Proposers:

Ka Ki Fung, Claremont McKenna College, kfung20@cmc.edu

Thitaree Tanprasert, Harvey Mudd College, ttanprasert@hmc.edu

Research Advisor:

Professor Colleen Lewis, Harvey Mudd College, lewis@cs.hmc.edu

Category:

Undergrad

Title:

Problem Decomposition in Computer Programming and Spatial Reasoning

Abstract

Previous research has documented correlations between spatial reasoning ability and success in STEM fields [8, 9, 10]. We aim to better understand this correlation, specifically between CS (computer science) performance in an introductory course and spatial reasoning [3, 7, 9]. To better understand this correlation, we observe the common characteristics between students' strategies in solving CS and spatial reasoning problems. We conducted interviews with eight students who have prior experience in CS but have never taken more than two introductory-leveled CS courses in college. In the interview, we asked the participants to solve a total of four problems; two CS performance problems and two spatial reasoning problems. The CS performance problems were code-reading problems in Python, this includes basic programming concepts, such as nested loops and functions. We observed the participants' problem-solving strategies and asked participants to explain their strategies regarding each problem. Then, we analyzed the participants' explanations in connection to our observation of their problem-solving process and their scratch work. From these evidences, we observed that, for both types of problem, participants appeared to first observe the problem for its fundamental structures then decomposes the problem into smaller sub-problems. From this observation, we conjecture that problem decomposition may be a required skill to solve both computer programming and spatial reasoning problems, and thus, it could be a factor that contributes to the correlation between these two fields. Further study into utilization of problem decomposition in CS and spatial reasoning may provide more insights into the correlation between success in CS and spatial reasoning ability.

1 Introduction

While connection between spatial reasoning and other STEM fields like physics and calculus seem obvious, as a field, we do not have theories to explain the correlation between CS performance and spatial reasoning. Based on previous research that has found correlations between performance in CS and spatial reasoning [8, 9], we believe that it is important explore this correlation because understanding the mechanism that produces this correlation could help improve pedagogy.

From existing literature and our observation from our eight research interviews, we explore the relationship between CS performance and spatial reasoning focusing on the presence of problem decomposition skill in novice programmers' problem-solving schemes. Our preliminary observation shows that problem decomposition appeared to be a required step in solving both CS and spatial reasoning problems. With further analysis, we gain deeper and more detailed understanding into how problem decomposition bridges these two fields. We also hope that this research could provide a basis and hypothesis for additional research related to this correlation.

The primary goal of this work is to better understand the correlation between CS performance in an introductory course and spatial reasoning by analyzing participants' problem-solving strategies. And the primary contribution is proposing that problem decomposition may be a foundation for both solving introductory CS problems and spatial reasoning tasks.

2 Background and Related Work

There have been decades of research documenting the importance of spatial ability in educational pursuits in STEM (science, technology, engineering, and mathematics) domains [10]. They had 400,000 participants who were tracked for 11+ years and their longitude findings were aligned with historical findings. They found that for decades, spatial ability was “salient psychological attribute among those adolescents who subsequently go on to achieve advanced educational credentials and occupations in STEM.” Such quantitative research is the basis of our research, proving the existence of a solid correlation between STEM performance and spatial reasoning, in which we could investigate and explore.

In the field of CS, there have been attempts to find the reason or cause of this correlation. Tai, Tu, Lai, and Lin study the effects of spatial ability in promoting logical thinking abilities of students with regard to programming language. [9] They found that students with high spatial ability scored significantly higher than those with low spatial ability in logical thinking ability. And those who had better logical thinking ability had significantly better achievements in computer programming than those with lower scores. Another research found a correlation between spatial reasoning and specifically for introductory computer programming education. [2] These researches provide a starting point for further exploration in the exact correlation between spatial reasoning and computer programming.

One of the most important skills in programming is problem decomposition. Problem decomposition is the ability to divide the problem into multiple, organized components and tackle each component, independently, before combining them to create the final solution for the problem. Problem decomposition is an important component of computational thinking, including, but not limited to programming. In the context of computer science, it is used for dividing objects, methods, and functions, as well as overseeing the order of operations. [1] Computational Thinking is an emerging term in educational research and is believed to be a deterministic characteristic for children’s competency in STEM education. [4] These researches suggest that we may gain insights into the contributing factors of the development of programming ability through problem decomposition skill.

