attend the study. Participation in the study was voluntary. The test was given at participating public primary schools during school time in participants’ own classrooms.

However, after the second pilot test, it seemed that test was suitable for students who take computer science lessons. In the final study, participants were selected from the students who took computer science lessons with purposive sampling. Purposeful sampling is a non-probability sampling technique where the researcher intentionally selects individuals who meet specific criteria or have certain characteristics (Creswell, 2012). In the current study, participants were selected from the students who were taking computer science lessons because the test would be more suitable for this group of students. This decision was made based on piloting results that the test was more appropriate for students who had some background knowledge or experience in computer science. By selecting participants who took computer science lessons, the researcher was able to better control for the potential influence of prior knowledge or experience on test performance.

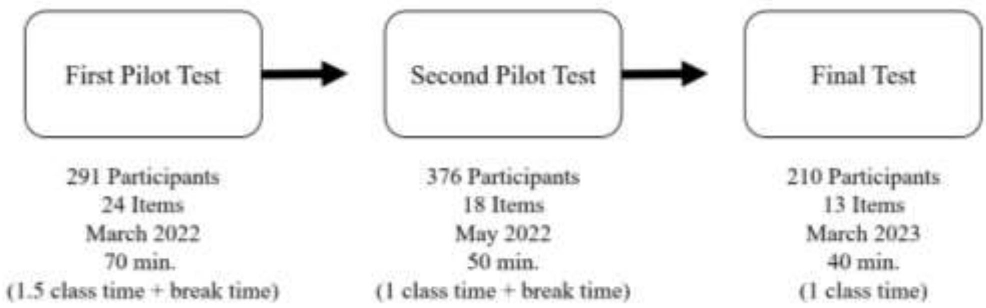


Fig. 1 Test implementation overview

The number of participants was determined according to the number of test items. Having at least 20 students per item (Crocker & Algina, 1986) and minimum of 150 participants are suggested (Pallant, 2016). There were 24 items in the first pilot version of the test (see Appendix B). Therefore, the number of participants for the first pilot study is suggested to be around 480. However, in the implementation process of first pilot study, only 291 participants were willing to give permission to use data in the analysis (see Fig. 1). 134 of them never took any coding or robotic courses. 157 participants stated that they took coding courses. Participants were generally between the ages of 7 and 8 when the test was administered in March 2022. The data collected from 5 different public primary schools in İstanbul.

The second pilot test was conducted with 376 participants in May 2022. The test consisted of 18 items (see Appendix B). 125 participants have taken a coding class, and 251 of them have not taken any coding classes. The data was collected from 7 different schools.

In the final study, there were 13 items and 210 participants (see Appendix C). The implementation of the test was completed in March 2023. The data was collected from 10 different schools. All of the participants have taken coding classes in their schools.

3.2 Item Writing

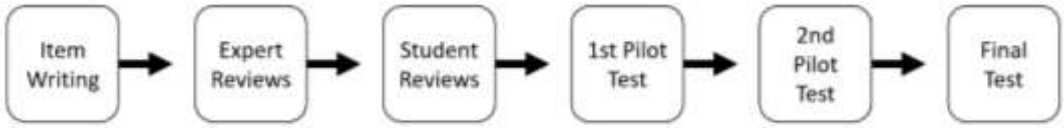


Fig. 2 Procedure

The procedure of test development process in this study shown in Fig. 2. For constructing the test, operational definitions were established to turn constructs into measurable variables (Crocker & Algina, 1986). CT was defined with domain specific definitions and domain general definitions (Tsai et al., 2021). Domain specific definitions were based on computer programing content dimensions such as sequencing, loops, and conditionals (Brennan & Resnick, 2012). And domain general definitions were about the thinking processes as cognitive dimensions like algorithms, pattern recognition, and debugging. A table of specifications was developed related to these two categories (see Table 1). Items were written based on the table of specifications. In the beginning of the study, there were 30 items. Every item had both content and cognitive dimensions itself. The test was structured as measuring cognitive dimensions by using content dimensions in the items. For cognitive dimensions, 13 items were developed for assessing Algorithmic Thinking, 10 items were for Pattern Recognition, and 7 items for Debugging. In content dimensions, 7 items were written for assessing the sequencing, 13 items for loops, and 10 items for conditionals (see Table 1).

Table 1. Table of Specifications for the First Version of the Test

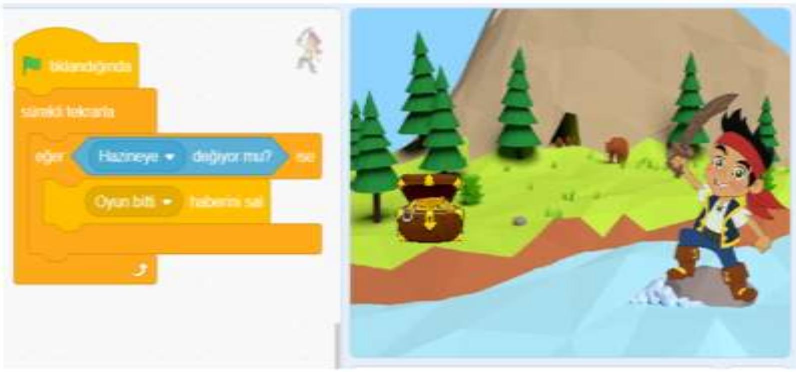
Cognitive Dimensions / Content Dimensions	Algorithmic Thinking	Pattern Recognition	Debugging
Sequencing	I1, I3, I5, I8	I7	I14, I29
Loops	I2, I13, I17	I4, I9, I10, I11, I12, I18, I30	I6, I19, I27
Conditionals	I15, I16, I21, I22, I23, I26	I20, I24	I25, I28
Total Number of Items	13	10	7

3.2.1 Expert Reviews

The first version of the items was revised based on experts’ reviews. The test was sent to three experts: one primary school teacher who taught second graders at the time of the study, and two computer science teachers in a primary school. As a result of their views, 6 items were eliminated. The rationale behind this elimination is that the test duration was too long, and there were too many items for second graders' attention span. Experts evaluated the items according to their congruence with the proposed dimensions. When the experts were evaluating the items, they were comparing them to the proposed dimensions or constructs that the test was intended to measure. The proposed dimensions might have been based on a theoretical framework or previous research in the field. The experts looked at each item and determined if it was aligned with the intended construct or not. For example, if the proposed dimension was "algorithmic thinking", the experts would assess whether each item actually tested algorithmic thinking or not. This process helped ensure that the items were measuring what they were intended to measure, which is an important aspect of content validity. By evaluating the items in this way, the experts were able to provide evidence for the validity of the test. In this way, content validity was evaluated, and validity evidence was provided (see Table 8).

In Figure 3, the question asks, “According to the code that makes the pirate move, which situation will end the game?”. The test option a is the pirate touches the treasure, b is the pirate touches the mountain, and c is the treasure touches the water. In this question, the item did not measure what it aimed to measure because the right answer was obvious for the ones who even did not look at the code. The pirate had one aim which was to get the treasure. Because of this, this item was eliminated.

17.



Yukarıda korsanı hareket ettiren koda göre oyunu bitirecek durum aşağıdakilerden hangisidir?

a) Korsanın hazineye değmesi
b) Korsanın dağa değmesi
c) Hazinenin suya değmesi

Fig. 3 Example test item for eliminated items

3.2.2 Student reviews

The test was administered to three 2nd grades students, and afterward, interviews about the items were conducted. Items were revised to have clearer meanings for the age group based on their views. For example, in item 1 (see Figure 4), the question asks, “How does the gray dinosaur moving in the direction of the arrow go next to its friend, the red dinosaur?”. In this item, turn command was not clear for second graders because they got confused about the right and left directions. So, the item was changed into “turn to side.” This change was applied to all items in the test.

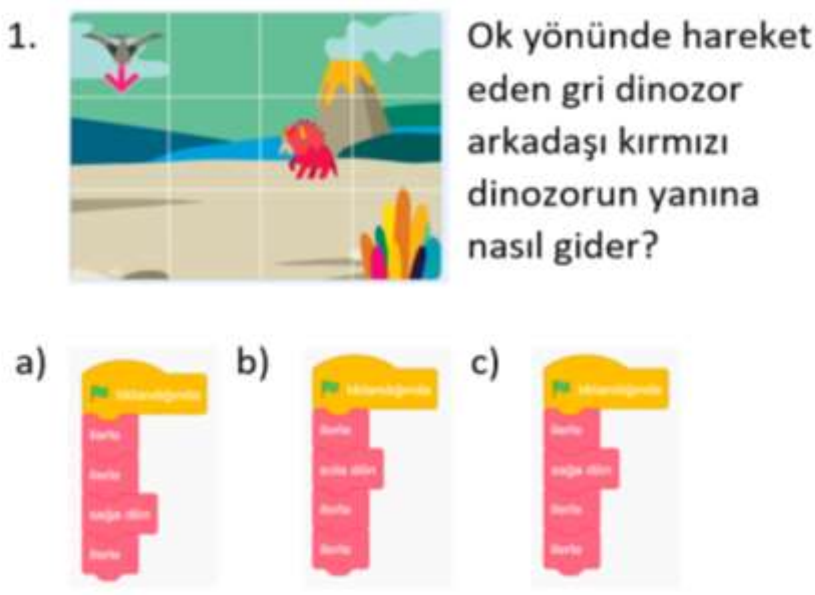


Fig. 4 Example test item for revised items

After the reviews of experts and students, items were revised and eliminated (see Table 2). For the first pilot test, there were 24 items: 9 for algorithmic thinking, 8 for pattern recognition, and 7 for debugging.

Table 2. Table of Specifications for the First Pilot Test

Cognitive Dimensions / Content Dimensions	Algorithmic Thinking	Pattern Recognition	Debugging
Sequencing	I1, I3, I6	I5, I24	I12, I23
Loops	I2, I9, I11	I7, I8, I10	I4, I15, I21
Conditionals	I13, I14, I16	I17, I18, I19	I22, I20
Total Number of Items	9	8	7

After the analysis of the first pilot test, items were eliminated and revised. Sequencing, loops, and conditionals were still used as a content of the items, but items were divided according to cognitive dimensions. Content dimensions were not used to group items (see Table 3). Second pilot test included 18 items.

Table 3. Table of Specifications for the Second Pilot Test

Cognitive Dimensions	Items	Total
Algorithmic Thinking	I1, I2, I4, I7, I9, I11	6
Pattern Recognition	I3, I5, I6, I12, I13, I18	6
Debugging	I8, I10, I14, I15, I16, I17	6

As the result of second test implementation, items were analyzed, revised, and eliminated. In the pilot implementations, the test took 2 hours to solve, and this amount of time was over the attention span of the students. Hence, there were 13 items (see Table 4), and the test took one hour to solve in the final test.

Table 4. Table of specifications for Final Test

Cognitive Dimensions	Items	Total
Algorithmic Thinking	I1, I2, I4, I11	4
Pattern Recognition	I3, I5, I8, I12	4
Debugging	I6, I7, I9, I10, I13	5

3.3 Data analysis

3.3.1 Reliability

The reliability of the test scores was evaluated to see the consistency of the test over the several administrations or with the similar tests (Crocker & Algina, 1986). The coefficient of internal consistency was computed to evaluate the degree of consistency of items with each other, from a single administration. (Crocker & Algina, 1986; Gay, Mills, & Airasian, 2012). The Cronbach's Alpha reliability coefficients were used to report the internal consistency of the data. Alpha was calculated with SPSS program. Alpha value 0.70 is the minimum expected value. It is considered as "acceptable." if the alpha value is between 0.70 and 0.80. If the alpha value is from 0.80 to 0.90, it is regarded as "good." The "excellent" value of alpha is above 0.90 (Cicchetti, 1994).

3.3.2 Validity

High reliability coefficient is evidence of consistency in participants' scores, but this alone does not necessarily demonstrate that the assessment tool has defensible implications. To ensure the soundness of inferences made based on test scores, it is necessary to evaluate validity of the assessment. Validity can be assessed in three categories: content validity, criterion validity, and construct validity. In the current study, content and construct validation procedures were used to assess how well the test correlates with the variables it was intended to measure (Cronbach, 1971).

3.3.2.1 Content Validation

Through the content validity, whether the test could measure the targeted skills were evaluated. The content validation process included defining the main interest of the test (Crocker & Algina, 1986). The current study defined CT with its cognitive dimensions. Those were algorithmic thinking, pattern recognition, and debugging. Secondly, the test was reviewed by experts from the field. The test was sent to three computer science teachers who teach primary school students. And then, whether item specifications fit the items or not was checked. Lastly, there was a quantitative resolution (Crocker & Algina, 1986). To find the content validity of an item for given objectives, the index of item-objective congruence (Rovinelli & Hambleton, 1977) was used. The experts gave points to items from -1 to 1. They gave -1 if an item did not measure the specification in the table. If it was fixed, they gave 1. And if it was not fully fixed but relevant, they gave 0. The index of item-objective congruence was calculated as $I_{ik}= N / (2N-2). (\mu_k - \mu)$. N is the number of objectives, μk is the mean rating of item i on kth objective and μ is the judges' mean rating of item i on all objectives. The highest possible value of the index is 1.00. If an item is matched more than one specification, the index is less than 1.00 (Crocker & Algina, 1986).

3.3.2.2 Construct Validity

Construct validity is important for the tests that measure not-directly-observable constructs such as creativity and intelligence (Crocker & Algina, 1986). Thinking skills are also the constructs that are not directly observable (Gülbahar et al., 2019). So, the construct validity evidence was collected. To collect evidence for the construct validity, factor structure of the test was evaluated.

3.3.2.2.1 Confirmatory Factor Analysis

A confirmatory factor analysis (CFA) which is a type of structural equation modeling was conducted to evaluate the fit of the proposed structure to the participants' answers (Bentler & Chou, 1987). Because this test had proposed structure, the CFA was used to confirm that structure. Steps of CFA are explained below (Schumacker & Lomax, 2010). Model specification is the first. The variables that include the analysis and the relationships between variables is determined by specifying the relationships. The following step is model identification. Here, the number of data points should be more than the parameters, which is called "overidentified". If not, parameters cannot be estimated. After that, third step is model estimation. Jöreskog (1994) suggested that when analyzing ordinal variables, Weighted Least Squares (WLS) should be used. This is due to the fact that ordinal variables are not continuous and should not be evaluated in this manner. Because ordinal variables lack sources and units of measurement, their means, variances, and covariances do not have an obvious meaning. The WLS method relies on polychoric correlation and an asymptotic covariance matrix to obtain accurate standard errors and chi-squares. If the optimal weight matrix is utilized, this method yields efficient estimates that include valid asymptotic chi-squares and standard errors (Aish & Jöreskog, 1990). Lastly, a model testing should occur. This is for testing whether a model is “good” or not. If the difference between sample covariance matrix and the population covariance matrix is minimum, the model is “good.” Comparative Fit Index (CFI), Tucker Lewis Index (TLI), and Root Mean Square Error of Approximation (RMSEA) are the fit indices that are mainly checked (Ullman, 2001). For RMSEA value, smaller than 0.080 is acceptable and 0.060 is preferable (Browne & Cudeck, 1993). CFI and TLI values more than 0.95 are evaluated as well (Hu & Bentler, 1998). When the data's CFI and TLI values are greater than 0.95 and the RMSEA is less than 0.06, the model fits the data well (Hu & Bentler, 1998).

Three-factor structure was proposed in the current study (see Figure 5). The first factor was algorithmic thinking, the second was pattern recognition, and the third one was debugging.

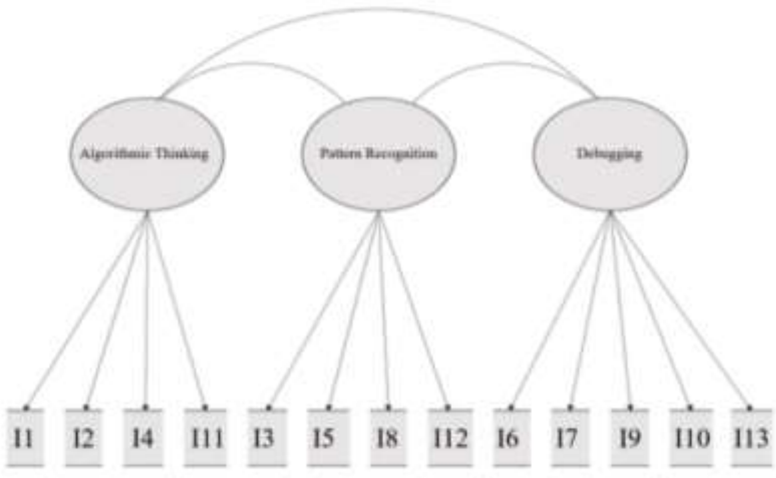


Fig. 5 Three factor model plot

3.3.2.2.2 Exploratory factor analysis

Exploratory factor analysis (EFA) is a statistical technique used to explore the underlying structure of a set of variables or items. It is used to explore the relationships between items and identify factors that are associated with one another. EFA is often used to investigate complex relationships between variables (Pallant, 2016).

To conduct EFA, researchers typically use a method called principal axis factor extraction, which is a technique that seeks to maximize the amount of shared variance between items. In addition, direct oblimin rotation is commonly used to obtain a more interpretable factor solution. If the factors are totally different from each other, varimax method can be used (Pallant, 2016).

During the analysis, researchers identify underlying factors and problematic items that have low factor loadings, indicating that they do not contribute significantly to any factor. A problematic item is an item whose item loading is 0.400 or below in its primary factor. Additionally, if an item is loaded to at least two factors simultaneously and with a factor loading difference between an item's major factor and another factor of less than 0.100, that item is also referred to be a problematic item (Field, 2013).

Before conducting EFA, it is important to ensure that the data is suitable for analysis. Researchers typically assess the suitability of the data by using Bartlett's test of sphericity, which is used to determine the significance of factor analysis ($p < 0.050$). The test determines whether the correlation matrix differs from an identity matrix, indicating whether the data is suitable for factor analysis. Additionally, the Kaiser-Meyer-Olkin (KMO) measure of sampling adequacy is used to check the data’s ability to make factors. The KMO index ranges from 0 to 1, with a recommended minimum value of 0.600 for successful factor analysis (Tabachnick & Fidell, 2013). After the items are loaded into factors, the nature of each factor is found, and then factors are named (Pallant, 2016).

3.3.2.2.3 Item Analysis

Item analysis is conducted to evaluate items. As a result of item analysis, the items that do not function well are examined for elimination and replacement. In Classical Test Theory (CTT), for item difficulty, high value means that item is easy ($p >.90$ - too easy) and the low value means that item is difficult ($p <.10$ - too difficult) (Crocker & Algina, 1986; Zumbo, 1999).

