Talent Search
Bebras Computational Thinking Challenge
(Pen-and-Paper Version)

www.olympiad.org.za

2016
The Talent Search Computational Challenge could not have been run this year without the generous support of the following sponsors:
  Standard Bank
  The South African Computer Olympiad Trust
  The Institute for IT Professionals South Africa

Finally it is important to recognise all the National Coordinators and members of the International Bebras community who have been immensely generous in their support and help in getting this competition off the ground as well as of course developing the challenges.

Working Group providing this manual

Working Group for UK Bebras Computational Challenge 2015:
  Daphne Blokhuis
  Peter Millican
  Chris Roffey
  Eljakim Schrijvers
  Sue Sentance
Introduction

The Talent Search Computational Challenge is an online competition open to students in the SA and English speaking Schools in SADC countries. It requires intelligence, but no previous knowledge. It is hoped it will raise general interest in Computer Science and young people to understand that Computational Thinking has wide application in solving all sorts of problems that might be met in life. The philosophy emphasises participation whilst celebrating achievement.

This competition is anything but unique to SA. This is the International Bebras Competiton’s 12th year and there are over 30 countries participating.

On the following pages you will find the tasks used in the Talent Search Challenge. Above each question is noted which age groups and at what level the problems were used.

After each question there is an answer, an explanation of how the answer could be obtained plus a section on how the tasks are related to Computational Thinking. We have also mapped the tasks to the Computational Thinking Concepts that feature in the Progression Pathways Assessment Framework created by Mark Dorling. It is our sincere hope that the Computational Thinking information provided will enhance the usefulness of this booklet for Computer Science teachers and their students whether in the Primary or Secondary phase. It is for this reason that this booklet is being distributed to as many schools as possible and made freely available as a PDF that can be further distributed freely.

The 2016 competition was conducted in four age groups:

<table>
<thead>
<tr>
<th>Age Group</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Elite</td>
<td>Grade 12, post matrics &amp; students (individuals)</td>
</tr>
<tr>
<td>Seniors</td>
<td>Grade 10 and 11 (individual learners)</td>
</tr>
<tr>
<td>Intermediate</td>
<td>Grade 8 and 9 (individual learners)</td>
</tr>
<tr>
<td>Juniors</td>
<td>Grade 6 and 7 (individual learners)</td>
</tr>
<tr>
<td>Teams</td>
<td>Grade 4 to 7 – for teams of up to four.</td>
</tr>
</tbody>
</table>
Introduction

The contest is completed online in schools under the supervision of teachers.

For Elite, Higher, Intermediate, Juniors and teams there are 18 multiple-choice questions to be attempted in 45 minutes.

The organisers wish to pass on a special thanks to all the teachers who have made it possible for their learners to enter into this competition by taking on a huge amount of the necessary administration for us. We invite schools to participate again in March 2017.