3 Methods

3.1 Data Collection Methods

For our research, we interviewed eight participants who had taken introductory level of knowledge for CS, those who had taken no more than two CS courses in undergraduate school. With those classes, they developed python or java skill that allows them to understand conditionals, loops, and how to read code. The interviews were one on one ranging from 25-45 minutes, depending on the speed of participant’s ability to solve coding and spatial reasoning problems, and how in depth they go into explaining their strategies.

We asked the participants a total of five questions. The first question is a CS question. We gave them a nested loop that prints out lines of text, and they had to figure out what will be printed. The second question is another CS question, but this time with multiple choices provided. This question was able operations and figuring out which order of operation gave the desired output. The third question was a spatial reasoning question where multiple choices are provided, and participants have to find the correct reflection of the original image. The fourth question was another spatial reasoning question with multiple choices provided asking the participant how many blocks make up this structure. Lastly, there was one reflection question asking the participants what they thought about the correlation between CS and spatial reasoning. For each question except the reflection question, we told participants that they will be timed, and we gave them the questions. After they inform us they had reached a solution, the interviewers ask them to explain their strategy, this part is not timed.

We answered when participant asked questions about syntax, however, we decided that we would not help participant reach the solution. We also sometimes deviated from the script as different questions arises from the participants’ thinking and explanation of their strategy for problem solving.

3.2 Problem Description

We had four problems for participants to solve in the interview. After solving each problem, we asked them to explain their solution in detail, as if they are teaching this problem to others. At the end, we asked the participants for their opinions on correlation between CS performance and spatial reasoning and what they thought was the insight of this correlation.

The first two problems in the interview are code-reading problems, taken from research on introductory CS language assessment [7] which contains language-independent code-reading problems. We modified the codes in the problems we selected to be in Python, since Python was a common language that all of our participants know.

The first problem (Q1, see appendix) provides the following code and asks for the output of executing the code. This problem focuses on the ability to trace nested loops, and the understanding of list indices manipulation. The participant needs to calculate i-j from tracing of loop parameters before retrieving the letter at that index from the list.

The second problem (Q2, see appendix) focuses on the ability to understand the relationship between functions and variables, specifically how values in variables are affected by operations and how they are stored. It also requires tracing but in a less conventional form that differs from most problems in introductory CS course. This means that the participant would have to come up with a novel approach to solve it.

The 3rd and 4th problems in the interview are spatial reasoning problems taken from online source. [11] No prior experience is required to solve these problems. We ask the participant to solve the problem and explain briefly about their thought processes. The spatial reasoning problems have arguably more possible approaches towards the solution. So, we are more interested in the overall approach than the details of the problem-solving process.

The 3rd problem (Q3, see appendix) asks for a possible reflection of the given grid pattern. To solve this problem, it is important to recognize the relationship between positions of objects as well as possible manipulation and direction of each object.

The 4th problem (Q4, see appendix) asks for the number of blocks shown in the figure. To solve this problem, it is important to understand the relationship between a 3-dimensional figure and its 2-dimensional representation, in order to determine the number blocks which are not explicitly shown in the picture.

3.3 Data Analysis Methods

After the interviews, we noted a few interesting patterns. One of which was how participants use problem decomposition to solve both CS and spatial reasoning, which developed into the focus of our work.

To make sure we are observing all parts of the interview, we created content logs, which are transcripts of the interviews. [3] From these content logs, we listed out relevant episodes in the interview videos, which includes both the participants' explanation and their writing when they were solving the problems. We focused on the participants' strategies to problem solving and noticed that there was an interesting recurring theme for every participant and in almost every interview problem. That is, we observe that participants broke down the problems into smaller sub-problems for each question. This observation matched with our immediate opinions at the end of each interview. Therefore, we decided to do a more thoroughly analysis focusing on this topic.