3.3.2.2.4 Item Response Theory (IRT)

Item response theory (IRT) is a statistical framework for measuring latent traits (Crocker & Algina, 1986). The latent trait of the current test was CT skills. IRT models assume that the probability of a participant's answering an item correctly depends on the level of the latent trait being measured and on the characteristics of the item, such as its difficulty and discrimination. In the current study, IRT was used for item analysis, and assumptions were evaluated. Then, item difficulty, item discrimination, and guessing parameters were estimated. Additionally, IRT models can be used to estimate the level of the latent trait for each person based on their responses to a set of items and to estimate the properties of the items themselves, such as their difficulty and their ability to discriminate among individuals with different levels of the trait. By using IRT (Hambleton & Jones, 1993), how the participants at different ability levels respond to an item was depicted by Item Characteristics Curves (ICC). ICC was the S-shaped curve that shows the probability of correct responses of participants on an item according to their ability level (Baker, 2001). When constructing IRT models, there were two assumptions. The first was the test data's dimensional structure. The assumption of unidimensionality means that the test items measure only one underlying latent trait, and that a participant's level on the trait is the only factor that determines their probability of responding correctly to any given item. Another one is local independence. Local independence is the assumption that an individual's response to one item should not influence their response to another item once their level on the latent trait is taken into account. Local dependence can lead to biased estimates of both the individual's trait level and the item parameters (Crocker & Algina, 1986). The other is the mathematical shape of the item characteristic function or ICC (Hambleton & Jones, 1993). By using IRT, test items could be evaluated with the help of ICC.

It is assumed that a group of items assesses a single common feature or skill with logistics models. (Hambleton & Jones, 1993). They evaluate the probability of participants' answering the items correctly. There are three logistics models which is differentiate in the amount of item parameters: One Parameter Logistics Model (1-PL), 2-PL, and 3-PL. These item parameters are used in the equation of cumulative logistic distribution function (Crocker & Algina, 1986). A three-parameter model from logistic models was used in the current study. This model was chosen because it was the most suitable model for multiple-choice tests (Crocker & Algina, 1986). It not only takes into account of difficulty and discrimination but also calculates the guessing factor as a parameter.

CHAPTER 4

FINDINGS

The test was developed based on the Computational Thinking Framework (Brennan & Resnick, 2012). For collecting information about the 30 multiple-choice items, interviews were conducted with three second-grade students and two primary school teachers and a computer science teacher. These data were used to revise the items. After the interviews, 24 items were decided to be included in the pilot study.

4.1 First Pilot test

The test was implemented to 291 2^{nd} grades students as pilot test. The second form of the test was developed according to item analysis. Item analysis was conducted to evaluate the individual test items' quality based on statistics (McCowan & McCowan, 1999).

4.1.1 Reliability of first pilot test

The minimum "acceptable" value of the Cronbach's Alpha is 0.70 (Cicchetti, 1994). The total alpha value of the pilot test was 0.67 (see Table 5). Although Cronbach's alpha value was close to .70, it was below the acceptable value.

Table 5. Reliability Statistics of Pilot Test

	Cronbach's Alpha	N of Items
Algorithmic Thinking	.53	9
Pattern Recognition	.25	8
Debugging	.47	7
Total	.67	24

The correlation between items and total scores as follows (see Table 6). Overall, many items were below the criteria of 0.30.

Table 6. Corrected Item-Total Correlations of 24 Item Test

	Corrected Item-Total Correlation
Item1	.22
Item2	.24
Item3	.35
Item4	.19
Item5	.30
Item6	.25
Item7	.27
Item8	.01
Item9	.29
Item10	.28
Item11	.35
Item12	.11
Item13	.23
Item14	.23
Item15	.34
Item16	.24
Item17	.15
Item18	.16
Item19	-.03
Item20	.22
Item21	.26
Item22	.20
Item23	.25
Item24	.34

4.1.2 Content validity of the pilot test

There were three experts to judge the items of the test. They rated the items as -1, 0, and 1 (see Table 7). The closer the item objective congruence was to 1.00, the more valid it was. It can be seen that item 17 was not valid for its objective (see Table 8). Also, item 9 was not valid according to experts.

Cognitive dimensions were close to each other by their content nature, and items might have the same content dimension even though they did not aim to measure different cognitive dimensions. Because the factors of the current test were not totally different, the item objective congruences were closer to zero more than one.

Table 7. Expert Ratings

	Algorithm			Pattern Recognition			Debugging		
Items	R1	R2	R3	R1	R2	R3	R 1	R 2	R 3
I1	1	1	1	0	1	0	-1	-1	0
I 2	1	0	1	1	1	1	-1	-1	-1
I 3	1	1	1	0	-1	-1	-1	0	-1
I 4	0	1	0	1	0	-1	1	1	0
I 5	1	1	1	1	0	1	-1	0	-1
I 6	1	1	1	0	-1	-1	-1	0	0
I 7	0	0	-1	1	1	1	0	-1	-1
I 8	0	0	1	1	1	1	0	-1	0
I 9	0	0	1	1	1	1	0	-1	-1
I 10	1	1	1	1	1	1	-1	-1	-1
I 11	1	1	0	-1	-1	-1	1	1	0
I 12	0	0	1	-1	-1	0	1	1	1
I 13	1	1	1	-1	0	0	-1	-1	-1
I 14	1	-1	1	-1	1	0	-1	-1	0
I 15	1	1	0	0	-1	-1	1	1	1
I 16	1	0	1	-1	-1	0	-1	-1	0
I 17	1	1	1	0	-1	0	-1	-1	-1
I 18	0	0	-1	1	1	1	-1	0	-1
I 19	-1	-1	0	1	1	1	-1	-1	-1
I 20	0	0	-1	-1	0	0	1	1	1
I 21	0	-1	0	0	-1	0	1	0	1
I 22	-1	0	0	0	0	1	1	1	1
I 23	0	1	1	0	-1	-1	1	1	1
I 24	1	0	1	1	1	1	-1	-1	0

4.1.3 Factor analysis of pilot test

The test had three factors: algorithmic thinking, pattern recognition, and debugging (see Table 10). Factor 1 was the Algorithmic Thinking. Items 1, 3, 6, 9, 11, 12, 13, 14, and 16 belonged to the Algorithmic Thinking. Items 5, 7, 8, 10, 17, 18, 19, and 24 belonged to pattern recognition, factor 2. And the factor 3 was debugging which consisted of items 4, 12, 15, 20, 21, 22, and 23.

Table 8. Item-Objective Congruence Indices

Items	Item-Objective Congruence Indices	Algorithms	Pattern Recognition	Debugging
I 1	0.24	1*	.33	-.66
I 2	0.33	.67*	1	-1
I 3	0.82	1*	-.66	-.66
I 4	0.24	.33	0	.66*
I 5	0.24	1	.66*	-.66
I 6	0.75	1*	-.66	-.33
I 7	0.75	-.33	1*	-.66
I 8	0.50	.33	1*	-.33
I 9	0.08	.33*	1	-.66
I 10	0.50	1	1*	-1
I 11	0.41	.66*	-1	.66
I 12	0.17	.33	-.66	1*
I 13	0.83	1*	-.33	-1
I 14	0.33	.33*	0	-.66
I 15	0.48	.66	-.66	1*
I 16	0.66	.66*	-.66	-.66
I 17	-0.16	1	-.33*	-1
I 18	0.75	-.33	1*	-.66
I 19	0.91	-.66	1*	-1
I 20	0.67	-.33	-.33	1*
I 21	0.49	-.33	-.33	.66*
I 22	0.48	-.33	.33	1*
I 23	0.48	.66	-.66	1*
I 24	0.48	.66	1*	-.66

In this model (see Table 9), both TLI (0.810) and CFI (0.828) were smaller than 0.95. In this model RMSEA value was 0.034. So, it was a good fitting model. Because of the TLI and CFI values, the model cannot be accepted as good.

Table 9. Fit Indices in CFA

Model	χ^2	df	χ^2/df	TLI	CFI	RMSEA (90% CI)
3 Factor model	237.96	167	1.42	0.81	0.83	0.034 (0.024; 0.044)

Table 10. Factor Loading Results

Factor	Indicator	Estimate
Factor 1	I1	0.090*
	I3	0.223*
	I6	0.213*
	I11	0.189*
	I12	0.173*
	I9	0.117*
	I13	0.190*
	I14	0.128*
	I16	0.123*
Factor 2	I5	0.132*
	I7	0.183*
	I8	-0.064
	I10	0.177*
	I17	0.032
	I18	0.163*
	I19	0.016
	I24	0.187*
Factor 3	I4	0.142*
	I12	0.144*
	I15	0.294*
	I20	0.078*
	I21	0.151*
	I22	0.140*
	I23	0.151*

4.1.4 Item analysis of the pilot test

Item analysis was conducted to evaluate items of the pilot test. As a result of item analysis, the items that did not function well were examined for elimination and replacement.

4.1.4.1 Classical test theory

For item difficulty, high value means that item is easy ($p > .90$ - too easy) and the low value means that item is difficult ($p < .10$ - too difficult). In the current test, items 4, 8, and 19 were more difficult than the others (see Table 11). For item discrimination in multiple-choice tests, $0.30 < d < 1.0$ is accepted as good. Items whose value is lower than .30 are problematic. Items 19, 8, 9, 4,13, 14, 17, 18, 20 and 22 did not discriminate the participants well. Items 13, 14, 17, 18, 20 and 22 were the conditional items. The problem on these items might have resulted from the dimension level. Item 19 was very poor and problematic (d=0.07). Although its difficulty level was high, it was not a discriminative item (see Figure 6). It was eliminated. Other conditional items were revised and included in the final study. Items 4 (d=0.29), 8 (d=0.12), and 9 (d=0.23) were also very poor and problematic items, and they were eliminated. Item 1 was easy and did not discriminate participants' responses. Because of this, it was also eliminated.

Table 11. Items' Difficulty and Discrimination Values

	Difficulty	Discrimination RIT
Item1	0.66	0.34
Item2	0.58	0.36
Item3	0.56	0.46
Item4	0.23	0.29
Item5	0.68	0.41
Item6	0.53	0.36
Item7	0.68	0.37
Item8	0.25	0.12
Item9	0.49	0.23
Item10	0.44	0.40
Item11	0.57	0.40
Item12	0.59	0.46
Item13	0.51	0.35
Item14	0.37	0.35
Item15	0.48	0.45
Item16	0.33	0.35
Item17	0.34	0.27
Item18	0.64	0.28
Item19	0.23	0.07
Item20	0.33	0.33
Item21	0.40	0.38
Item22	0.45	0.32
Item23	0.30	0.36
Item24	0.46	0.45

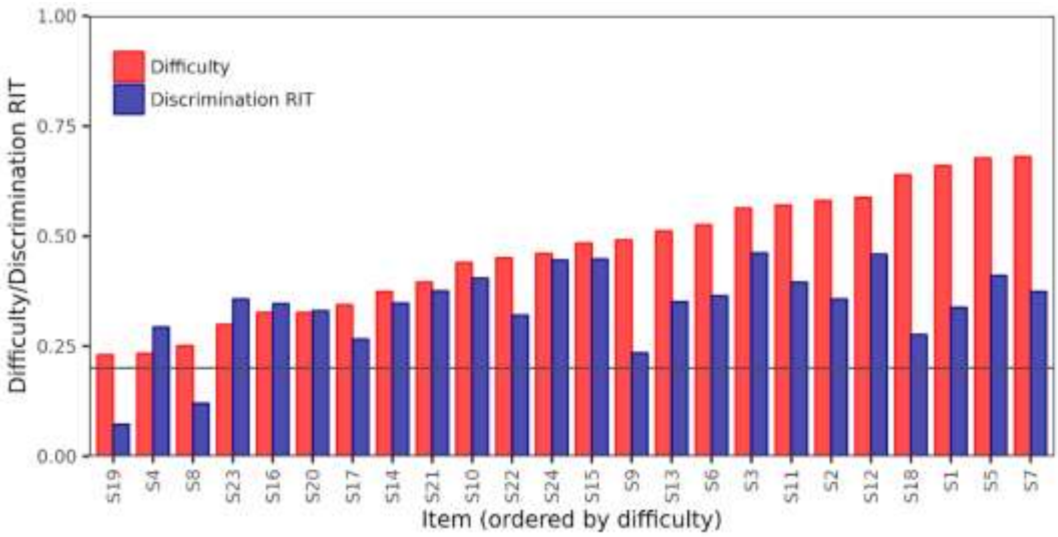


Fig. 6 Item difficulty and discrimination table

4.1.4.2 Item response theory

IRT was used for item analysis, and assumptions were evaluated. Then, item difficulty, item discrimination, and guessing parameters were estimated. Additionally, IRT (Crocker & Algina, 1986) was used to depict how participants of different ability levels responded to items, as represented by the ICC that is probability curve of correct participants' responses (Baker, 2001).

There was a single underlying factor that can explain the covariance among the items as it can be seen from Figure 7 (Edelen & Reeve, 2007). Test information curve was shown in Figure 9 and the standard error was shown in Figure 6. In table 12, ability estimations for scale were given.

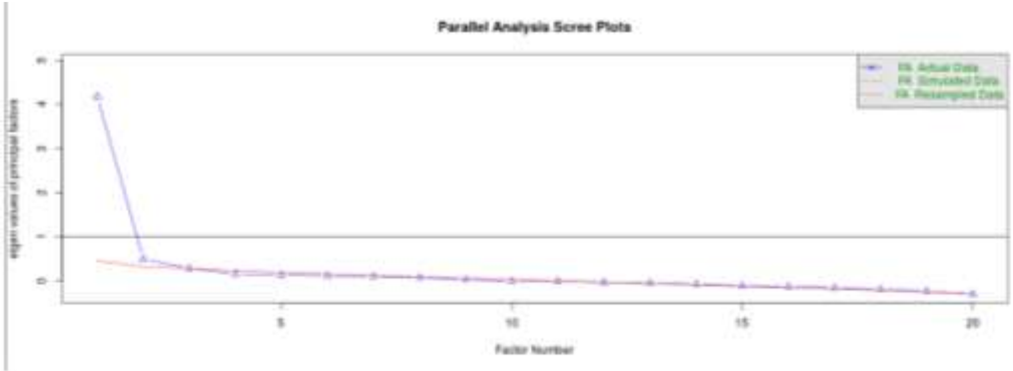


Fig. 7 Parallel analysis scree plots

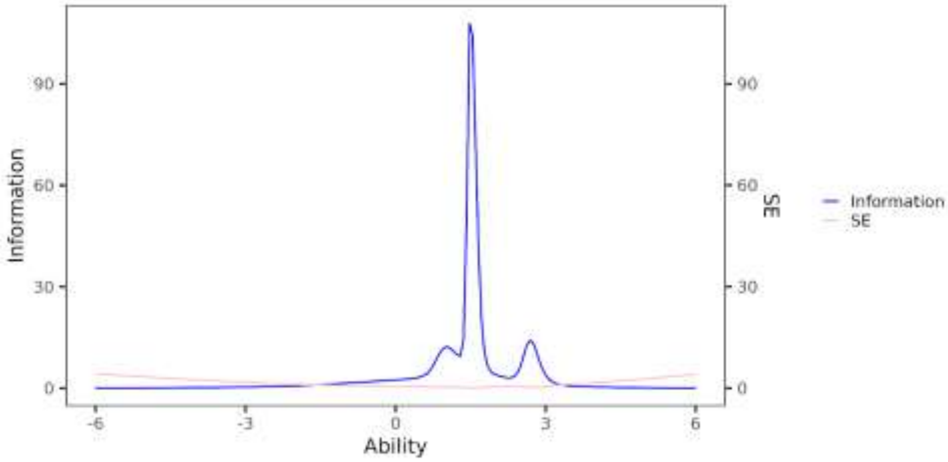


Fig. 8 Test information curve with standard error

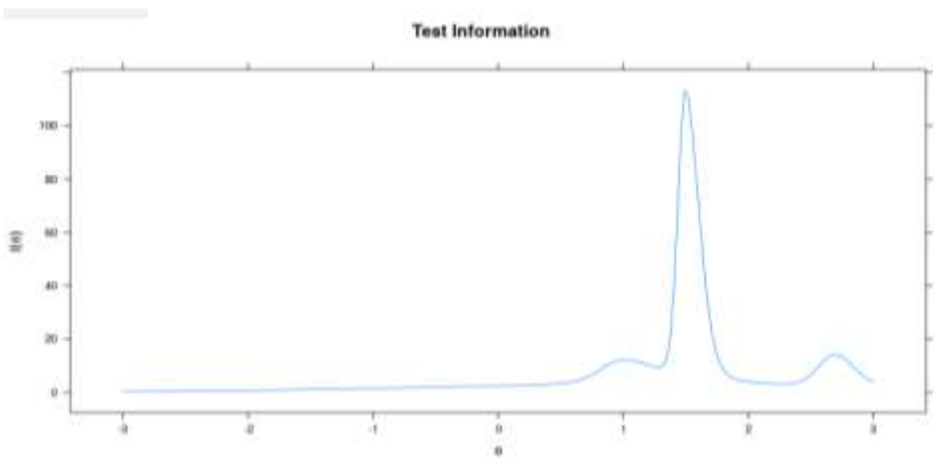


Fig. 9 Test information curve

Table 12. Ability Estimations for the Scale

Number of individuals	Mean Theta	SD of Thetas	Marginal Reliability	Mean of Theta Standard Errors	Max Theta	Min Theta
500	0	0.89	0.79	0.45	2.48	-2.09

4.1.4.3 A three-parameter model

The researchers utilized a three-parameter model derived from logistic models to ascertain the likelihood of participants' answering test items correctly. This particular model was deemed optimal for multiple-choice tests, as stipulated by Crocker and Algina (1986). The model takes into consideration the effect of guessing, which was included as a parameter.

The best model is the model that identifies the lowest value of the specified information criterion. In accordance with the Akaike information criterion (Akaike, 1974), the optimal model was found to be the three-parameter model, as demonstrated in Table 13. Minimum AIC value was three-parameter logistic models.

Table 13. Model Fit Indices

	AIC	BIC	logLik
1 PL	8886.351	8886.351	-4418.175
2 PL	8860.577	9036.896	-4382.288
3 PL	8856.790	9121.270	-4356.395

In Table 14, a is difficulty, b is discrimination, and c is guessing factor. Concerning the guessing factor (c value), every item had a reasonable c index. Item 1 was easy and did not discriminate participants' responses. Guessing factors of item 1 and item 18 were higher than maximum expected c value ($c_{expected} = 0.33 < c_{item1} = 0.54$ & $c_{item18} = 0.51$).

4.1.4.4 Item characteristics curves

As shown in the Figure 10, item 19 was problematic because it did not have proper S shape. While the ability of students is getting higher, the probability of the correct responses for this item getting lower.

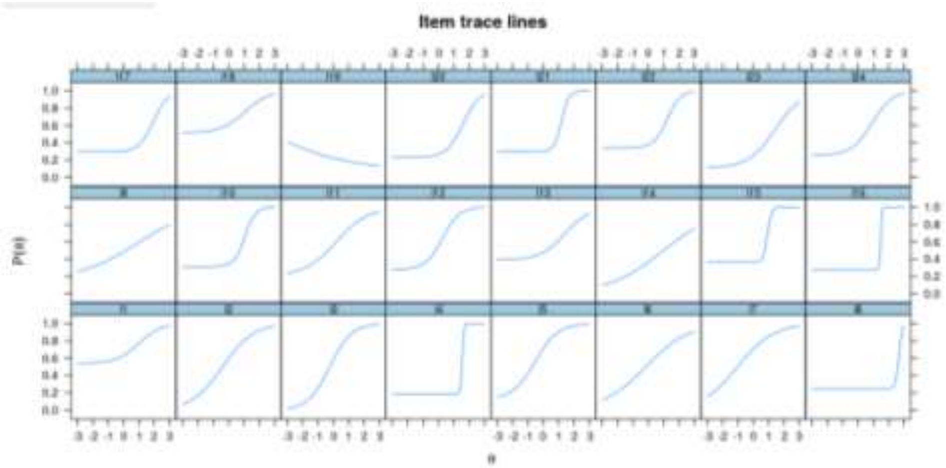


Fig. 10 Item characteristics curves

Table 14. Items' Parameter Estimations

	a	b	c
Item1	1.40	1.01	0.54
Item2	0.99	-0.39	0.00
Item3	1.40	-0.25	0.00
Item4	17.32	1.56	0.18
Item5	1.36	-0.53	0.12
Item6	0.72	-0.12	0.01
Item7	0.85	-1.02	0.00
Item8	8.94	2.66	0.24
Item9	0.53	0.89	0.16
Item10	2.89	1.06	0.31
Item11	0.96	0.20	0.20
Item12	1.82	0.25	0.27
Item13	1.26	1.43	0.39
Item14	0.55	1.00	0.00
Item15	7.19	0.93	0.36
Item16	23.37	1.47	0.27
Item17	2.15	1.95	0.29
Item18	1.27	1.08	0.51
Item19	-0.39	-4.56	0.09
Item20	1.90	1.60	0.23
Item21	4.14	1.19	0.29
Item22	2.29	1.24	0.34
Item23	1.15	1.44	0.11
Item24	1.49	0.90	0.24

4.2 Second pilot study

Second pilot test consisted of 18 items. There were 377 participants from public primary schools.