Keep informed by visiting www.olympiad.org.za
<table>
<thead>
<tr>
<th>Junior</th>
<th>Interm</th>
<th>Senior</th>
<th>Elite</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Animal Competition</td>
<td>C1</td>
<td></td>
<td></td>
<td>16</td>
</tr>
<tr>
<td>Animation</td>
<td>B2</td>
<td>A5</td>
<td></td>
<td>15</td>
</tr>
<tr>
<td>Beaver Dam</td>
<td>C1</td>
<td></td>
<td></td>
<td>13</td>
</tr>
<tr>
<td>Beaver Logs</td>
<td></td>
<td></td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Beaver Lunch</td>
<td>C3</td>
<td>C3</td>
<td></td>
<td>30</td>
</tr>
<tr>
<td>Beaver the Alchemist</td>
<td>C5</td>
<td>B4</td>
<td></td>
<td>45</td>
</tr>
<tr>
<td>Beaver Tutorials</td>
<td>A5</td>
<td></td>
<td></td>
<td>53</td>
</tr>
<tr>
<td>Biber-Hotel</td>
<td>B4</td>
<td>A4</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Bowl Factory</td>
<td></td>
<td>C2</td>
<td></td>
<td>37</td>
</tr>
<tr>
<td>Bracelet</td>
<td>A1</td>
<td>A1</td>
<td></td>
<td>11</td>
</tr>
<tr>
<td>Building a Chip</td>
<td></td>
<td>C5</td>
<td></td>
<td>54</td>
</tr>
<tr>
<td>Chakhokhbili</td>
<td>C5</td>
<td></td>
<td></td>
<td>32</td>
</tr>
<tr>
<td>Crane Operating</td>
<td>A2</td>
<td></td>
<td></td>
<td>7</td>
</tr>
<tr>
<td>Cross Country</td>
<td>C4</td>
<td>B5</td>
<td></td>
<td>19</td>
</tr>
<tr>
<td>Drawing Patterns</td>
<td>C2</td>
<td>B2</td>
<td></td>
<td>24</td>
</tr>
<tr>
<td>Drawing Stars</td>
<td>B1</td>
<td>A4</td>
<td>A3</td>
<td>29</td>
</tr>
<tr>
<td>Dream Dress</td>
<td>A3</td>
<td>A2</td>
<td></td>
<td>10</td>
</tr>
<tr>
<td>E-mail Scam</td>
<td></td>
<td>A5</td>
<td>A4</td>
<td>39</td>
</tr>
<tr>
<td>Fair share</td>
<td>C5</td>
<td>C2</td>
<td>B3</td>
<td>21</td>
</tr>
<tr>
<td>Fireworks</td>
<td></td>
<td>C2</td>
<td></td>
<td>49</td>
</tr>
<tr>
<td>Geocaching</td>
<td>B3</td>
<td>A3</td>
<td></td>
<td>17</td>
</tr>
<tr>
<td>Irrigation System</td>
<td></td>
<td>A1</td>
<td></td>
<td>40</td>
</tr>
<tr>
<td>Kangaroo</td>
<td></td>
<td>B5</td>
<td></td>
<td>57</td>
</tr>
<tr>
<td>Mobiles</td>
<td>C4</td>
<td></td>
<td></td>
<td>51</td>
</tr>
<tr>
<td>Mushrooms</td>
<td>A4</td>
<td></td>
<td></td>
<td>8</td>
</tr>
<tr>
<td>Pirate Hunters</td>
<td></td>
<td>C1</td>
<td></td>
<td>43</td>
</tr>
<tr>
<td>Popularity</td>
<td>B4</td>
<td>B3</td>
<td></td>
<td>46</td>
</tr>
<tr>
<td>Quick Beaver Code</td>
<td></td>
<td>C3</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Setting the Table</td>
<td>A5</td>
<td></td>
<td></td>
<td>9</td>
</tr>
<tr>
<td>Spies</td>
<td></td>
<td>C4</td>
<td></td>
<td>59</td>
</tr>
<tr>
<td>Stack Computer</td>
<td>C1</td>
<td>B2</td>
<td></td>
<td>26</td>
</tr>
<tr>
<td>Super Power Family</td>
<td>A3</td>
<td>B1</td>
<td></td>
<td>41</td>
</tr>
<tr>
<td>Theatre</td>
<td>C4</td>
<td>B5</td>
<td></td>
<td>35</td>
</tr>
<tr>
<td>Throw the Dice</td>
<td>C3</td>
<td>B4</td>
<td>B1</td>
<td>28</td>
</tr>
<tr>
<td>Trains</td>
<td>B5</td>
<td></td>
<td></td>
<td>12</td>
</tr>
<tr>
<td>Walnut Animals</td>
<td>B1</td>
<td></td>
<td></td>
<td>22</td>
</tr>
<tr>
<td>Word Chains</td>
<td>B2</td>
<td>A1</td>
<td></td>
<td>48</td>
</tr>
<tr>
<td>You Won't Find It</td>
<td>B3</td>
<td>A2</td>
<td>A2</td>
<td>34</td>
</tr>
</tbody>
</table>
Each problem in this booklet has a flag indicating the country of origin. However, many people were involved in the further editing, translating and providing additional material. The South African team is indebted to the generosity of spirit and community of Computer Scientists around the world!
The crane in the port of Lodgedam responds to six different input commands:

1. left
2. right
3. up
4. down
5. grab
6. release

Crate A is in the left position, crate B is in the position on the right.