Once we identified the key data on the content log and interpreted those evidence, we further analyzed participants' strategies and observed how problem decomposition is utilized when solving code-reading problems compared to spatial reasoning problems. In addition, we focused our literature review on problem decomposition in CS and spatial reasoning, in order to analyze problem decomposition from our interviews in details. We selected data from every interview problem, to see the different ways in which problem decomposition is used. These data come from several participants, since we found it interesting that participants used problem decomposition regardless of how much experience they had in CS. In the analysis section, we will present four examples. Each example came from a different participant and a different problem.

4 Analysis

The claim of our paper is that CS performance problems and spatial reasoning problems both require the ability to decompose problems. To show this, we will present evidence that students use problem decomposition to solve both CS and spatial reasoning problems in our interviews. Specifically, we will show that problem decomposition leads to pattern recognition or simplified computation, that helps participants reach the solution successfully. The evidence will include the participants' explanation of their strategies, our observation of their problem-solving process, as well as scans and screenshots of their scratch work.

The participants in our interview are eight students, who have taken CS courses in high school and/or college but have taken no more than two CS courses in college. All participants have experience with CS concepts presented in the interview questions, such as functions and nested loops.

Below are summaries and analyses of four episodes from our interviews. In our transcription below, we use "I" to indicate interviewer's statements.

4.1 Problem Decomposition in CS problems

4.1.1 Episode with Q1

In this episode, the participant, Nelson was asked to solve Q1 (see appendix). After he finished solving the problem, he was asked to explain his solution as though he was teaching someone who was not very familiar with the concept of nested loops. This episode contains our observation of his problem-solving process, the summary of his explanation, and the connection between our observation and his explanation.

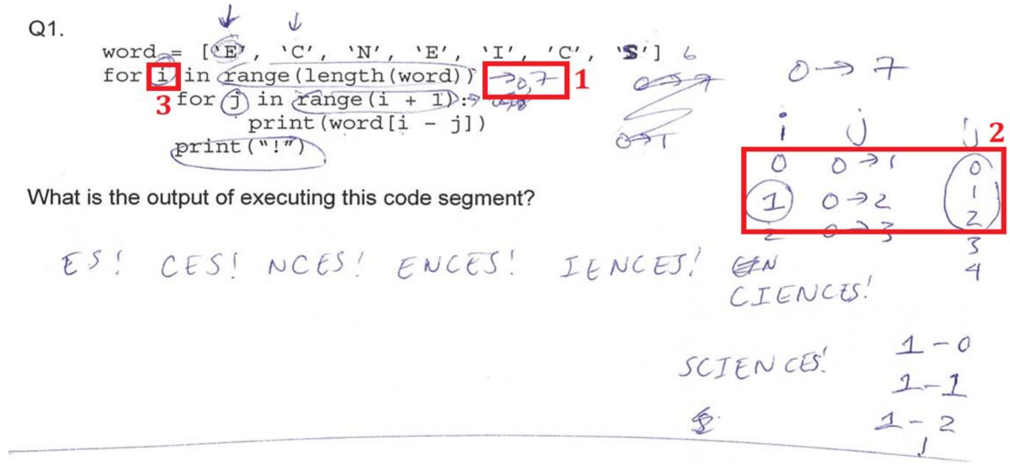


Figure 1: Nelson’s scratch work from when he solved the problem the first time and when he explained the solution.

Based on the video of the interview, we observe that his explain of his strategy towards the problem matches with his actions while solving the problem initially. First, he tapped his pen once on each letter in the array *word* before writing “-> 0, 7” (box 1 in figure 1) on the line of the outer for loop, which represents the range of indices of the array *word.* Secondly, he tapped the pen on the inner for loop, and then at the expression *i-j*, proceeding to write down the first letter. We observed that he repeated this pattern of tapping pens on loop parameters and writing down outputs, respectively.

Nelson described the method of solving the problem in detail while writing down variables and their values to accompany his explanation. He stated that he started the problem initially by looking at the variable *j*, to determine the starting and finish values of the inner loop parameter. After that, he calculated the corresponding index of the array *word* by going through the values of *i* and *j* and subtracting them. Finally, after finishing calculating every corresponding index of the array *word* for one value of *i*, he wrote the exclamation mark. Then, he explained that this process would repeat for every value of *i* and *j*. Therefore, his problem-solving process appears to be consistent with his explanation.