4.2.1 Reliability of second pilot application

The minimum “acceptable” value of the Cronbach’s Alpha is 0.70 (Cicchetti, 1994). The total alpha value of the second pilot test was 0.62 (see Table 15). It was under the acceptance value.

This study had a spelling mistake in item 9 (see Appendix C). It was stated in the item that the astronaut would earn 10 points instead of gaining 1 point when touching the star. This situation created difficulties for participants to understand because when they looked for 10 points gaining but choices had only 1 point earn blocks. This item was eliminated. Therefore, reliability analysis was completed again without item 9.

Table 15. Reliability Statistics of Second Pilot Test

	Cronbach's Alpha	N of Items
Algorithmic Thinking	.32	6
Pattern Recognition	.36	6
Debugging	.39	6
Total	.62	18

Table 16. Reliability Statistics of Second Pilot Test without Item9

	Cronbach's Alpha	N of Items
Algorithmic Thinking	.36	5
Pattern Recognition	.36	6
Debugging	.39	6
Total	.64	17

After it found that Cronbach’s alpha value was still low (0.64 < 0.70), the reason behind this was investigated. And the characteristics of the participants were considered. Along, it was found that the reliability score of the data from the participants who took the coding class (n = 125) was the highest ($\alpha_{CodingClassTaken}$= 0.68 > $\alpha_{DidNotTakeCodingClass}$ = 0.56). There were 251 participants who did not take any coding class.

Table 17. Reliability Statistics from the Participants who Took Coding Class

	Cronbach's Alpha	N of Items
Algorithmic Thinking	.48	5
Pattern Recognition	.38	6
Debugging	.46	6
Total	.68	17

Table 18. Reliability Statistics from the Participants who did not Take Coding Class

	Cronbach's Alpha	N of Items
Algorithmic Thinking	.27	5
Pattern Recognition	.33	6
Debugging	.32	6
Total	.56	17

According to the reliability analysis of the second pilot, items were revised, and the final test was only implemented to 2nd-grade students who took coding classes.

4.3 Final test

Final test was implemented with 210 students from 10 different public primary school in Istanbul (Asian side).

4.3.1 Item analysis of final test

In this section, items were evaluated according to difficulty and discrimination index.

4.3.1.1 Classical test theory

According to classical test theory, when assessing item difficulty, a high value indicates that the item is easy (p > .90 - excessively easy), while a low value suggests that the item is difficult (p < .10 - excessively difficult). According to item difficulty values presented in Table 19, there was no item excessively difficult or excessively easy. The test consisted of both hard and easy items (see Fig. 11). The most difficult item was item13 (p = 0.30) and the easiest item was Item 2 (p = 0.77).

With regards to item discrimination in multiple-choice tests, values between 0.30<d<1.0 are considered good, while values lower than 0.30 are problematic. Item 5 (RIT = 0.25) was considered problematic (see Table 19).

Table 19. Items' Difficulty and Discrimination Values

	Difficulty	Discrimination RIT
Item1	0.67	0.38
Item2	0.77	0.47
Item3	0.75	0.30
Item4	0.50	0.45
Item5	0.48	0.25
Item6	0.70	0.46
Item7	0.43	0.36
Item8	0.31	0.33
Item9	0.44	0.44
Item10	0.42	0.34
Item11	0.54	0.35
Item12	0.59	0.48
Item13	0.30	0.32

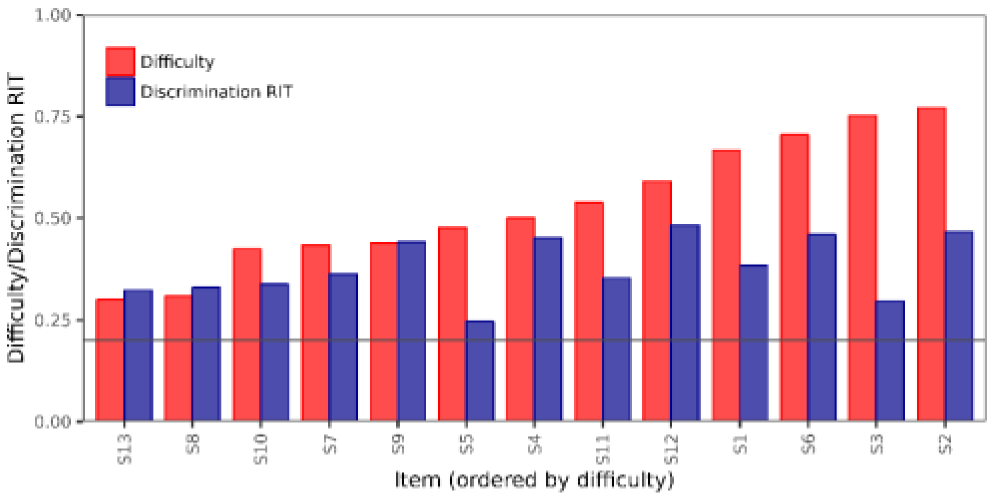


Fig. 11 Item difficulty and discrimination table of final test

4.3.1.2 Item response theory

IRT was used to estimate guessing factor (c) and to show how participants of various levels of ability answered items. The ICC was used to represent how participants of different ability levels responded to items.

Table 20. Items' parameter estimations of final test

	a	b	c
Item1	3.09	-0.05	0.31
Item2	10.46	-0.75	0.00
Item3	1.78	0.28	0.57
Item4	7.85	0.71	0.34
Item5	-1.67	-2.48	0.45
Item6	1.78	0.28	0.49
Item7	0.23	2.22	0.09
Item8	7.23	1.62	0.27
Item9	6.02	1.51	0.39
Item10	3.74	1.83	0.39
Item11	0.42	-0.37	0.00
Item12	8.10	0.88	0.49
Item13	2.26	1.86	0.25

The test is unidimensional if only the eigenvalue is higher than 1.0 (Edelen & Reeve, 2007). As shown in Figure 12, only one eigenvalue was higher than 1. The test assessed CT skills. Although three dimensions were proposed, the main factor in the test is CT skills. Figure 12 indicates there was one general factor.

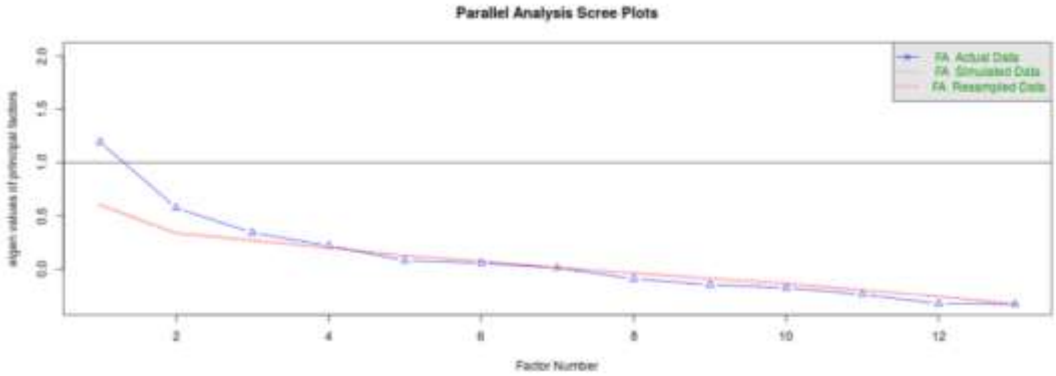


Fig. 12 Parallel analysis scree plots

According to ICC (see figure 13), the probability of the participants' correct responses on item 5 was decreasing while their ability level was increasing.

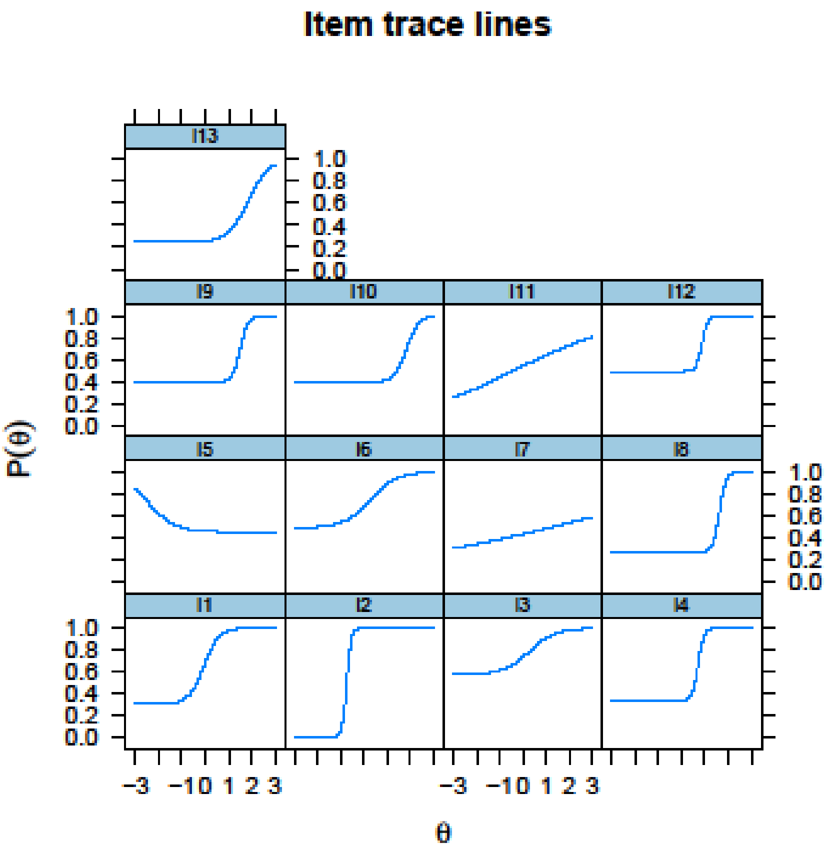


Fig. 13 Item characteristics curves

4.3.1.3 Comparison of models

Three-parameter logistic model was used. According to Crocker and Algina (1986), this specific model is regarded as the best one for multiple-choice exams. As presented in Table 21, 3 PL model had the minimum AIC and maximum BIC values. AIC and BIC values supported 3-PLM. Including guessing factor and using 3 PLM was the best for this test.

Table 21. Model Fit Indices

	AIC	BIC	logLik
1 PLM	3486.549	3533.409	-1729.275
2 PLM	3480.422	3567.447	-1714.211
3 PLM	3474.607	3605.144	-1698.303

4.3.2 Factor analysis of final test

4.3.2.1 Confirmatory factor analysis

In the current study, a 3-factor model was proposed, and its goodness-of-fit was evaluated using CFA. WLS method was used. All items were coded either 0 (indicating an incorrect response) or 1 (indicating a correct response).

The proposed model had three factors: algorithmic thinking, pattern recognition, and debugging (see Table 23). Factor 1 was the Algorithmic Thinking. Items 1, 2, 4, and 11 belonged to Algorithmic Thinking. Items 3, 5, 8, and 12 belonged to pattern recognition, factor 2. And factor 3 was debugging, consisting of items 6, 7, 9, 10, and 13.

According to CFA results presented in Table 22, the RMSEA value of 0.043 suggests that the model was a good fit (RMSEA = .043 < .060). However, this model's TLI and CFI values were less than 0.95 (CFI = .817< .950; TLI = .769 < .950), indicating that the model did not fit well. Despite the good-fitted RMSEA value, the model may not be considered acceptable because of the low TLI and CFI values.

Table 22. Fit Indices in CFA

Model	χ^2	df	χ^2/df	TLI	CFI	RMSEA (90% CI)
3 Factor model	206.874	78	2.652	0.769	0.817	0.043 (0.016 - 0.063)

Additionally, factor loadings were reported in Table 23. It is expected that the estimated value should be at least 0.40. In the first factor, item 11 (0.26) was lower than expected. All items related to the pattern recognition were lower than expected value. And in the last factor, debugging, 3 items were lower than the expected value (I7 = .33 & I10 = .27 & I13 = .26 < .40).

Table 23. Factor Loading Results CFA

Factor	Indicator	Estimate
Algorithmic Thinking	I1	0.64
	I2	0.86
	I4	0.44
	I11	0.26
Pattern Recognition	I3	0.26
	I5	0.01
	I8	0.13
	I12	0.34
Debugging	I6	0.58
	I7	0.33
	I9	0.40
	I10	0.27
	I13	0.26

As a result of CFA, from both global fit and item fit values showed that the proposed 3-factor structure did not fit to the students' responses. To further comprehend the test's structure, exploratory factor analysis (EFA) was used.

4.3.2.2 Exploratory factor analysis

Exploratory factor analysis (EFA) was performed to explore the relationships between items. As factors were associated with one another, direct oblimin was used to principal axis factor extraction.

Bartlett's test of sphericity (Bartlett, 1954) and the KMO measure of sampling adequacy were used to check whether the sample size of the data was enough for constructs individual factors. Factor analysis was significant due to Bartlett's test of sphericity ($p = 0.000 < 0.050$). The KMO index for the data used in the study was 0.607 (> 0.600), indicating that the data were suitable for EFA. Thus, the researchers concluded that the data were appropriate for conducting exploratory factor analysis.

Four factors were identified when EFA was initially conducted (see Table 24). But factors did not produce any meaningful groups. Even if the items in Factor 2 and Factor 3 were constructed as separate factors, they aimed to measure similar contents.

Table 24. Factor Loadings of Four Factor Structure

	Factor 1	Factor 2	Factor 3	Factor 4
I1	.645			
I2	.625			
I11	.261			
I6	.257			
I8		.416		
I4		.336		
I9		.329		
I7			.581	
I10			.362	
I12			.265	
I13		.367		
I5				-.549
I3				.476

The 4-factor structure did not create significant factors with item groups and as it was explained in CFA section, the proposed 3-factor structure tested in CFA did not work well in the current study. Therefore, items were forced to create 2 factors.

The test had a two-factor structure. As shown in Table 25, Items 1, 2, 3, and 6 created a factor, and items 7, 8, 9, 10, and 12 constructed another factor. These two factors varied from each other in terms of items' complexity. Factor 1 was sequencing and knowledge and comprehension levels of CT skills. Participants needed to find a way to send a main character in the question item to the target step by step. The items required that participants have knowledge about CT concepts; sequencing, loops, and conditionals, and they need to apply this knowledge for solving problems. The second factor, on the other hand, required higher cognitive abilities. The items in the second factor were two-step question items and intent to measure students thinking way. Students first needed to understand the given code and then find a new solution. In the second factor's items, participants were required to analyze the situation and synthesize their background knowledge. It is the CT practices in Brennan and Resnick's (2012) CT Framework.

Table 25. Factor Loadings of Two Factor Structure

	Factor 1	Factor 2
I1	.659	
I2	.597	
I3	.354	
I6	.282	
I4		
I11		
I9		.407
I12		.379
I7		.345
I10		.272
I8		.270
I5		
I13		

4.3.2.2 Confirmatory factor analysis for two factor structure

After EFA and deciding on a final form with two-factor structure, for confirming the two-factor structure with student data, CFA was conducted. The RMSEA value of 0.021 (RMSEA = .021 < .060). and TLI and CFI values (CFI = .977 > .950; TLI = .968 > .950), indicated that the model fit well and supported by student responses (see in Table 26).

Table 26. Fit Indices in CFA

Model	χ^2	df	χ^2/df	TLI	CFI	RMSEA (90% CI)
2 Factor model	135.98	36	3.77	0.968	0.977	0.021 (0.000; 0.060)

4.3.3 Reliability analysis of final test

For the first factor, items 1, 2, 3, and 6, reliability score was 0.51. And for the second factor, items 7, 8, 9, 10, and 12, reliability score was 0.39.

Table 27. Reliability Statistics

	Cronbach's Alpha	N of Items
CT Concepts	.51	4
CT Practices	.39	5
Total	.47	9

Table 28. Corrected Item-Total Correlations of CT Concepts Factor

	Corrected Item-Total Correlation	Cronbach's Alpha Value If Item Deleted
Item1	.35	.39
Item2	.40	.36
Item3	.24	.49
Item6	.23	.51

Table 29. Corrected Item-Total Correlations of CT Practices Factor

	Corrected Item-Total Correlation	Cronbach's Alpha Value If Item Deleted
Item7	.22	.33
Item8	.10	.41
Item9	.21	.33
Item10	.21	.33
Item12	.24	.31

CHAPTER 5

DISCUSSION AND CONCLUSION

The intended purpose of the current study was to develop a test for assessing CT skills of 2nd grade students and to examine its factor structure. According to Kalelioğlu et al. (2016), it is required to prioritize the teaching and measurement aspects of CT by designing appropriate assessment tools and tasks. And there is a need for assessment tool for CT in Turkish language (Eryılmaz & Deniz, 2019). To investigate the factor structure, the current study began with a deconstruction of CT. Numerous studies were examined because CT is the basis for many other skills (Doleck et al., 2017). As a result, the proposed structure of the test was based on the three cognitive dimensions of CT: algorithmic thinking (Brennan & Resnick, 2012; Kalelioğlu et al., 2016; Tsai et al., 2018; Wing, 2008), pattern recognition (Gülbahar et al., 2019; Kalelioğlu et al.,2016; Wing, 2010), and debugging (Tsai et al., 2018; Yadav et al., 2011). In the current study, the cognitive dimensions were measured using content dimensions of CT which were the sequencing, loops, and conditionals (Brennan & Resnick, 2012; Zapata Cáceres et al., 2021). After the specifications of the dimensions were determined, a 30-item item pool was created. Following the expert and student reviews, the first pilot test was conducted with 24 items. Items were decomposed into both cognitive and content dimensions of CT. Data were examined using reliability analysis (Crocker & Algina, 1986; Cicchetti, 1994) and validity analysis with CFA (Hu & Bentler, 1998; Schumacker & Lomax, 2010), with item analysis (Crocker & Algina, 1986; Zumbo, 1999), and with IRT (Baker, 2001; Crocker & Algina, 1986; Hambleton & Jones, 1993) following the first implementation. Its reliability score was under the acceptable value. And factors did not show the 3-factor structure; there were more than three factors. The interpretation of this result was that dividing the 24 test items into three cognitive and three content dimensions was extremely complex for 2nd grade students. Although the test aimed to measure CT cognitive dimensions through content dimensions, the pilot study indicated that content dimensions were also extraneous factors, affecting the test results. As a result of these findings, a second pilot study with an 18-item test was conducted. Items were identified only with their cognitive dimensions, eliminating the content dimensions. Yet, the second pilot study failed to create a significant reliability index (α=0.623 < 0.700), and there was a faulty question. Because of this, items were reviewed and either removed or updated. The final test included 13 items. The final test was conducted through the same procedures as pilot studies. It took one lesson hour (40 minutes). After conducting CFA, it was found that the proposed three-factor structure did not fit the data from the final test implementation. Therefore, EFA was conducted to investigate the data structure (Pallant, 2016; Tabachnick & Fidell, 2013). And the data analysis using EFA showed that the test had a two-factor structure instead. The two factors were found to be associated with CT Framework's (Brennan & Resnick, 2012) two levels: CT concepts and CT practices.