We asked him to confirm whether he was using the same strategy he explained why he was solving the problem. The participant replied:

> **[1]** I think, for me, at least, uhm, I had, uhm, I had this [pointing at box 2 in figure 1] visually in my head for, for the first two [pointing at box 3 in figure 1, likely indicating the value of *i*].
> **[2]** So, like, ES, and then I had CES.
> **[3]** And then, I just figured out the pattern from there.

Interpretation: Nelson’s explanation seems to show that he used problem decomposition to solve this problem. According to his explanation, he first looked at the range of each variable first. He looked at the range of *i* in, and then all ranges of *j* for each value of *i*. After that, he started tracing the values that got print out. As a result of this process, he was able to recognize the patterns of the output. We could conjecture that he was able to see that he separated his method into two parts: finding patterns (transcript line [1] and [2]) and using patterns (transcript line [3]). After he recognized the pattern, he did not do the tracing anymore and could solve the problem, efficiently, based upon the overall picture of the problem he had constructed in the beginning.

4.1.2 Episode with Q2

The participant, Greg, was asked to solve Q2 (see appendix). After he finished, he was asked to explain his strategy or thought process in detail. He started by pointing out that the desired output forced *epsilon* to be the last operation. With this realization, he searched for the operations that could modified *x* and *b*. He said that he chose to explore operation that changes the value of *b* first because he could see that there were less operations that could contribute to changing the value of *b*. He said that, after he realized that *b* could not be modified into anything but 4, he focused on making *x* 11. He, then, observed that he had to perform *beta* to modify *x*, and to do that, he had to make *y* equals 6. He explained the last portion of his strategy while pointing to the scratch work in box 1 of figure 2 below:

> **[1]** And so, then, here, we have 2 options for *y*. We had alpha and gamma. And because there were only two, it's easy to, sort of, plug in and see which one we have to do first.

Then, the participant said that this process took him awhile because he made some arithmetic error. Then, he said:

> **[2]** Eventually, I saw that, if we do alpha first, we get *y* is equal to 3 minus 2. That's equal to 1.
> **[3]** And then, we can do gamma, which is *y* times *a*, which would be equal to, uh, 1 times 6.

The participant reiterates at this point that this was the answer he wanted and that the calculation process ended here because he had used up every available operation. Then, he described how he obtained the final answer:

> **[4]** And then, it was just about remembering the order we would want them in.

It should be noted that, although he explained his thought process after he had already finished the problem, the video of the interview showed that he wrote down calculations for the values of each variable using each operation in exact order that he described. In addition, we observed that he wrote down the final order of operations at the end (box 2 of figure 2), consistent with his explanation about recalling the order of every operation after finishing the calculation. Therefore, we could conjecture that his explanation is an accurate representation of his thought process while he was solving the problem.

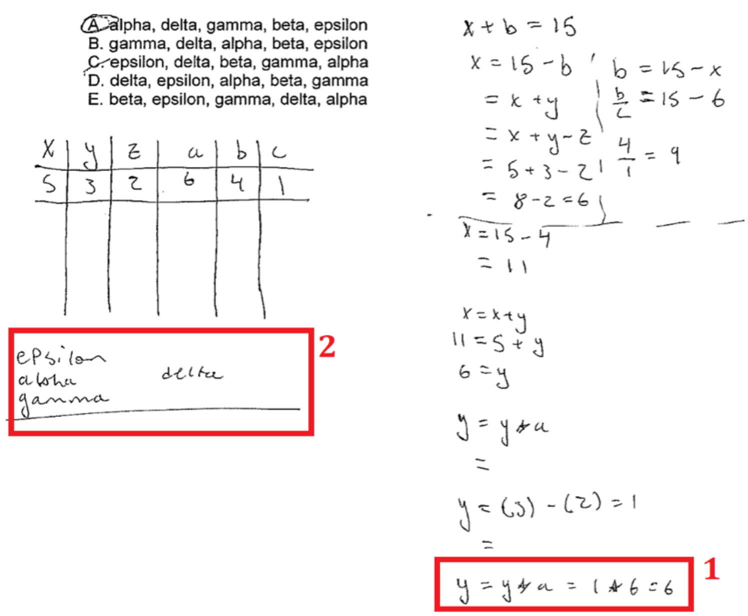


Figure 2: Greg's scratch work from when he solved the problem.