The findings of the current study revealed a two-factor structure for the test, which aligns with Brennan and Resnick's CT framework (2012). The two factors identified were categorized under CT concepts and CT practices. The test did not include CT perspectives from the CT framework like BCTt (Zapata-Cáceres et al., 2021), because it is hard to observe perspectives of students with a single assessment tool in a limited time as Brennan and Resnick (2012) also stated. Most of the multiple-choice traditional tests were designed by using just two factors of CT

Framework: CT concepts and CT practices (Kong & Liu, 2020). Factor 1 focused on CT concepts and required participants to apply their knowledge of CT concepts such as sequencing, loops, and conditionals to guide a main character through a series of steps to reach a target (Zapata-Cáceres et al., 2020). CT concepts factor comprised items 1, 2, 3, and 6 (see Table 25). The other factor was CT practices in the CT framework which illustrated the building methods and design techniques that 2nd-grade students used to make their programs. Through CT practices, how students learned was observed rather than just what they learned (Brennan & Resnick, 2012). That is, the items in this factor were designed as two-step questions to assess students' thinking processes. Firstly, students needed to comprehend the given code, and then they had to develop innovative solutions. CT practices factor consisted of items 7, 8, 9, 10, and 12 (see Table 25).

The factor associated with CT concepts assessed students' conceptual knowledge of CT, focusing on the subdimensions of sequencing, loops, and conditionals (Brennan & Resnick, 2012; Roman-Gonzalez, 2015; Zapata Caceres et al., 2021). There are a few test development studies that use CT concepts to assess learners' CT. Items in CTt (Roman-Gonzalez et al., 2016) and BCTt (Zapata-Caceres et al., 2020) are examples of measuring CT concepts. In CTt (Roman-Gonzalez et al., 2016) item 6 is a loops item. It asks how many times the code given in the question should be repeated so that the character can reach the target. Students need to understand the given code and use the code over and over to find out how many runs required for the character to reach the goal. In BCTt (Zapata-Caceres et al., 2020), item number 3 is a sequencing item. In this item, students need to find how chicken goes to the mother chicken by not touching cat. They need to select the true algorithm to execute codes. Students are required to know CT concepts and apply the knowledge to solve the given code problem.

Within the scope of this study, sequencing was defined as forming program commands and putting them in an order to create a meaningful desirable outcome. Loops enabled students to do the same thing for multiple times defined in the code, and conditionals were used to execute a code based on a condition (Parker et al., 2021). The factor comprised items 1, 2, 3, and 6 in the final test implementation, which specifically aimed to measure these CT concepts. Item 1 designed to measure Algorithmic Thinking skills was a basic sequencing question. Students were asked to describe how a penguin can reach a seal by selecting the appropriate code. Similarly, item 2 was an algorithmic thinking item that required sequencing. It included a task similar to the one in item 1, prompting students to determine where a character will go when the given code blocks are executed. Both items required 2nd-grade students to use sequencing to solve the problem. Students needed to find a step-by-step algorithm by doing sequencing. Item 3 required basic knowledge of loops. Students were asked to identify the code group that would guide a robot to reach a computer. The options included different numbers of repetitions of the "move forward" block. The first choice was "4 times repeat" the "move forward" block, the second was "5 times repeat" the "move forward" block, and the third was "6 times repeat" the "move forward" block. This item required students to do repetition by loops. Lastly, item 6 was initially designed to measure debugging skills of students, but it only required basic sequencing knowledge. In the item, it was asked that "In which choice should the code shown be changed to the "turn down" code so that the chick can go to the rooster?". Students were asked to identify the correct code in a given set that needed to be changed to "turn down" so that a chick can reach a rooster. This item required students to determine the correct sequence of actions for the chick. Since the code for this particular item is easy to follow, students can solve it as a regular sequencing item. Therefore, for this item, there is no need for students to go through the expected debugging steps which are testing the given code, identifying any present errors, and then fixing those errors. This is because the code is straightforward and does not require extensive troubleshooting. They could just find the right algorithm by comparing the choices and without a testing process. So, this item was also a sequencing item. As a result, the factor of CT concepts focused on measuring students' understanding and application of sequencing, loops, and conditionals. As explained above, the specific items in this factor provided tasks related to these CT concepts.

On the other hand, the CT practices factor required not only the application of knowledge but also the thinking process (Brennan & Resnick, 2012). Students were required to use the subskills of CT, such as Algorithmic Thinking, Pattern Recognition, and Debugging (Brennan & Resnick, 2012; Relkin et al., 2020; Zeng, Yang, & Bautista, 2023). According to Zeng et al. (2023), debugging that includes testing is a typical CT practice subdimension, and algorithmic thinking and pattern recognition are newly arising subdimensions of CT practices. These subskills of CT can be measured by using specific CT concepts. According to Guggemos, Seufert, and Roman-Gonzalez (2019), in the CT practices factor, algorithmic thinking is a key subskill and can be observed in almost all items because all programming tasks demand a step-by-step sequencing to find a resolution. Loops can be used for repetitions of a code sequence. The term "decomposition" refers to the process of breaking down the problem into smaller, simpler components. Students find errors in the offered code block sequences through debugging (Guggemos et al., 2019). This study shows item examples to measure CT practices.

Items 7, 8, 9, 10, and 12 were the items of CT practices factor. Item 7 was constructed to measure debugging skills through conditional loops. Students firstly needed to know the concepts of loops and conditionals and then to use them in the problem-solving process with a debugging practice as a way of thinking process (Brennan & Resnick, 2012; Zeng et al., 2023). Also in this item, algorithmic thinking was supposed to be used because every algorithm creation process includes algorithmic thinking (Guggemos et al., 2019). "Which of the code options should be changed in order for the mother dinosaur to go to the baby dinosaur?" was asked. And participants were expected to understand given code, find the error in the code, compare algorithms of the codes in options, and select the right code group to solve the problem. Item 8 was a pattern recognition item, and it included both conditionals and loops from CT concepts. "With which code group in the choices can the rabbit go to the square where her house is by just picking the carrots without getting the apples?" was the question of item 8. To solve this item, students needed to understand the given conditional code algorithm and to select how many times the code should be run to reach the target. Students need to think algorithmically, understand the given code, see the pattern in the question, and select the right algorithm. Item 9 focused on debugging skills with conditionals, requiring students to identify the code that needed to be changed to achieve the desired outcome. The question for the item was "Which code shown in red in the choices should be changed in order for the game to be played this way?". Students should explain the codes and select the wrong code that should change. By doing so, they used their algorithmic thinking skills and debugging skills. Item 10 was developed to assess debugging skills with conditional loops. "Which of the choice should the code change with so that the rabbit can go to the tree?" was the question for item 10. Similar to the other debugging items in this factor, students were supposed to compare the choices with the given code and choose the appropriate code group to solve the challenge in this item. They should test the given code, find the error, and find the code that fixed the error. And the last item, item 12, was an algorithmic thinking item that was developed by using loops. The item question was "With which code group in the choices can the gray cat pass through the green and blue squares and go to the yellow cat?". Basically, students needed to find a true algorithm to solve that question. While doing so, they should make relations between the loops and solve the problem. Items in the factor of CT practices in this study aligned with the thinking skills emphasized in the CT framework (Brennan & Resnick, 2012) and the sub-dimensions identified by Guggemos et al. (2019). The integration of algorithmic thinking, pattern recognition, and debugging within the items reflected the complexity of CT practices.

In the beginning, the current study used CT concepts to measure CT practices. Brennan and Resnick (2012) divided CT practices into three “practices”. First one is being incremental and iterative. These practices compensated with algorithmic thinking in this test. Second practices are reusing and remixing that are the pattern recognition (Zeng et al., 2023). The third practices of the framework is testing and debugging which are handled in debugging in this test. While writing the test items, the items were created to measure CT practices. Yet, they contained CT concepts in addition to CT practices. This may have affected the final result. Because the test consisted of easy and complex items (see the Table 19), some items remained too easy to measure CT practices, and those items did not measure CT applications accurately. And they only measured the CT concepts. This can explain the reason why the items are loaded as two factors: CT concepts and CT practices. For example, in CTT-ES (Tsai et al., 2022), they aimed to measure CT practices; decomposition, algorithm, abstraction, generalization, and evaluation. To achieve this aim, they used a non-programming context. As a contradictory example, in BCTt, Zapata-Cáceres et al. (2021) measured CT concepts by using these contexts: sequencing, loops, and conditionals. There are several CT assessment tools that handle CT from different approaches; however, the social and cognitive development progresses of children must be investigated in order to evaluate their CT skills at an early age (Relkin et al., 2020). In terms of the assessing CT skills of 2nd-grade students, this study shows that it is better to focus on this two-factor structure and differentiate items based on CT Framework. By considering the distinctions between CT concepts and practices, a more effective evaluation of students' CT skills can be achieved, aligning with the framework proposed by Brennan and Resnick (2012).

Another possible reason behind the proposed structure did not fit the data may be that participants of the study did not receive any explicit instruction on CT at their schools. Since computer science classes begin in the fifth and sixth grades (MoNE, 2018) and coding classes are not very common in preschools and primary schools (Eryılmaz & Deniz, 2019), the 2nd grade students only attended after-school programs to practice coding. Moreover, participants that took the final test mostly started to take coding classes four months before the study was conducted. This possible lack of formal instruction and background knowledge may have influenced the results and affected participants’ comprehension of the test items. Kalelioğlu and Gülbahar (2015) also stated that different levels of students' background knowledge or absence of background knowledge can cause low student success and a lack of understanding of the diverse ways that the questions are phrased.

Additionally, the use of Scratch visuals and block-code-based structures in creating the test items may have posed challenges for 2nd-grade students. Although Scratch is the most popular coding environment at the K-12 level (Eryılmaz & Deniz, 2019; Zhang & Nouri, 2019), the participants in this study primarily learned coding through unplugged robotics classes rather than using Scratch as a coding platform. This unfamiliarity with Scratch's environment, block codes, and characters might have added extraneous cognitive load for 2nd-grade students. Students who have no prior knowledge of programming in a computer needs unplugged assessment (Relkin et al., 2020). As a result, the visuals used in the test, which were Scratch screenshots, could have also distracted their attention. This situation could also be understood from the second pilot test implementation. The second pilot test implementation further supported this observation, indicating that participants who had prior Scratch-based coding classes achieved more reliable results. In that pilot study, the sample that took a Scratch-based coding class had a more meaningful reliability index than the one that did not take one. Because the participants did not know the environment in the final test implementation, they tried to understand the structure of the items first, and after that, they tried to understand what was asked. This might have caused cognitive load for second graders. For example, Scratch's original starting code "when the green flag clicked" got noticed on almost every item. For the ones who did not take Scratch, it can cause the extraneous load. In addition, not being familiar with Scratch environment may have caused the test item visuals to distract their attention. Considering these factors, it may be beneficial to assess CT skills using programming contexts aligned with the CT Framework while considering the students' familiarity with the coding environment. Cruz Castro, Shoaib, and Douglas (2021) suggested utilizing Brennan and Resnick's CT Framework (2012) in the context of programming using Scratch. However, the findings indicate that the construct may be more intricate than initially thought. Consequently, the results suggest that it would be beneficial to explore alternative approaches for measuring it.

One of the limitations of this study is its sample size. Due to time and resource constraints, the data were collected only from 210 participants for the final test application. While this sample size was sufficient to detect statistically significant effects (Pallant, 2016), it may not be representative of the larger population. Additionally, the study was conducted in a single geographic region, which may limit the generalizability of the findings to different areas or cultures. The test was only implemented with public school students. Also, the public schools where data were collected were located only in two regions in the Asian side of Istanbul.

This study only focused on the assessment of CT skills using a paper-and-pencil test, which may not capture the full range of CT skills in young children. Scratch, a web-based programming environment, allows students to make projects on computer by using code blocks (Maloney et al., 2010). Because of this, coding on Scratch is meaningful on computers. But when it is transformed to a paper-and-pencil environment, some blocks may lose their meaning, such as “click on button”. To reduce this, some items, such as Item 16, were eliminated in the pilot study (see Appendix A). Therefore, it could be more meaningful to do the test on a computer, and such tests could produce significant results. This limitation implies that the study results should be interpreted carefully, and further research is needed to confirm or refute the findings with a more representative sample. The results can change if the test is applied in private schools or with different age groups. Students in private schools usually have coding classes for many years, mostly from kindergarten. It is assumed that if students are familiar with the Scratch environment, there is a greater probability that test items are more meaningful to them. To build on the findings of this study, future research could explore alternative structures for measuring the construct or consider alternative measurement approaches altogether. Additionally, it would be valuable to investigate whether the lack of fit between the proposed structure and the data is due to the sample characteristics, or if it is a more fundamental issue with the construct itself. Further research could also investigate potential moderating variables that may explain why the proposed structure did not fit the data.

In the realm of CT, to further advance our understanding of its implementation and assessment, it is suggested to explore various dimensions of CT through interactive coding tasks within a computer-based environment, especially when working with 2nd-grade students (Brennan & Resnick, 2012; Guggemos et al., 2019). Incorporating interactive coding tasks can offer valuable opportunities to actively engage young learners and provide them with hands-on experiences in understanding how codes function. This approach aligns with the constructionist approach to learning, which emphasizes the importance of young people's participation in the creation process (Kafai & Resnick, 1996). By actively participating in interactive design activities, students can not only enhance their CT skills but also develop essential competencies such as algorithmic thinking, pattern recognition, and debugging (Brennan & Resnick, 2012). The interactive design activities foster a deeper understanding of CT concepts by allowing students to move beyond mere imagination and enabling them to directly interact with and manipulate code. This hands-on approach to learning serves as a powerful tool for nurturing students' CT abilities and promoting their overall engagement and creativity.

It is worth noting that students in private schools in Turkey are more often exposed to CT concepts through specialized curricula and resources, which may result in a higher familiarity with CT compared to students from other educational settings. Therefore, if the CT test developed in this study is administered in private schools, the results may vary due to the students' pre-existing CT knowledge and skills. It is suggested to implement this test in private schools and investigate the factor analysis, reliability, and validity scores.

It is important to recognize that the dimensions of the CT test implemented in this study focused specifically on algorithmic thinking, pattern recognition, and debugging. However, the final version of the test expanded to include two dimensions: CT concepts and CT practices. This refinement allows for a more comprehensive evaluation of students' CT proficiency, capturing both their theoretical knowledge and their ability to apply it in practical scenarios. This test can be implemented together with reliable and valid tests by researchers and teachers who aim to identify CT skills of students.

In conclusion, there has been a growing interest in the development of CT in education systems worldwide (Brennan & Resnick, 2012; Guzdial, 2010; Wing, 2016). Despite the efforts to implement CT in classrooms, there is a limited number of studies focusing on the assessment of CT (Demir & Seferoğlu, 2019; Eryılmaz & Deniz, 2019). Therefore, the need for a valid and reliable CT measurement tool is increasingly evident (Demir & Seferoğlu, 2019; Relkin, Ruiter, & Bers, 2020).

Assessing the CT skills of second-grade students is essential because it provides insight into their developing cognitive abilities and problem-solving skills, which are rapidly evolving at this age (Babazadeh & Negrini, 2022; Barr & Stephenson, 2011). Moreover, early assessment of CT skills can help identify future STEM leaders and shape the development of CT curricula, ensuring that it addresses the needs of students and providing them with the necessary skills to succeed in the 21st century (Grover & Pea, 2013; Tang et al., 2020). Therefore, the results of the current study will have practical implications for teachers, curriculum developers, and policymakers, contributing to the overall goal of enhancing the quality of computer science education in primary schools in Turkey. In this study, the test was developed for assessing CT skills of 2nd grade primary school students and investigated its factor structure using EFA in this study. The results showed that the test had a two-factor structure associated with the cognitive domain levels of knowing and applying. Although the proposed three-factor structure did not fit the data, the two-factor structure provides valuable insights into the nature of CT skills in young children and highlights the importance of differentiating between cognitive domain levels in CT assessments. However, future research is required to validate the two-factor structure of the test in larger and more diverse samples and to explore the potential of the test for assessing CT skills.

APPENDIX A

PERMISSIONS

T.C.
İSTANBUL VALİLİĞİ
İl Milli Eğitim Müdürlüğü

Sayı : E-59090411-20-66140977 19/12/2022
Konu : Anket ve Araştırma İzni (Sena ÖZTÜRK SÖYLEMEZ)

VALİLİK MAKAMINA

İlgi : a) Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarihli ve 2020/2 sayılı genelgesi.
b) Valilik Makamının 28.01.2022 tarihli ve 42238060 sayılı oluru.
c) Sena ÖZTÜRK SÖYLEMEZ'in 15.12.2022 tarihli ve 65976462 numarada kayıtlı dilekçesi.

Araştırma Konusu	: İlkokul 2. Sınıf Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerisi Testi Geliştirme Çalışması
Araştırma Türü	: Anket
Araştırma Yeri	: Beykoz, Beşiktaş, Kadıköy, Sarıyer, Şişli, Ümraniye, Üsküdar
Araştırma Kişiler	: İlkokul Öğrencileri
Araştırmanın Süresi	: 2022 - 2023 Eğitim - Öğretim Yılı

Yukarıda bilgileri verilen araştırmanın; 6698 sayılı Kişisel Verilerin Korunması Kanununa aykırı olarak kişisel veri istenmemesi, öğrenci velilerinden açık rıza onayı alınması, yüz yüze eğitime geçmiş olan kurumlarımızda, Covid-19 tedbirlerinin araştırmacı ve ilgili kurum idarelerince alınması, bilimsel amaç dışında kullanılmaması, bir örneği Müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanması, katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun kamuoyuyla paylaşılmaması ve araştırma bittikten sonra 2 (iki) hafta içerisinde Müdürlüğümüze gönderilmesi, okul idarelerinin denetim, gözetim ve sorumluluğunda, eğitim ve öğretimi aksatmayacak şekilde, ilgi (a) genelge esasları dâhilinde uygulanması kaydıyla Müdürlüğümüzce uygun görülmektedir.

Makamınızca da uygun görüldüğü takdirde olurlarınıza arz ederim.

Levent YAZICI
İl Milli Eğitim Müdürü

OLUR
Dr. Hasan Hüseyin CAN
Vali a.
Vali Yardımcısı

Ek:
1- İlgi (b) Olur ve Ekleri (27 Sayfa)
2- İlgi (c) Dilekçe (1 Sayfa)

Bu belge güvenli elektronik imza ile imzalanmıştır.