Interpretation: Based on the Greg's explanation of his strategy, we see that he divided the problem into two parts: calculating the output of each operation (transcript line [1] and [2]) and organizing the operations (transcript line [3]). He, then, divided the first problem further, identifying sub-problems, based on previous steps, as well as the method to solve the problem. For instance, calculation of previous operations allowed him to construct a sub-problem of making *y=6*, and that he should solve the problem by plugging in variable values into *alpha* and *gamma* operations in two different orders. Then, he combined the result of that with answers from the previous step, before moving on to constructing the complete order of operations. In this case, Greg's explanation appears to show he utilized problem decomposition to select a way to divide the problem into sub-problems, hierarchically. He determined the order of when he would solve each sub-problem and reconstruct the final solution from the sub-solutions.

4.2 Problem decomposition in spatial reasoning problems

4.2.1 Episode with Q3

Q3 (see appendix) is the first spatial reasoning question and the participant, Chris, is asked to explain his solution and strategy after he circle answer choice A and indicate he reached the solution in 16 seconds.

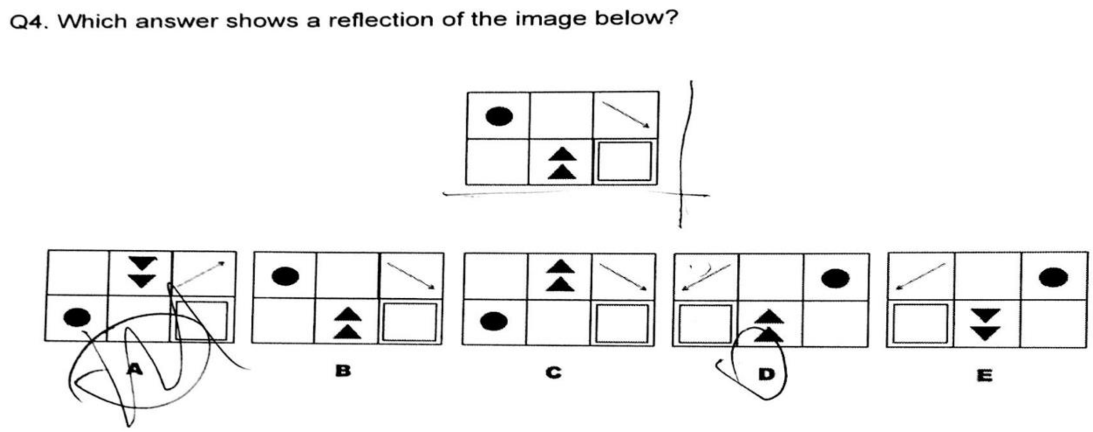


Figure 3: Chris's scratch work from when he solved and explained his strategy for the problem.

As Chris started to explain how he identified the solution, he looked specifically at the orientation of the arrow by pointing his pencil at the arrow and recognizes a mistake in his reasoning towards the solution as he spoke aloud and continue to focus on a small part of the problem by first looking at the orientation of the arrow.

[1] First I picked arrow -*points to arrow with pen*- then I flipped all the other shapes too (pause)
[2] Oh wait I made a mistake. This is wrong. Can I change it?

I told Chris that he can change his answer. After thinking for a while, he proceeded to scratch out A and circle another answer D (the correct answer). He explained how he had started by observing the orientation of the arrow and then drew lines of axis to determine which axis the images are reflected across, as shown in figure above.

Interpretation: We observe that Chris decomposed the problem (indicated by him pointing to the arrow with his pen in the video) and decided to focus that subpart of the image. Then we observe that he further broke down the problem into another small part by identifying the axis of reflection. After looking at the orientation of the arrow, he began to draw out lines that acted as lines of axis to the images. This part helps him identify what the image and the objects should look like depending on the axis it is reflected across.

Based on the Chris's explanation and drawing of his strategy, we see that problem decomposition is key to solving spatial reasoning problems above. Because the problem involves many parts such as the six different tiles with different objects and image reflected across different axis for each answer choice, Chris solves the problem by breaking up the problem into multiple smaller sub-parts, first by observing the orientation of the objects such as the arrow and then figuring out the axis of reflection.