Adres : Binbirdirek Mah. İmran Öktem Cad. No: 1 Sultanahmet Fatih İstanbul
Telefon : 0212 384 36 30
E-posta : stratejigelistirme34@meb.gov.tr
Kep Adresi : meb@hs01.kep.tr
Belge Doğrulama : https://www.turkiye.gov.tr/meb-ebys
Bilgi için : Aykut ÇELİK
Unvanı : Büro Hizmetleri
İnternet Adresi : http://istanbul.meb.gov.tr/

5aa1-1c8a-3ea2-839b-68bb

T.C.
İSTANBUL VALİLİĞİ
İl Milli Eğitim Müdürlüğü

Sayı : E-59090411-20-42238060 28/01/2022
Konu : Anket ve Araştırma İzni (Sena ÖZTÜRK)

VALİLİK MAKAMINA

İlgi : a) Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.01.2020 tarihli ve 2020/2 sayılı genelgesi.
b) Boğaziçi Üniversitesinin 11.01.2022 tarihli ve 46916 sayılı yazısı.
c) Müdürlüğümüz Araştırma ve Anket Komisyonunun 24.01.2022 tarihli tutanağı.

Araştırma Konusu	: İlkokul 2. Sınıf Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerisi Testi Geliştirme Çalışması
Araştırma Türü	: Anket
Araştırma Yeri	: İstanbul
Araştırma Yapılacak Kişiler	: İlkokul Öğrencileri
Araştırmanın Süresi	: 2021 - 2022 Eğitim ve Öğretim Yılı

Yukarıda bilgileri verilen araştırmanın; 6698 sayılı Kişisel Verilerin Korunması Kanununa aykırı olarak kişisel veri istenmemesi, öğrenci velilerinden açık rıza onayı alınması, yüz yüze eğitime geçmiş olan kurumlarımızda, Covid-19 tedbirlerinin araştırmacı ve ilgili kurum idarelerince alınması, bilimsel amaç dışında kullanılmaması, bir örneği Müdürlüğümüzde muhafaza edilen mühürlü ve imzalı veri toplama araçlarının kurumlarımıza araştırmacı tarafından ulaştırılarak uygulanması, katılımcıların gönüllülük esasına göre seçilmesi, araştırma sonuç raporunun kamuoyuyla paylaşılmaması ve araştırma bittikten sonra 2 (iki) hafta içerisinde Müdürlüğümüze gönderilmesi, okul idarelerinin denetim, gözetim ve sorumluluğunda, eğitim ve öğretimi aksatmayacak şekilde, ilgi (a) genelge esasları dâhilinde uygulanması kaydıyla Müdürlüğümüzce uygun görülmektedir.

Makamınızca da uygun görüldüğü takdirde olurlarınıza arz ederim.

Levent YAZICI
İl Milli Eğitim Müdürü

OLUR
28/01/2022
Dr. Hasan Hüseyin CAN
Vali a.
Vali Yardımcısı

Ek:
1- İlgi (b) Yazı ve Ekleri (27 Sayfa)
2- İlgi (c) Tutanak (1 Sayfa)

Adres : Binbirdirek Mah. İmran Öktem Cad. No: 1 Sultanahmet Fatih İstanbul
Telefon : 0212 384 36 30
E-posta : stratejigelistirme34@meb.gov.tr
Kep Adresi : meb@hs01.kep.tr
Belge Doğrulama : https://www.turkiye.gov.tr/meb-ebys
Bilgi İçin : Aykut ÇELİK
Unvanı : Büro Hizmetleri
İnternet Adresi : http://istanbul.meb.gov.tr/

Boğaziçi Üniversitesi Evrak Tarih ve Sayısı: 31.01.2022-50360

T.C.
İSTANBUL VALİLİĞİ
İl Milli Eğitim Müdürlüğü

GÜNLÜDÜR

Sayı : E-59090411-44-42282094 28.01.2022
Konu : Anket ve Araştırma İzni (Sena ÖZTÜRK)

BOĞAZİÇİ ÜNİVERSİTESİ REKTÖRLÜĞÜNE
(Sosyal Bilimler Enstitüsü Müdürlüğü)

İlgi : a) Yenilik ve Eğitim Teknolojileri Genel Müdürlüğünün 21.02.2020 tarihli ve 2020/2 sayılı genelgesi.
b) Valilik Makamının 28.01.2022 tarihli ve 42238060 sayılı oluru.

Valilik Makamının Anket ve Araştırma İzni konulu ilgi (b) oluru ve kullanılması uygun görülen ölçme araçlarının Müdürlüğümüzce mühürlenen örnekleri ekte gönderilmiştir.

İlgi (a) genelgenin 28. maddesinde; "Araştırma uygulama izni alan kamu kurum ve kuruluşları, uluslararası kuruluşlar, üniversiteler, sivil toplum kuruluşları ve araştırmacılar tamamladıkları bilimsel araştırma ile ilgili sonuç raporlarını, izni aldıkları ilgili birime çalışma bitiminden itibaren 30 gün içerisinde göndereceklerdir." ifadesi yer almaktadır.

Olur gereğince işlem yapılması ve araştırma sonuç raporunun ekte sunulan örneğe göre Müdürlüğümüz Strateji Geliştirme Şubesine gönderilmesi hususlarında gereğini arz ederim.

Abdurrahman ENSARİ
İl Milli Eğitim Müdürü a.
Şube Müdürü

Ek:
1- Valilik Oluru (1 Sayfa)
2- Rapor Örneği
3- Ölçekler

Evrak Tarih ve Sayısı: 19.12.2021-43510

T.C.
BOĞAZİÇİ ÜNİVERSİTESİ
SOSYAL VE BEŞERİ BİLİMLER YÜKSEK LİSANS VE DOKTORA TEZLERİ ETİK İNCELEME KOMİSYONU
TOPLANTI KARAR TUTANAĞI

Toplantı Sayısı	:	25
Toplantı Tarihi	:	15.12.2021
Toplantı Saati	:	15:00
Toplantı Yeri	:	Zoom Sanal Toplantı
Bulunanlar	:	Prof. Dr. Ebru Kaya, Prof. Dr. Fatma Nevra Seggie, Dr. Öğr. Üyesi Yasemin Sohtorik İlkmen
Bulunmayanlar	:	

Sena Öztürk
Bilgisayar ve Öğretim Teknolojileri Eğitimi Bölümü

Sayın Araştırmacı,
"Development of an computational thinking skills test for 2nd grade primary school students" başlıklı projeniz ile ilgili olarak yaptığınız SBB-EAK 2021/85 sayılı başvuru komisyonumuz tarafından 15 Aralık 2021 tarihli toplantıda incelenmiş ve uygun bulunmuştur.

Bu karar tüm üyelerin toplantıya çevrimiçi olarak katılımı ve oybirliği ile alınmıştır. COVID-19 önlemleri kapsamında kurul üyelerinden ıslak imza alınamadığı için bu onay mektubu üye ve raportör olarak Fatma Nevra Seggie tarafından bütün üyeler adına e-imzalanmıştır.

Saygılarımızla, bilgilerinizi rica ederiz.

Prof. Dr. Fatma Nevra SEGGIE
ÜYE

e-imzalıdır
Prof. Dr.Fatma Nevra SEGGIE
Raportör

SOBETİK 25 15.12.2021

Evrak Tarih ve Sayısı: 11.01.2022-46916

T.C.
BOĞAZİÇİ ÜNİVERSİTESİ
Sosyal Bilimler Enstitüsü Müdürlüğü

Sayı : E-60914867-300-46916 11.01.2022
Konu : Araştırma İzni (Sena ÖZTÜRK) Hk.

İSTANBUL VALİLİĞİNE
(İstanbul İl Milli Eğitim Müdürlüğü)

Boğaziçi Üniversitesi Sosyal Bilimler Enstitüsü'ne bağlı Bilgisayar ve Öğretim Teknolojisi Anabilim Dalı, Eğitim Teknolojisi Yüksek Lisans Programı öğretim elemanı Dr. Öğr. Üyesi Duygu UMUTLU'nun danışmanlığında, Eğitim Teknolojisi Yüksek Lisans Programı Öğrencisi (Öğrenci no: 2020770015) Sena ÖZTÜRK tarafından yürütülen "İlkokul İkinci Sınıf Öğrencilerine Yönelik Bilgi İşlemsel Düşünme Becerisi Testi Geliştirme Çalışması" başlıklı çalışmanın Etik Kurul onayı alınmış olup, ilgili çalışma kapsamında İstanbul İl Milli Eğitim Müdürlüğü'ne bağlı devlet ilköğretim okullarında 2021-2022 akademik yılı 2. yarıyılında araştırma verileri toplanmak istenmektedir. Üniversitemiz öğrencisi Sena ÖZTÜRK'e EK'te detayları ve veri toplama araçları iletilen çalışmasını tamamlayabilmesi için gerekli iznin verilmesi hususunda araştırma izin talebini bilgilerinize ve onayınıza sunarım.

Saygılarımla,

Doç. Dr. İlhami ÖZTÜRK
Enstitü Müdürü

Ek:
1- Sena Öztürk - MEB (48 sayfa)
2- Bölümden Gelen Talep Yazısı Sena Öztürk Veri Toplama HK. (1 sayfa)

Bu belge, güvenli elektronik imza ile imzalanmıştır.

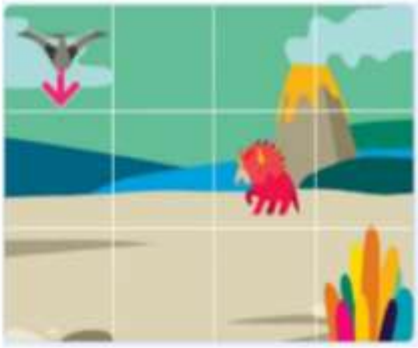
Doğrulama Kodu :BSC30DF2P2 Pin Kodu :54142 Belge Takip Adresi : https://turkiye.gov.tr/ebd?eK=4787&eD=BSC30DF2P2&eS=46916

34342 Bebek-İstanbul
Telefon No:0212 359 68 10 Faks No:0212 265 70 06
İnternet Adresi:www.sbe.boun.edu.tr
Kep Adresi:bogaziciuniversitesi@hs01.kep.tr
Bilgi için: Ayşe ÇEKİCİ
Unvan: Teknisyen

APPENDIX B

FIRST PILOT TEST

1. Ok yönünde hareket eden gri dinozor, arkadaşı kırmızı dinozorun olduğu kareye gitmek istiyor. Gri dinozor, seçeneklerdeki hangi kod grubu ile kırmızı dinozorun olduğu kareye gidebilir?



A) 

B) 

C) 

2. Pengueni, fok balığının olduğu kareye seçeneklerdeki hangi kod grubu ulaştırır?

A) 

B) 

C) 

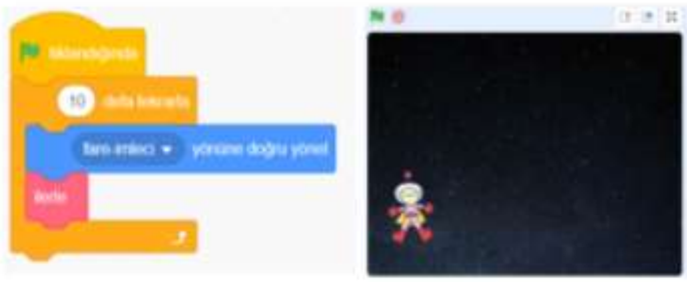
3. Papağan, aşağıdaki kod grubuna göre hareket ediyor. Ok yönünde harekete başlayan papağan seçeneklerdeki hangi kareye ulaşır?

A) 

B) 

C) 

4. Astronot oyun boyunca fareyi takip etmek istiyor. Astronotun fareyi takip edebilmesi için seçeneklerdeki hangi kod değiştirilmelidir?

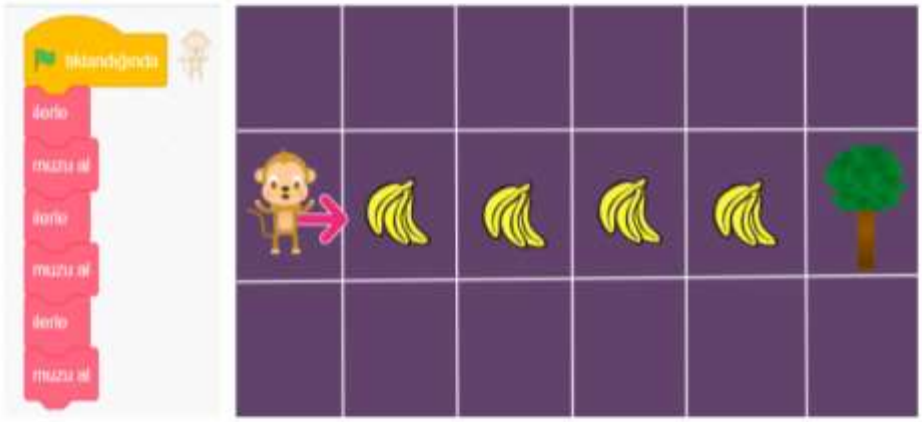


A) 

B) 

C) 

5. Maymun bütün muzları toplayarak ağaca gitmek istiyor. Bunu başarmak için verilen koda seçeneklerdeki hangi ekleme yapılmalıdır?

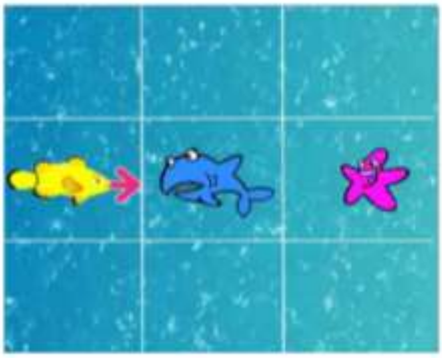


A) 

B) 

C) 

6. Ok yönünde ilerleyen sarı balık mor denizyıldızına gitmek istiyor. Sarı balık, seçeneklerdeki hangi kod grubu ile mavi köpek balığına değmeden mor denizyıldızının olduğu kareye ulaşır?

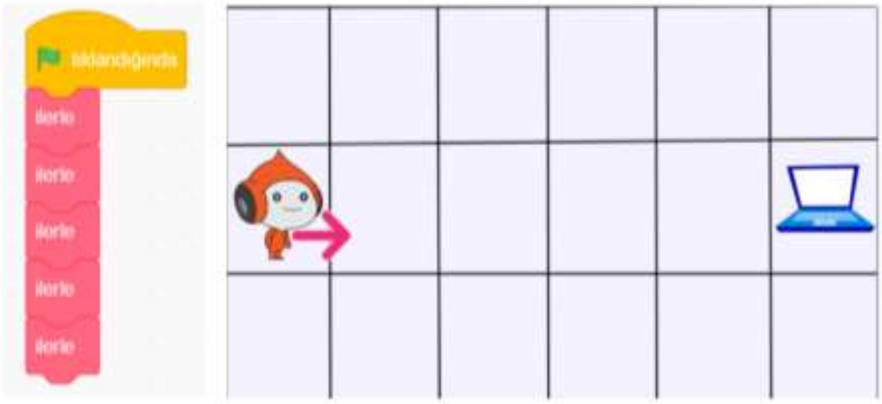


A)

B)

C)

7. Robot aşağıdaki kod grubu ile bilgisayara gidebiliyor. Robot, bir ilerle kodu ile bilgisayar gitmek isterse seçeneklerdeki hangi kod grubu ile gidebilir?

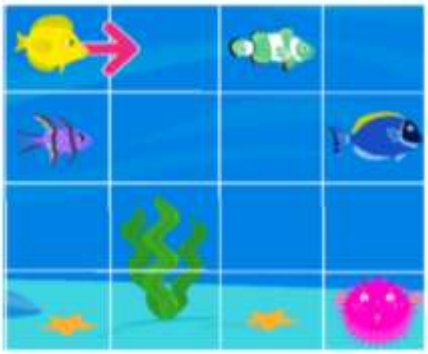


A) 

B) 

C) 

8. Sarı balık, pembe balon balığına gitmek istiyor. Sarı balık, sadece boş karelerden geçerek pembe balon balığının olduğu kareye seçeneklerdeki hangi kod grubu ile gidebilir?

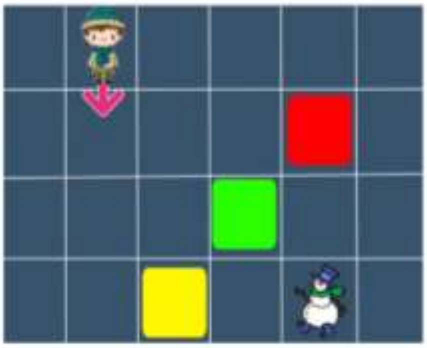


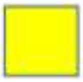
A)

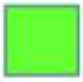
B)

C)

9. Ok yönünde hareket eden Mete, kardan adama gidiyor. Mete, aşağıdaki koda göre hangi renk kareden geçer?



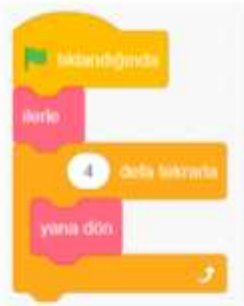
A) 

B) 

C) 

10. Kalem ok yönünde hareket ederek kare çizecektir. Kalem, seçeneklerdeki hangi kod grubu ile şekildeki gibi bir kare çizebilir?

A) 

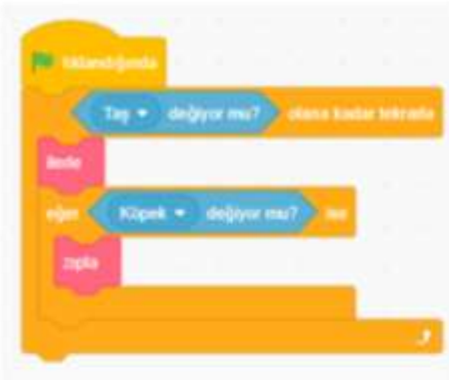
B) 

C) 

11. Melisa taşa değmeden köpeğine ulaşmak istiyor. Melisa, seçeneklerdeki hangi kod grubu ile köpeğine gidebilir?

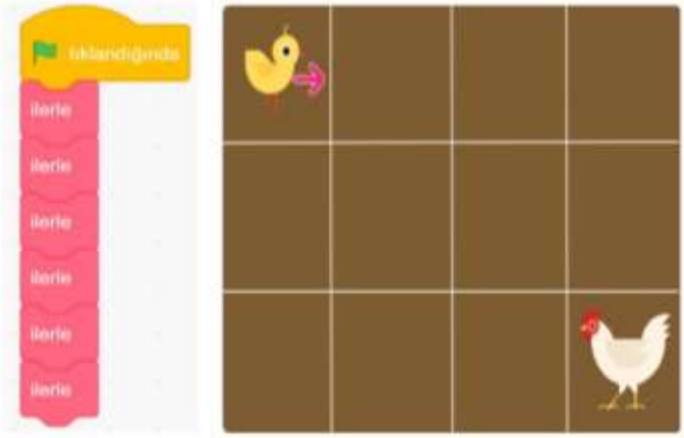
A)

B)



C)

12. Civciv horoza gitmek istiyor. Civcivin horoza gidebilmesi için seçeneklerde gösterilen hangi kodu <u>yana dön kodu</u> ile değiştirmek gerekir?