4.2.2 Episode with Q4

This is the second spatial reasoning question and the participant, Marco, is prompted to explain his strategy for this problem. Marco explains his drawing of the three-block tile, as seen in top figure below, and explained that he noticed that most of the tower was constructed of these fundamental 3-block slices (in line 1). He then multiplied 3 by the two different tower lengths and added 1 extra block for each tower since there was a unconnected block, (his arithmetic seen in the top figure below). The first tower had a height of 5 so he multiplied 5 by 3 to get 15 and then added 1 to get 16 for that tower. The second rightmost tower had a height of 3 so they multiplied 3 by 3 to get 9 and then added 1 to get 10. Then he added those two values together 16 + 10 to get the correct solution of 26 blocks.

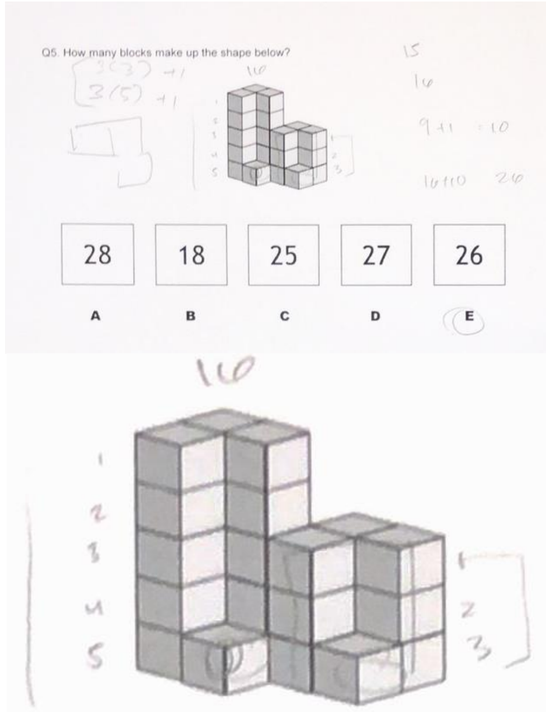


Figure 4: Marco's scratch work from when he solved and explained his strategy for the problem.

Marco explains his strategy below:

> **[1] M:** So these columns are a 3 block thing that repeated all the way down.
> **[2] I:** Like a tower?
> **[3] M:** Yeah so there's 5 of these here (points to left tower) and then 3 of them here (points to right tower) so I just did 3 times 5, 3 time 3,
> **[4] M:** and there's just one extra block at the bottom of it so I added one.
> **[5] M:** Then I added them together.

<u>Interpretation:</u> Instead of counting each block individually or even counting how many single cube towers existed, he broke down the problem by first identifying the fundamental structure to this problem, they divided the structure into 3-cube slice pattern (from line 1). By using this slice and dividing this structure into two "towers", they were able to solve the problem by using it as a reference and found how many of these slices were in the cubes (from line 3) and added the rest (from line 4). Marco decomposed this problem into sub-parts to keep track of the blocks and reached the solution successfully.

5 Conclusion

Based on previous research that documented the existence of a correlation between CS performance and spatial reasoning ability, our study explores the factors of this correlation by focusing on problem-solving schemes for both fields. With our observations of problem-solving strategy towards CS and spatial reasoning problems, we propose that problem decomposition may be a foundation for both solving introductory CS problems and spatial reasoning tasks. When presented with these problems, every participant first observed the problem for its fundamental structures then decomposed the problem into smaller sub-problems to reach the solution.

Unique from previous research, which only suggests possible factors for the correlation between CS and spatial reasoning, our observation and analysis in the previous sections provides supporting evidence for the why CS performance is correlated to spatial reasoning. It should be noted that our evidence and analysis are limited by number of participants, diversity in interview questions, and the depth in which each participant explained his or her strategies. Therefore, our study could serve as a hypothesis for a deeper and broader study of how problem decomposition skill is the bridge between CS and spatial reasoning. Further study into this topic may be very beneficial in improving both teaching and learning processes of CS. Moreover, it may also provide valuable insight on how CS should be taught and introduced. For instance, adding problem decomposition into the curriculum through lectures, problem sets, or games, may be a possible way to improve students' CS problem-solving skills.

Acknowledgement

The authors would like to thank our interviewees. We would like to thank Eric Nguyen and Amy Sorto, who helped us formulate interview questions and provide additional interview data. Finally, we would like to thank Professor Colleen Lewis for supervising us.

References

- [1] 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.
- [2] Cooper, S., Wang, K., Israni, M., & Sorby, S. (2015, July). Spatial skills training in introductory computing. In *Proceedings of the Eleventh Annual International Conference on International Computing Education Research* (pp. 13-20). ACM.
- [3] Engle, R. A., Conant, F. R., & Greeno, J. G. (2007). Progressive refinement of hypotheses in video-supported research. *Video research in the learning sciences*, 239-254
- [4] Grover, S., & Pea, R. (2013). Computational thinking in K–12: A review of the state of the field. *Educational Researcher*, *42*(1), 38-43.
- [5] Jones, S. J., & Burnett, G. E. (2007). Spatial skills and navigation of source code. ACM SIGCSE Bulletin, 39(3), 231-235.
- [6] More, N. O(2000). *Handbook of Known Solutions to the Traveling Salesman Problem*, Amarillo, TX: Big House Publishing.
- [7] Parker, Miranda C., Guzdial Mark, & Engleman, Shelly .2016. Replication,validation,and use of a language independent CS1 knowledge assessment. *Proceedings of the 2016 ACM conference on international computing education research*. ACM, 93–101
- [8] Parkinson, J., & Cutts, Q. (2018). Investigating the Relationship Between Spatial Skills and Computer Science. *Proceedings of the 2018 ACM Conference on International Computing Education Research - ICER 18*, 104-114.
- [9] Tai, D. W., Yu, C. H., Lai, L. C., & Lin, S. J. (2003). A study on the effects of spatial ability in promoting the logical thinking abilities of students with regard to programming language. *World Transactions on Engineering and Technology Education UICEE*, *2*(2), 251-26.
- [10] Wai, J., Lubinski, D., & Benbow, C. P. (2009). Spatial ability for STEM domains: Aligning over 50 years of cumulative psychological knowledge solidifies its importance. *Journal of Educational Psychology*, *101*(4), 817.
- [11] [n. d.]. https://www.wikijob.co.uk/content/aptitude-tests/test-types/ spatial-awareness-test

APPENDIX

Question 1

```
word = ['E', 'C', 'N', 'E', 'I', 'C', 'S']
for i in range(length(word)):
    for j in range(i + 1):
        print(word[i - j])
    print("!")
```

]What is the output of executing this code segment?

Question 2

Consider the following code segment:

```
x = 5
y = 3
z = 2
```

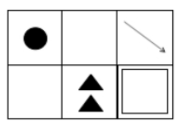
```
a = 6
b = 4
c = 1
# insert here operations presented below
return a
```

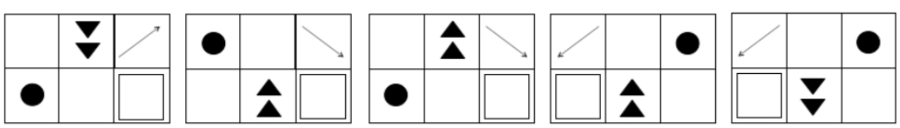
In what order will you need to perform the following operations such that the value of a is 15 at the end of the series of operations?

```
alpha: y = y - z
beta: x = x + y
gamma: y = y * a
delta: b = b / c
epsilon: a = x + b
```

A. alpha, delta, gamma, beta, epsilon
B. gamma, delta, alpha, beta, epsilon
C. epsilon, delta, beta, gamma, alpha
D. delta, epsilon, alpha, beta, gamma
E. beta, epsilon, gamma, delta, alpha

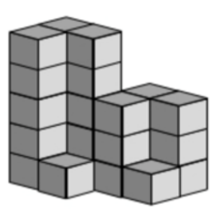
<u>Question 3</u>
Which answer shows a reflection of the image below?





A B C D E

<u>Question 4</u>
How many blocks make up the shape below?



A	B	C	D	E
28	18	25	27	26