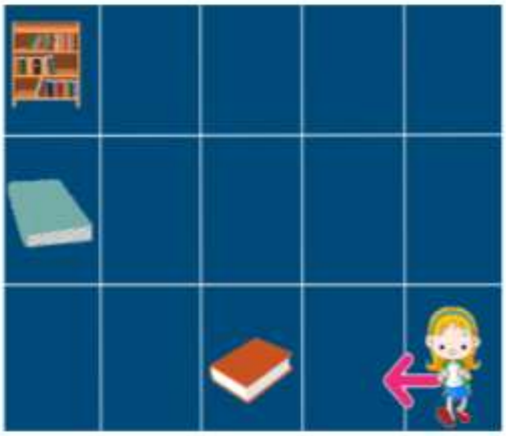


A)

B)

C)

13. Ela ok yönünde hareket ederek kitaplığa gitmek istiyor. Ela, kitapları alarak kitaplığa seçeneklerdeki hangi kod ile gidebilir?



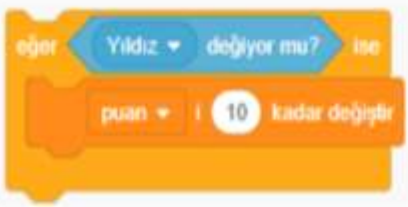
A)

B)

C)

14. Astronotun yıldıza değince 10 puan kazanması için hangi seçenekteki kod grubu kullanılmalıdır?

A)



B)

C)

15. Kırmızı anne dinozor, yeşil bebek dinozora aşağıdaki kod ile ulaşamıyor. Anne dinozorun bebek dinozora gidebilmesi için kodda seçeneklerdeki hangi değişiklik yapılmalıdır?

A) 

yerine

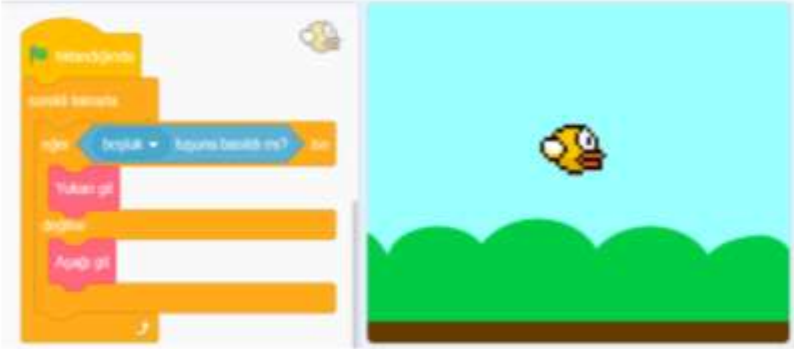

B) 

yerine


C) 

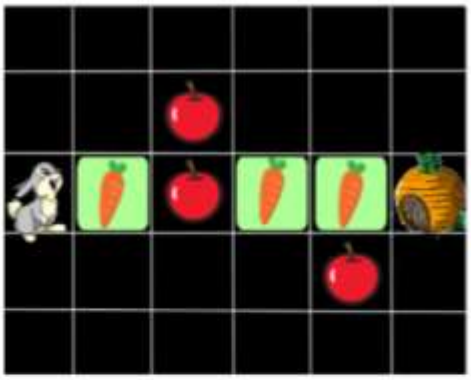
yerine


16. Kuş aşağıdaki kod grubuna göre hareket etmektedir. Koda göre boşluk tuşuna **basılmadığı** zaman kuş hangi yöne hareket eder?



A) Aşağıya B) Yukarıya C) Yana

17. Tavşan havuç toplayarak evine gitmek istiyor. Tavşan, **elmaları almadan sadece havuçları toplayarak** evinin olduğu kareye seçeneklerdeki hangi kod grubu ile gidebilir?

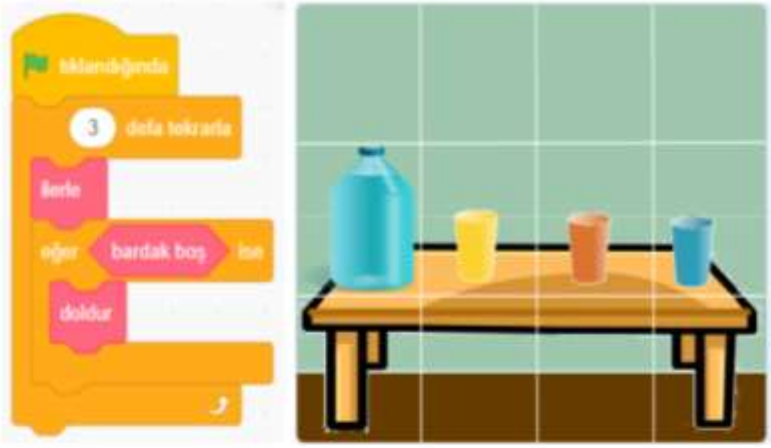


A) 

B) 

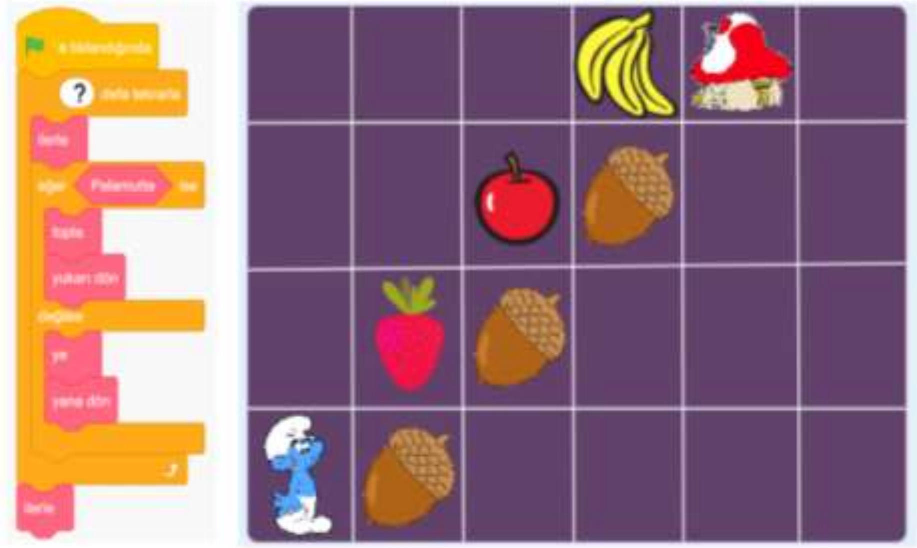
C) 

18. Şişe aşağıdaki kod grubuna göre hareket etmektedir. Koda göre, şişe **en çok** kaç bardağı su ile doldurabilir?



A) 1 B) 2 C) 3

19. Obur şirin, meyveleri yiyip palamutları toplayarak evine gitmek istiyor. Obur şirin, evine ulaşmak için aşağıdaki kod grubunu kaç kere tekrar etmelidir?



A) 3 B) 6 C) 7

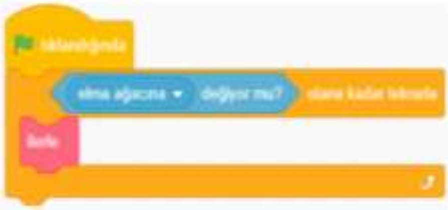
20. Aşağıdaki oyunda uzaylı, elmas toplayınca puan kazanıyor. İksir toplayınca puan kaybediyor. Oyunun bu şekilde çalışması için seçeneklerde gösterilen hangi kod değişmelidir?

A)

B)

C)

21. Tavşan aşağıdaki kod grubu ile elma ağacına gidemiyor. Tavşanın ağaca gidebilmesi için kod grubunda hangi seçenekteki gibi değişiklik yapılmalıdır?

A) 

B) 

C) 

22. Gemi, aşağıdaki kod grubu ile hazineye gitmek istiyor. Geminin hazineye ulaşması için seçeneklerdeki hangi iki kod yer değiştirmelidir?

A) 

ve 

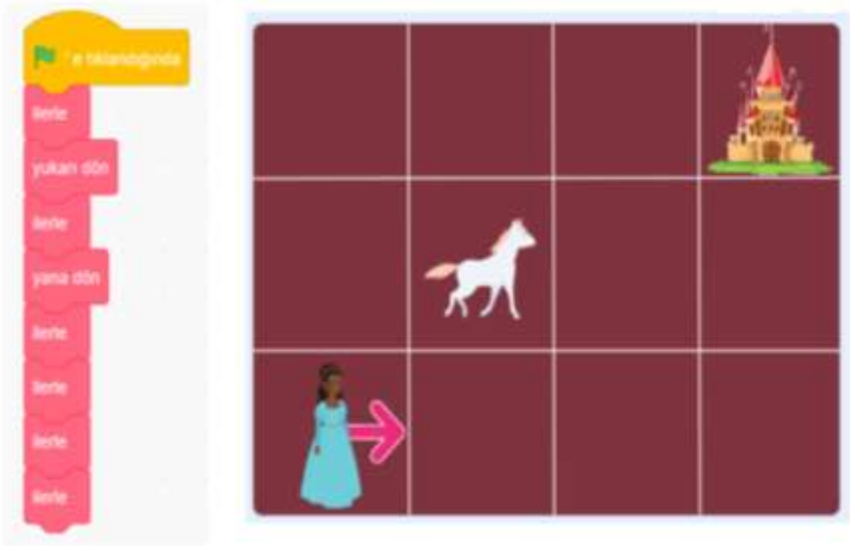
B) 

ve 

C) 

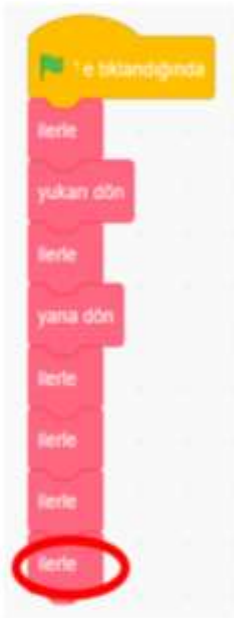
ve 

23. Prenses, beyaz atını alıp kalesine gitmek istiyor. Prensesin kaleye ulaşması için seçeneklerdeki hangi kod değiştirilmelidir?

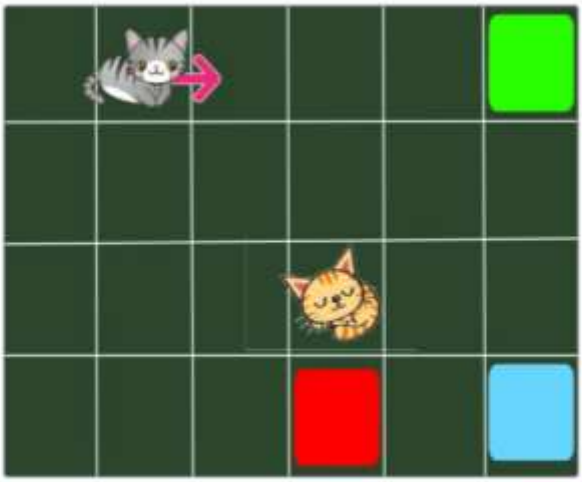


A)


B)


C)


24. Ok yönünde hareket eden gri kedi, sarı kediye gitmek istiyor. Gri kedi, seçeneklerdeki hangi kod grubu ile **sırasıyla** yeşil, mavi ve kırmızı karelerden geçip sarı kediye gidebilir?



A)

B)

C)

APPENDIX C

SECOND PILOT TEST

1. Pengueni, fok balığının olduğu kareye seçeneklerdeki hangi kod grubu ulaştırır?

A) 

B) 

C) 

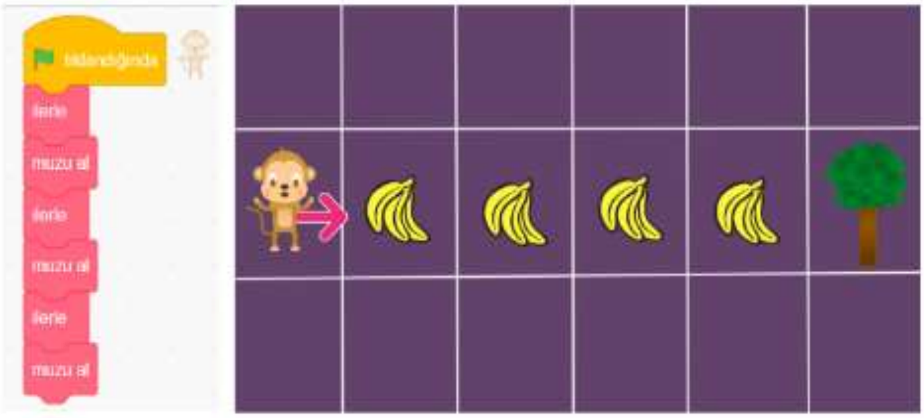
2. Papağan aşağıdaki kod grubuna göre hareket ediyor. Ok yönünde harekete başlayan papağan seçeneklerdeki yerlerden hangisinin olduğu kareye ulaşır?

A) 

B) 

C) 

3. Maymun bütün muzları toplayarak ağaca gitmek istiyor. Bunu başarmak için aşağıda verilen kod grubu, hangi seçenekteki kod ile devam etmelidir?

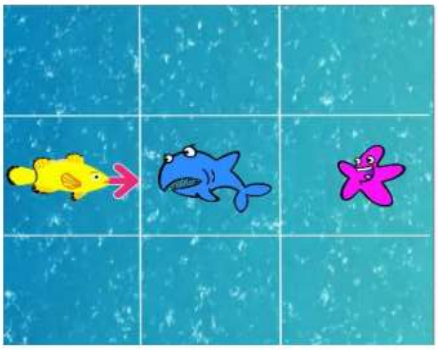


A) 

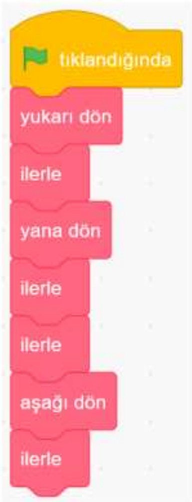
B) 

C) 

4. Ok yönünde ilerleyen sarı balık mor denizyıldızına gitmek istiyor. Sarı balık, seçeneklerdeki hangi kod grubu ile mavi köpek balığına <u>değmeden</u> mor denizyıldızının olduğu kareye ulaşır?

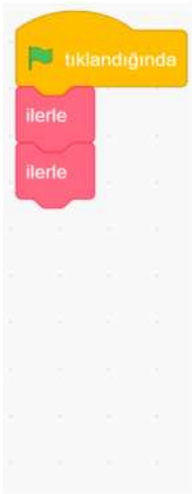


A)

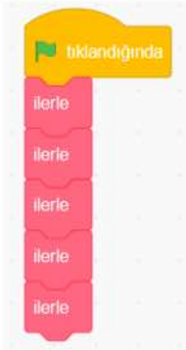


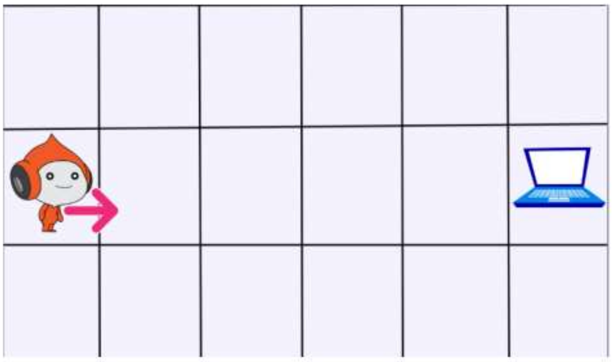
B)

C)



5. Robot aşağıdaki kod grubu ile bilgisayara gidebiliyor. Robot, sadece bir kere ilerle kodu kullanarak seçeneklerdeki hangi kod grubu ile bilgisayara gidebilir?



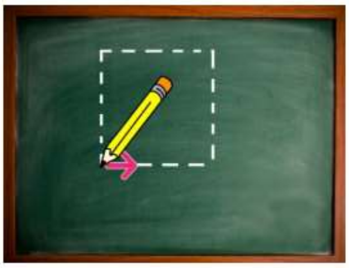


A) 

B) 

C) 

6. Kalem ok yönünde hareket ederek bir kare çizecektir. Kalem, seçeneklerdeki hangi kod grubu ile şekildeki gibi bir kare çizebilir?

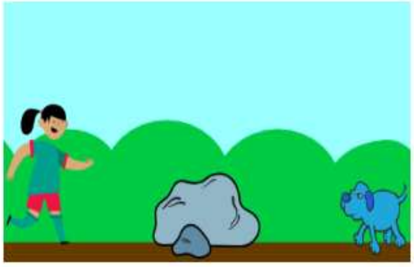


A) 

B) 

C) 

7. Zeynep taşa <u>değmeden</u> köpeğine ulaşmak istiyor. Zeynep, seçeneklerdeki hangi kod grubu ile köpeğine gidebilir?

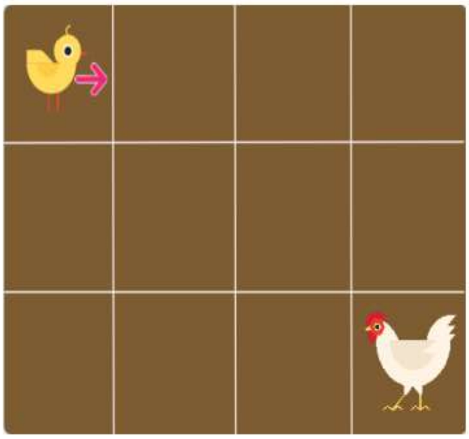


A)

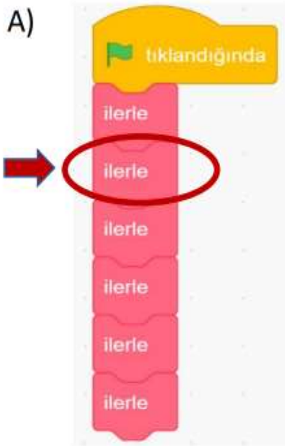
B)

C)

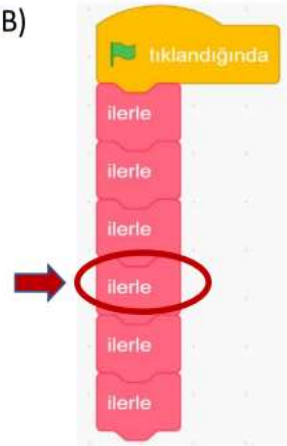
8. Civciv horoza gitmek istiyor. Civcivin horoza gidebilmesi için hangi seçenekte gösterilen kodu aşağı dön kodu ile değiştirmek gerekir?



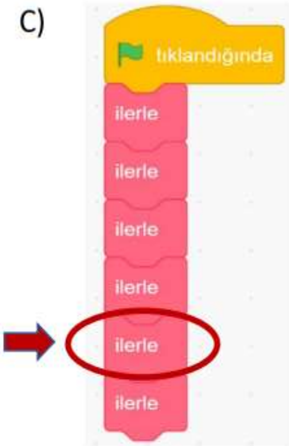
A)



B)



C)

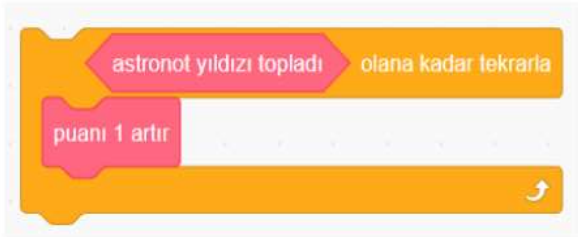


9. Astronotun yıldıza değince 10 puan kazanması için hangi seçenekteki kod grubu kullanılmalıdır?

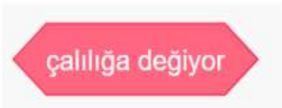
A)

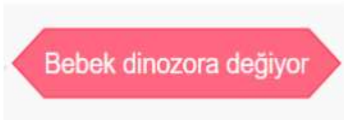
B)

C)

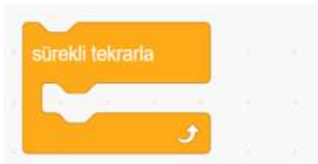


10. Kırmızı anne dinozor, yeşil bebek dinozora aşağıdaki kod ile ulaşamıyor. Anne dinozorun bebek dinozora gidebilmesi için kodda seçeneklerdeki hangi değişiklik yapılmalıdır?

A) 

yerine 

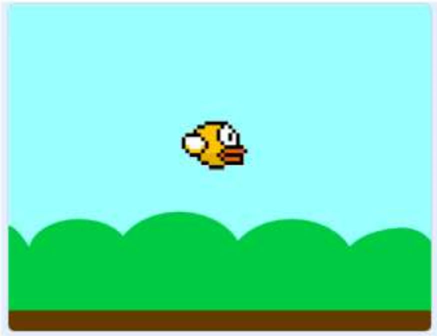
D) 

yerine 

E) 

yerine 

11. Kuş aşağıdaki kod grubuna göre hareket etmektedir. Koda göre boşluk tuşuna basılmadığı zaman kuş hangi yöne hareket eder?

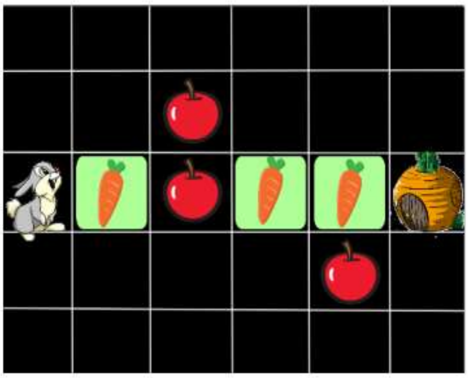


A) Aşağıya B) Yukarıya C) Yana

12. Mete aşağıdaki kod grubuna göre şişe ile bardaklara su dolduracaktır. Koda göre, Mete **en çok** kaç bardağı su ile doldurabilir?

A) 2 B) 3 C) 4

13. Tavşan havuç toplayarak evine gitmek istiyor. Tavşan, elmaları almadan sadece havuçları toplayarak evinin olduğu kareye seçeneklerdeki hangi kod grubu ile gidebilir?



A) 

B) 

C) 

14. Aşağıdaki oyunda uzaylı, elmas toplayınca puan kazanıyor. Toplayamazsa puan kaybediyor. Oyunun bu şekilde oynanması için seçeneklerde kırmızı ile gösterilen hangi kod değişmelidir?

A)

B)

C)

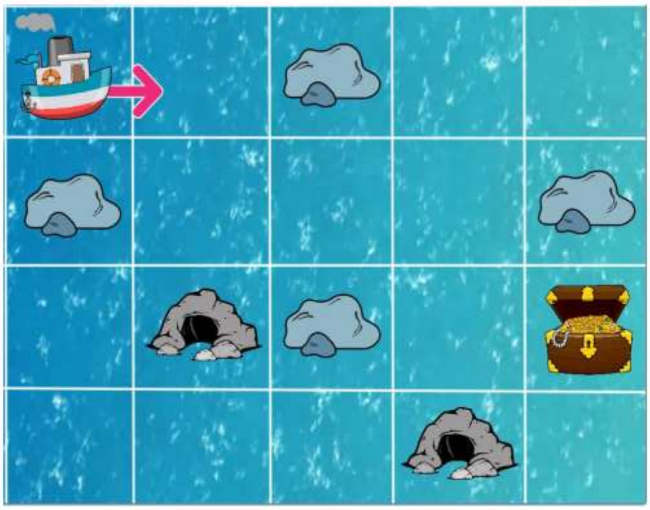
15. Tavşan aşağıdaki kod grubu ile elma ağacına gidemiyor. Tavşanın ağaca gidebilmesi için kodda seçeneklerdeki hangi değişiklik yapılmalıdır?

D) 

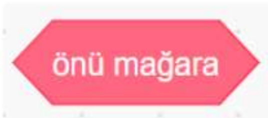
E) 

F) 

16. Gemi, aşağıdaki kod grubu ile hazineye gitmek istiyor. Geminin hazineye ulaşması için seçeneklerdeki hangi iki kod yer değiştirmelidir?



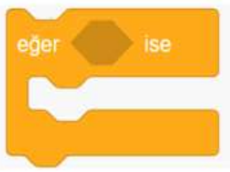
A) 

ve 

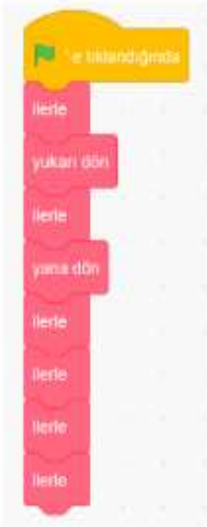
B) 

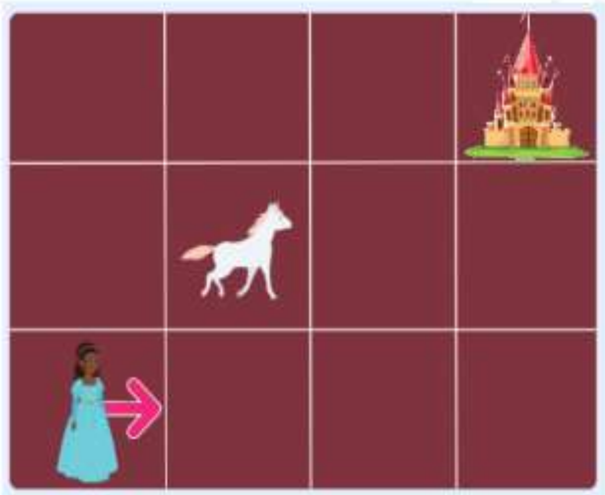
ve 

C) 

ve 

17. Prenses, beyaz atını alıp kalesine gitmek istiyor. Prensesin kaleye ulaşması için seçeneklerde gösterilen hangi kod değiştirilmelidir?

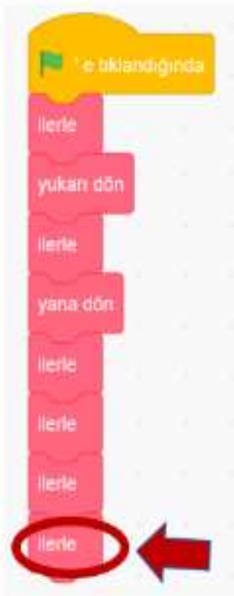




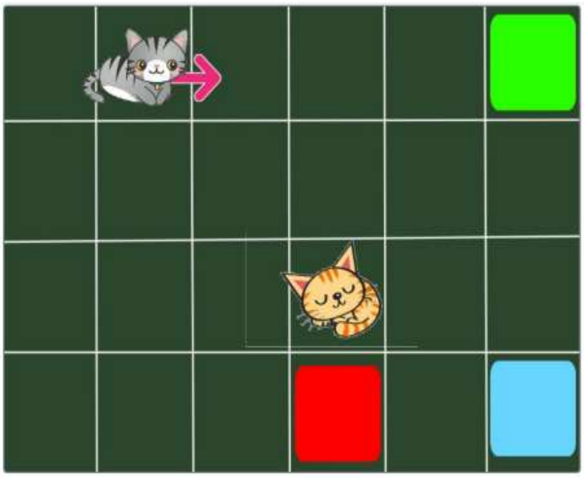
A)

B)

C)



18. Ok yönünde hareket eden gri kedi, sarı kediye gitmek istiyor. Gri kedi, seçeneklerdeki hangi kod grubu ile <u>sırasıyla</u> yeşil, mavi ve kırmızı karelerden geçip sarı kediye gidebilir?



A)

B)

C)

APPENDIX D

FINAL TEST

1. Pengueni, fok balığının olduğu kareye seçeneklerdeki hangi kod grubu ulaştırır?

A) 

B) 

C) 

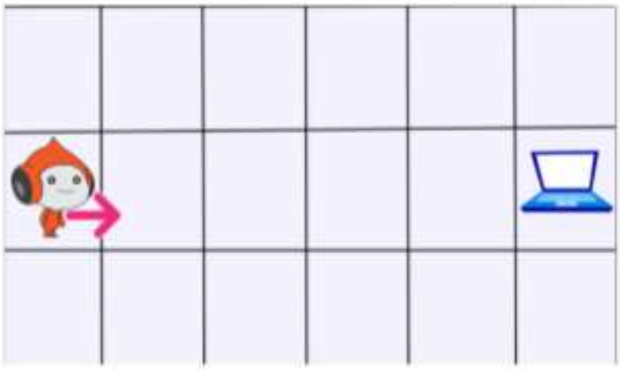
2. Papağan aşağıdaki kod grubuna göre hareket ediyor. Ok yönünde harekete başlayan papağan seçeneklerdeki yerlerden hangisinin olduğu kareye ulaşır?

A) 

B) 

C) 

3. Robotu bilgisayarın olduğu kareye seçeneklerdeki hangi kod grubu ulaştırır?

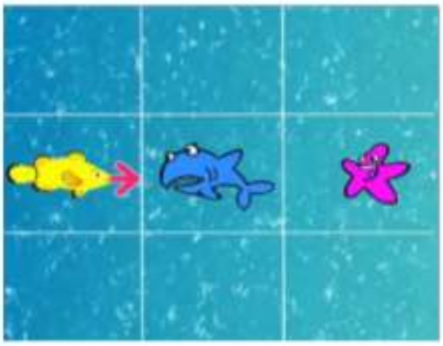


A) 

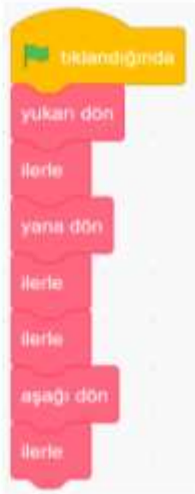
B) 

C) 

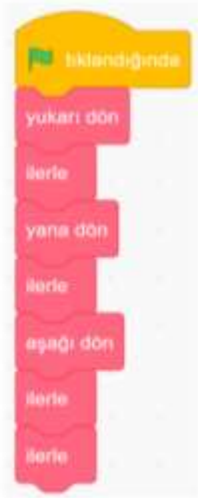
4. Ok yönünde ilerleyen sarı balık mor denizyıldızının olduğu kareye gitmek istiyor. Sarı balık, seçeneklerdeki hangi kod grubu ile mavi köpek balığına <u>değmeden</u> mor denizyıldızına ulaşır?



A)

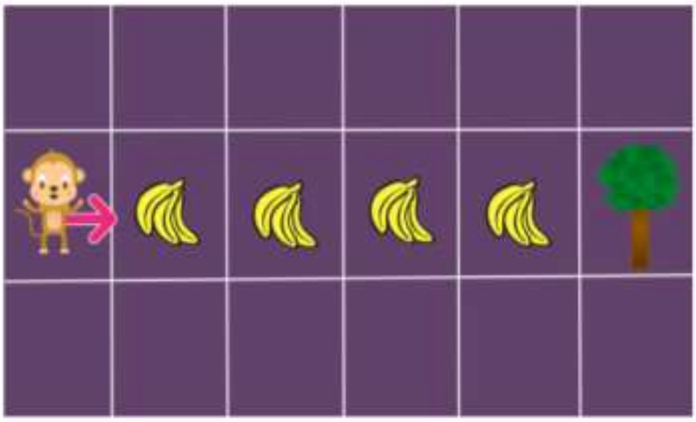


B)



C)

5. Maymun bütün muzları toplayarak ağaca gitmek istiyor. Maymunu ağacın olduğu kareye seçeneklerdeki hangi kod grubu ulaştırır?

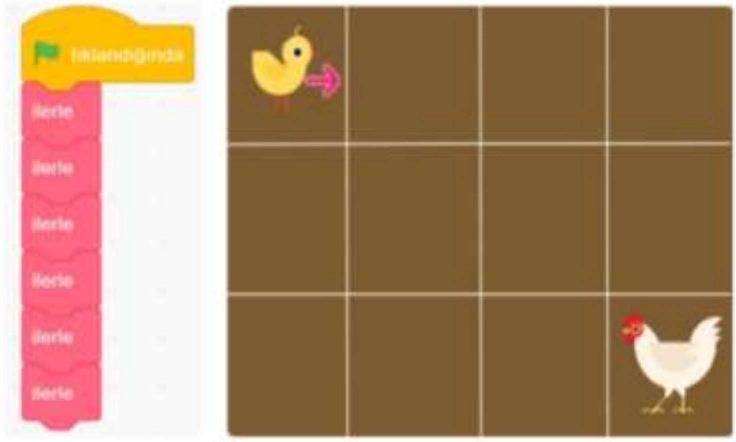


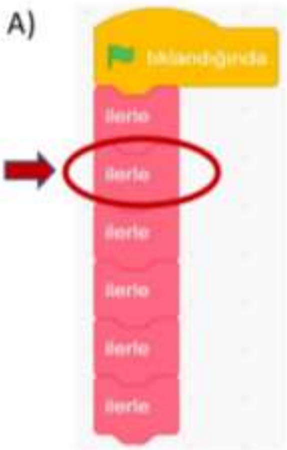
A)

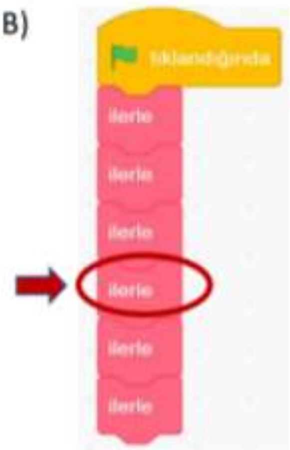
B)

C)

6. Civciv horozun olduğu kareye gitmek istiyor. Civcivin horoza gidebilmesi için hangi seçenekte gösterilen kod <u>aşağı dön kodu</u> ile değişmelidir?

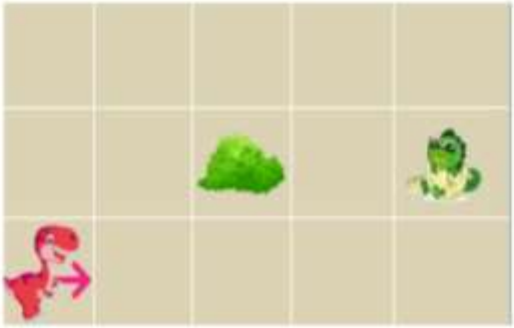


A) 

B) 

C) 

7. Kırmızı anne dinozor, yeşil bebek dinozorun olduğu kareye aşağıdaki kod ile ulaşamıyor. Anne dinozorun bebek dinozora gidebilmesi için kod seçeneklerdekilerden hangisi ile değişmelidir?

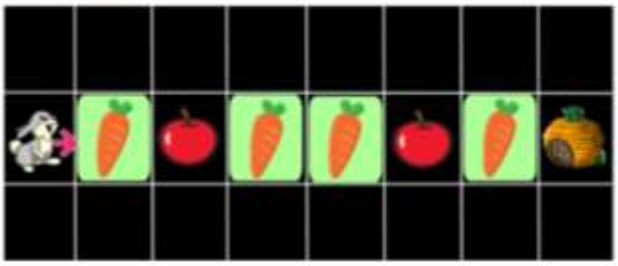


A)

B)

C)

8. Tavşan havuç toplayarak evine gitmek istiyor. Tavşan, <u>elmaları almadan sadece havuçları toplayarak</u> evinin olduğu kareye seçeneklerdeki hangi kod grubu ile gidebilir?

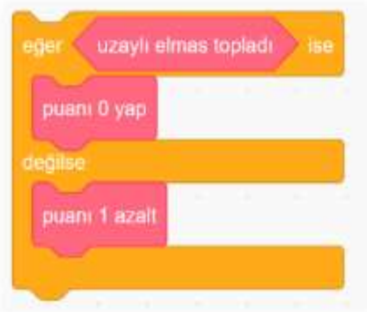


A) 

B) 

C) 

9. Aşağıdaki oyunda uzaylı, elmas toplayınca 1 puan kazanıyor. Toplayamazsa 1 puan kaybediyor. Oyunun bu şekilde oynanması için seçeneklerde kırmızı ile gösterilen hangi kod değişmelidir?



A)

B)

C)

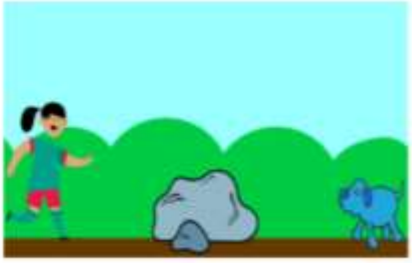
10.Tavşan aşağıdaki kod grubu ile elma ağacına gidemiyor. Tavşanın ağaca gidebilmesi için kod seçeneklerdekilerden hangisi ile değişmelidir?

A) 

B) 

C) 

11.Zeynep taşın üstünden zıplayarak köpeğine ulaşmak istiyor. Zeynep, seçeneklerdeki hangi kod grubu ile köpeğine gidebilir?

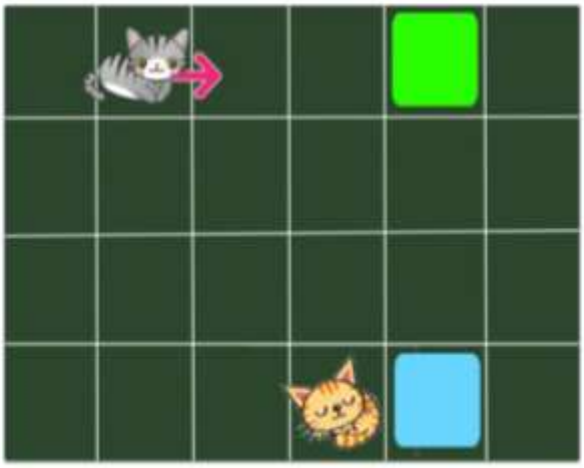


A)

B)

C)

12.Ok yönünde hareket eden gri kedi, sarı kedinin olduğu kareye gitmek istiyor. Gri kedi, seçeneklerdeki hangi kod grubu ile yeşil ve mavi karelerden geçip sarı kediye gidebilir?

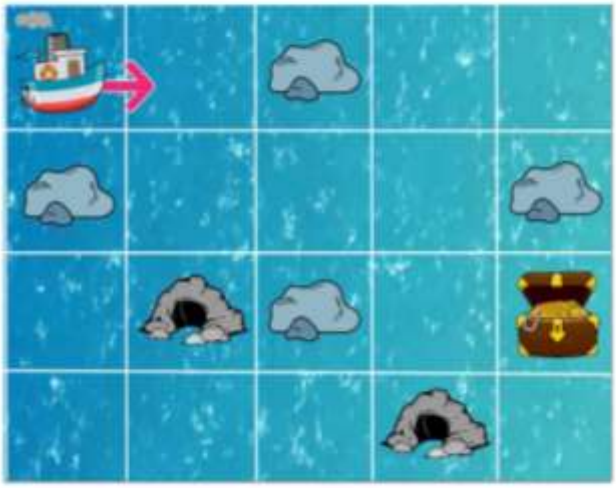


A)

B)

C)

13.Gemi aşağıdaki kod grubu ile hazineye gitmek istiyor. Geminin hazineye ulaşması için seçeneklerdeki hangi iki kod yer değiştirmelidir?



A)

B)

C)

REFERENCES

Aish, A.M. & Jöreskog, K.G. (1990). A panel model for political efficacy and responsiveness: an application of LISREL 7 with weighted least squares. *Khrwer Academic Publishers*, *24*, 405-426.

Araki, K., Furukawa, Z., & Cheng, J. (1991). A general framework for debugging. *IEEE Software*, *8*(3), 14–20. https://doi.org/10.1109/52.88939

Aytekin, A., Sönmez Çakır, F., Yücel, Y. B., & Kulaözü, İ. (2018). Geleceğe yön veren kodlama bilimi ve kodlama öğrenmede kullanılabilecek bazı yöntemler. *Eurasian Journal of Researches in Social and Economics (EJRSE), 5*(5), 24–41. https://dergipark.org.tr/tr/pub/asead/issue/40925/494055

Babazadeh, M., & Negrini, L. (2022). How is computational thinking assessed in European K-12 education? A systematic review. *International Journal of Computer Science Education in Schools*, *5*(4), 3–19. https://doi.org/10.21585/ijcses.v5i4.138

Baker, F. B. (2001). *The basics of item response theory* (2nd ed.). In C. Boston & L. Rudner (Eds.), ERIC Clearinghouse on Assessment and Evaluation.

Barr, V., & Stephenson, C. (2011). Bringing computational thinking to K-12: What is involved and what is the role of the computer science education community? *ACM Inroads*, *2*(1), 48–54. https://doi.org/10.1145/1929887.1929905

Batı, K. (2018). Computational thinking test (CTT) for middle school students. *Mediterranean Journal of Educational Research*, *12*(23), 89–101. https://doi.org/10.29329/mjer.2018.138.6

Bentler, P. M., & Chou, C. (1987). Practical issues in structural equation modeling. *Sociological Methods and Research, 16,* 78-117

Bers, M. U., Gonzalez-Gonzalez, C., & Armas-Torres, M. A. (2019). Coding as a playground: Promoting positive learning experiences in childhood classrooms. *Computers & Education*, *138*, 130–145.

Bers, M. U., Flannery, L., Kazakoff, E. R., & Sullivan, A. (2014). Computational thinking and tinkering: Exploration of an early childhood robotics curriculum. *Computers & Education*, *72*, 145-157. https://doi.org/10.1016/j.compedu.2013.11.013

Brennan, K., & Resnick, M. (2012). New frameworks for studying and assessing the development of computational thinking. In *Proceedings of the 2012 annual meeting of the American Educational Research Association*, (pp. 1–25). Vancouver, Canada. Retrieved from http://scratched.gse.harvard.edu/ct/files/AERA2012.pdf

Browne, M. W., & Cudeck, R. (1993). Alternative ways of assessing model fit. In K. A. Bollen & J. S. Long (Eds.), *Testing structural equation models* (pp. 137–162). Sage.

Cicchetti, D. V. (1994). Guidelines, criteria, and rules of thumb for evaluating normed and standardized assessment instruments in psychology. *Psychological Assessment*, *6*, 284–290. https://doi.org/10.1037/1040-3590.6.4.284.

Creswell, J. W. (2012). *Educational research: planning, conducting, and evaluating quantitative and qualitative research*. (4th ed.). Pearson Education.

Crocker, L. & Algina, J. (1986). *Introduction to classical and modern test theory*. Holt, Rinehart and Winston, Inc.

Cronbach, L. J. (1971). Test validation. In R. L. Thorndike (Ed.), *Educational Measurement* (pp. 443-507). Washington, DC: American Council on Education

Cruz Castro, L., Shoaib, H., & Douglas, K. (2021). Computational thinking frameworks used in computational thinking assessment in higher education. A Systematized Literature Review.10.18260/1-2--36824.

Çetin, İ., Otu, T., & Oktaç, A. (2020). Adaption of the computational thinking test into Turkish. *Turkish Journal of Computer and Mathematics Education*, *11*(2), 343-360.

Dagiene, V., & Jevsikova, T. (2014). Evaluation of the reliability and validity of the Bebras tasks. *Journal of Information Technology Education: Research*, *13*, 27-51. https://doi.org/10.28945/1933

Demir, Ö., & Seferoğlu, S. S. (2019). Developing a Scratch-based coding achievement test. *Information and Learning Sciences, 120*(5/6), 383–406. https://doi.org/10.1108/ILS-08- 2018-0078

Denning, P. J. (2010). Computational thinking. *Communications of the ACM, 53*(8), 27-29.

Doleck, T., Bazelais, P., Lemay, D. J., Saxena, A., & Basnet, R. B. (2017). Algorithmic thinking, cooperativity, creativity, critical thinking, and problem solving: Exploring the relationship between computational thinking skills and academic performance. *Journal of Computers in Education*, *4*(4), 355–369. https://doi.org/10.1007/s40692-017-0090-9

Edelen, M. O., & Reeve, B. B. (2007). Applying item response theory (IRT) modeling to questionnaire development, evaluation, and refinement. *Quality of Life Research*, *16*(SUPPL. 1), 5–18. https://doi.org/10.1007/s11136-007-9198-0

El-Hamamsy, L., Zapata-Cáceres, M., Marcelino, P., Bruno, B., Dehler Zufferey, J., Martin-Barroso, E., & Roman-Gonzalez, M. (2022). Comparing the psychometric properties of two primary school Computational Thinking (CT) assessments for grades 3 and 4: The Beginners' CT test (BCTt) and the competent CT test (cCTt). *Frontiers in Psychology, 13*, 1082659. https://doi.org/10.3389/fpsyg.2022.1082659

El-Hamamsy, L., Zapata Caceres, M., Martín, E., Mondada, F., Dehler Zufferey, J. & Bruno, B. (2022). The competent computational thinking test: development and validation of an unplugged computational thinking test for upper primary school. *Journal of Educational Computing Research, 60*, 073563312210817. https://doi.org/10.1177/07356331221081753.

Eryılmaz, S., & Deniz, G. (2019). Türkiye'de programlama eğitimi ile ilgili yapılan çalışmaların incelenmesi: Bir betimsel analiz çalışması. *Eğitimde Kuram ve Uygulama*, *15*(4), 319-338. doi:10.17244/eku.645387

Field, A. (2013). *Discovering statistics using IBM SPSS statistics*. Sage.

Futschek, G. (2006). Algorithmic thinking: The key for understanding computer science. Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 4226 LNCS(February), 159–168. https://doi.org/10.1007/11915355_15

Gay, L.R., Mills, G.E. & Airasian, P.W. (2012). *Educational Research: Competencies for Analysis and Application.* (10^{th} Eds.). Pearson, Upper Saddle River.

Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher, 42*(1), 38-43.

Guzdial, M. (2010). Paving the way for computational thinking. *Communications of the ACM*, *53*(11), 32-34.

Gülbahar, Y., Kert, S. B., & Kalelioğlu, F. (2019). Bilgi işlemsel düşünme becerisine yönelik öz yeterlik algısı ölçeği: Geçerlik ve güvenirlik çalışması. *Turkish Journal of Computer and Mathematics Education*, *10*(1), 1–29. https://doi.org/10.16949/turkbilmat.385097

Hambleton, R.K. & Jones, R. W. (1993). Comparison of classical test theory and item response theory and their applications to test development. *Educational Measurement: Issues and Practice*, 38-47.

Hu, L., & Bentler, P. M. (1998). Fit indices in covariance structure modeling: Sensitivity to underparameterized model misspecification. *Psychological Methods*, *3*(4), 424–453. https://doi.org/10.1037//1082-989x.3.4.424

ISTE (2015). ISTE standards for computational thinking. Retrieved from https://www.iste.org/standards/iste-standards-for-computational-thinking

ISTE (2021). ISTE standards for computational thinking. Retrieved from https://www.iste.org/standards/iste-standards-for-computational-thinking

João, P., Nuno, D., Fábio, S. F., & Ana, P. (2019). A cross-analysis of block-based and visual programming apps with computer science student-teachers. *Education Sciences, 9*(3). https://doi.org/10.3390/educsci9030181

Jöreskog, K. G. (1994). *Structural equation modeling with ordinal variables*. 297–310. https://doi.org/10.1214/lnms/1215463803

Kafai, Y. B., DeLiema, D., Fields, D. A., Lewandowski, G., & Lewis, C. (2019). Rethinking debugging as productive failure for CS education. *SIGCSE 2019 - Proceedings of the 50th ACM Technical Symposium on Computer Science Education*, 169–170. https://doi.org/10.1145/3287324.3287333

Kafai, Y. B., & Proctor, C. (2022). A revaluation of computational thinking in K–12 education: Moving toward computational literacies. *Educational Researcher*, *51*(2), 146–151. https://doi.org/10.3102/0013189X211057904

Kafai, Y. B., & Resnick, M. (1996). *Constructionism in practice: Designing, thinking, and learning in a digital world*. Routledge.

Kalelioğlu, F. (2015). A new way of teaching programming skills to K-12 students: Code.org, *Computers in Human Behavior*, *52*, 200-210.

Kalelioğlu, F. & Gülbahar, Y. (2014). The effects of teaching programming via scratch on problem solving skills: A discussion from learners' perspective. *Informatics in Education*, *13*, 33-50.

Kalelioğlu, F., & Gülbahar, Y. (2015). Bilge Kunduz: Uluslararası enformatik yarışması pilot uygulama sonuçları. *9th International Computer & Instructional Technologies Symposium*. Afyonkarahisar, Turkey, May 20-22.

Kalelioğlu, F., Gülbahar, Y., & Kukul, V. (2016). A framework for computational thinking based on a systematic research review. *Baltic Journal of Modern Computing*, *4*(3), 583.

Kanaki, K. & Kalogiannakis, M. (2019). Assessing computational thinking skills at first stages of schooling. *ICEEL 2019: 2019 3rd International Conference on Education and E-Learning,* (pp. 135-139). 10.1145/3371647.3371651.

Karabak, D., & Güneş, A. (2013). Curriculum Proposal for first class secondary School students in the field of Software Development. *Journal of Research in Education and Teaching*, *2*(3), 163–169.

Kim, C. M., Yuan, J., Vasconcelos, L., Shin, M., & Hill, R. B. (2018). Debugging during block-based programming. *Instructional Science, 46,* 767-787.

Kong, S. C., & Liu, B. (2020). A performance-based assessment platform for developing computational thinking concepts and practices: EasyCode. *Bulletin of the Technical Committee on Learning Technology*, *20*(2), 3–10.

Li, C., Chan, E., Denny, P., Luxton-Reilly, A., & Tempero, E. (2019). Towards a framework for teaching debugging. In *ACM International Conference Proceeding Series,* 79–86. Association for Computing Machinery. https://doi.org/10.1145/3286960.3286970

Maloney, J., Resnick, M., Rusk, N., Silverman, B., & Eastmond, E. (2010). The scratch programming language and environment. *ACM Transactions on Computing Education*, *10*(4), 1–15. https://doi.org/10.1145/1868358.1868363

McCowan, R. J. & McCowan, S. C. (1999). Item analysis for criterion-references tests. *Center for Development of Human Services*

Mehri, B. (2017). From Al-Khwarizmi to algorithm. *Olympiads in Informatics*, *11*(Special Issue), 71–74. https://doi.org/10.15388/ioi.2017.special.11

MoNE. (2018). *Bilişim teknolojileri ve yazılım dersi öğretim programı (Ortaokul 5 ve 6. sınıflar)*. Ankara

Morales-Navarro, L., Fields, D. A., & Kafai, Y. B. (2021). Growing Mindsets: Debugging by Design to Promote Students' Growth Mindset Practices in Computer Science Class. *International Conference of the Learning Sciences, June*, 362–369.

Muller, O. (2005). Pattern oriented instruction and the enhancement of analogical reasoning. *Proceedings of the 1st International Computing Education Research Workshop, ICER 2005*, 57–67. https://doi.org/10.1145/1089786.1089792

Mullis, I. V. S., & Hooper, M. (2015). *Methods and Procedures in TIMSS Advanced 2015*.

Mullis, I. V., Martin, M. O., Ruddock, G. J., O'Sullivan, C. Y., & Preuschoff, C. (2009). TIMSS 2011 assessment frameworks. Amsterdam: International Association for the Evaluation of Educational Achievement

Pallant, J. (2016). *SPSS Survival Manual*. Open University Press

Papert, S. (1980). Mindstorms Children, Computers, and Powerful Ideas.

Parker, M. C., Kao, Y. S., Saito-Stehberger, D., Franklin, D., Krause, S., Richardson, D., & Warschauer, M. (2021). Development and preliminary validation of the assessment of computing for elementary students (ACES). *SIGCSE 2021 - Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*, 10–16. https://doi.org/10.1145/3408877.3432376

Relkin, E., & Bers, M. (2021). TechCheck-K: A measure of computational thinking for kindergarten children. *IEEE Global Engineering Education Conference, EDUCON*, 1696–1702. https://doi.org/10.1109/EDUCON46332.2021.9453926

Relkin, E., de Ruiter, L., & Bers, M. U. (2020). TechCheck: Development and validation of an unplugged assessment of computational thinking in early childhood education. *Journal of Science Education and Technology*, *29*(4), 482–498. https://doi.org/10.1007/s10956-020-09831-x

Román-González, M., Pérez-González, J. C., & Jiménez-Fernández, C. (2017). Which cognitive abilities underlie computational thinking? Criterion validity of the Computational Thinking Test. *Computers in Human Behavior, 72*, 678–691. https://doi.org/10.1016/j.chb.2016.08.047

Román-González, M. (2015). Computational thinking test: Design guidelines and content validation. *Proceedings of the 7th Annual International Conference on Education and New Learning Technologies (EDULEARN 2015), July 2015*, 2436–2444. https://doi.org/10.13140/RG.2.1.4203.4329

Rovinelli, R. J., & Hambleton, R. K. (1977). On the use of content specialists in the assessment of criterion-referenced test item validity. *Dutch Journal of Educational Research, 2*, 49–60.

Ruiter, L.E. & Bers, M.U. (2021): The Coding Stages Assessment: development and validation of an instrument for assessing young children’s proficiency in the ScratchJr programming language. *Computer Science Education*. DOI:10.1080/08993408.2021.1956216

Schumacker, R. E., & Lomax, R. G. (2010). *A beginner’s guide to structural equation modeling* (3rd ed.). Taylor and Francis Group, LLC. https://doi.org/10.1111/j.1467985x.2012.01045_12.x

Selby, C. C. (2014). How can the teaching of programming be used to enhance computational thinking skills? (Unpublished PhD thesis). University of Southampton.

Sengupta, S. (2016). Computational thinking in K-12 education: A review of the research literature. *Journal of Educational Research and Practice*, *6*(1), 47-64.

Tabachnick, B. G., & Fidell, L. S. (2013). *Using multivariate statistics* (6th ed.). Boston, MA: Pearson.

Tang, X., Yin, Y., Lin, Q., Hadad, R., & Zhai, X. (2020). Assessing computational thinking: a systematic review of empirical studies. *Computers & Education, 148*, https://doi.org/10.1016/j.compedu.2019.103798

Threekunprapa, A., & Yasri, P. (2020). Unplugged coding using flowblocks for promoting computational thinking and programming among secondary school students. *International Journal of Instruction*, *13*(3), 207–222. https://doi.org/10.29333/iji.2020.13314a

Tsai, M. J., Chien, F. P., Wen-Yu Lee, S., Hsu, C. Y., & Liang, J. C. (2022). Development and validation of the computational thinking test for elementary school students (CTT-ES): Correlate CT competency with CT disposition. *Journal of Educational Computing Research*, *60*(5), 1110–1129. https://doi.org/10.1177/07356331211051043

Tsai, M. J., Liang, J. C., & Hsu, C. Y. (2021). The computational thinking scale for computer literacy education. *Journal of Educational Computing Research*, *59*(4), 579–602. https://doi.org/10.1177/0735633120972356

Tsai, M. J., Wang, C. Y., & Hsu, P. F. (2019). Developing the computer programming self-efficacy scale for computer literacy education. *Journal of Educational Computing Research*, *56*(8), 1345–1360. https://doi.org/10.1177/0735633117746747

Ullman, J. B. (2001). Structural equation modeling. *In: B. G. Tabachnick, & L. S. Fidell (Eds.)*, Using multivariate statistics. Boston, MA: Pearson Education.

Voogt, J., Fisser, P., Good, J., Mishra, P., & Yadav, A. (2015). Computational thinking in compulsory education: Towards an agenda for research and practice. *Education and Information Technologies*, *20*(4), 715-728.

Weintrop, D., Wise Rutstein, D., Bienkowski, M., & McGee, S. (2021). Assessing computational thinking: an overview of the field. *Computer Science Education*, *31*(2), 113–116. https://doi.org/10.1080/08993408.2021.1918380

Weintrop, D., Beheshti, E., Horn, M., Orton, K., Jona, K., Trouille, L., & Wilensky, U. (2016). Defining computational thinking for mathematics and science classrooms. *Journal of Science Education and Technology, 25*(1), 127–147. https://doi.org/10.1007/s10956-015-9581-5

Wing, J. M. (2017). Computational thinking's influence on research and education for all. *Italian Journal of Educational Technology*, *25*(2), 7–14. https://doi.org/10.17471/2499-4324/922

Wing, J. M. (2016). Computational thinking, 10 years later. Retrieved fromhttp://www.microsoft.com/enus/research/blog/computational-thinking-10-years-later.

Wing, J. M. (2010). Computational Thinking: What and Why? *The Link - The Magazine of the Carnegie Mellon University School of Computer Science*, March 2006, 1–6.

Wing, J. M. (2008). Computational thinking and thinking about computing. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, *366*(1881), 3717–3725. https://doi.org/10.1098/rsta.2008.0118

Wing, J. M. (2006). Computational thinking. *Communications of the ACM*, *49*(3), 33-35. doi:10.1145/1118178.1118215.

Yadav, A., Zhou, N., Mayfield, C., Hambrusch, S., & Korb, J. T. (2011). Introducing computational thinking in education courses. In *SIGCSE'11 - Proceedings of the 42nd ACM Technical Symposium on Computer Science Education*, (pp. 465–470). https://doi.org/10.1145/1953163.1953297

Zapata Cáceres, M., Roman-Gonzalez, M., & Martín-Barroso, E. (2021). *BCTt: Beginners computational thinking test*. Retrieved from www.raspberrypi.org in April 2021.

Zapata-Caceres, M., Martin-Barroso, E., & Roman-Gonzalez, M. (2020). Computational thinking test for beginners: Design and content validation. *IEEE Global Engineering Education Conference, EDUCON,* (pp. 1905–1914).

Zhang, S., Wong, G. K. W., & Pan, G. (2021). Computational thinking test for lower primary students: Design principles, content validation, and pilot testing. *TALE 2021 - IEEE International Conference on Engineering, Technology and Education, Proceedings*, (pp. 345–352, January 2022). IEEE https://doi.org/10.1109/TALE52509.2021.9678852

Zhang, L., & Nouri, J. (2019). A systematic review of computational thinking research in K- 12 education. *Educational Research Review, 27*, 246-261.

Zumbo, B. D. (1999). *A Handbook on the theory and methods of differential item functioning (DIF): Logistic regression modeling as a unitary framework for binary and likert-type (ordinal) item scores.* Directorate of Human Resources Research and Evaluation, Department of National Defense. Ottowa, ON, Canada