**Question:**

Which is the correct set of instructions to swap the position of the two crates. Write down the letter of the correct answer in the appropriate block on your answer sheet.

A. (Down, Grab, Up, Right, Down, Release, Up)
B. (Down, Grab, Up, Right, Down, Release, Up) (Right, Down, Grab, Up, Left, Left, Down, Release, Up) (Right, Down, Grab, Up, Right, Down, Release)
C. (Right, Right, Down, Grab, Up) (Left, Left, Down, Release, Up)
D. (Down, Grab, Up, Right, Right, Down, Release, Up) (Down, Grab, Left, Down, Release, Up) (Down, Grab, Up, Right, Down, Release, Up)

**Answer:**

B. Down, Grab, Up, Right, Down, Release, Up, Right, Down, Grab, Up, Left, Left, Down, Release, Up, Right, Down, Grab, Up, Right, Down, Release

**Explanation:**

When operating machines and computers, you must put the commands in the correct order. Online this is an interactive problem so most students will get this correct. The question is, how much time of the 45 minutes allowed does it take up.

**It's Computational Thinking:**

*Skills - Algorithmic Thinking (AL)*

In this problem a sequence of instructions is searched for. A set of instructions is called an algorithm. In this problem two objects can only be changed if one of the objects is placed on an empty place.

Most computers still work with sequentially run programs, so each exchange operation in the memory of the computer also needs an extra space.
Three beavers, Sad, Tail and Talky, are standing in a forest.

Each wants to go where there are mushrooms. Arrows in the picture to the right show the directions the beavers will walk.

**Where do the beavers end up?**
Indicate the square by a letter and number. Eg the Talky beaver is at C3 at the start.

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Write the reference of the square where the beavers end up:

Eg. Sad B2, Tail D4, Talky E5. You need to have all three correct to get marks. There are no part marks.

**Answer and Explanation:**

Sad E2, Tail C5, Talky A4 (in any order)

**It’s Computational Thinking:**

*Skills - Abstraction (AB), Algorithmic Thinking (AL)*

Simple sets of instructions called algorithms can help us solve problems. It is sometimes easier to do this with pictures and arrows than with words.
Setting the Table

Beaver Bob has set the breakfast-table as shown in the picture.

Question:
In which order has he placed the objects on the table?

A. table cloth, napkin, cup and saucer, knife, plate
B. table cloth, napkin, cup and saucer, plate, knife
C. napkin, knife, table cloth, cup and saucer, plate
D. table cloth, cup and saucer, napkin, plate, knife

Answer:
Choice D

Explanation:
The table cover was placed first because all the other things are on it. The next was the cup because the napkin is on the cup. The plate is on the napkin and the knife is on the plate.

It’s Computational Thinking:
Skills - Algorithmic Thinking (AL), Evaluation (EV)

Sequences are important in computational thinking.
When you give instructions to a computer, those will be implemented in the order in which they are given.
Layers (sequence of pictures) are important when making graphics programs. In a picture you can separate different elements and change their sequence.
Kate wants to buy her dream dress.
It must have:

- short sleeves,
- more than 3 buttons,
- stars on its sleeves.

Four shops sell only the dresses shown.

**Question:**
Which of these shops sells Kate's dream dress?

BeaverYoker, BeaverNova, B&B or TomTeaver

**Answer:**
The correct answer is B&B

**Explanation:**
To solve this task, we must simultaneously satisfy three requirements. This can be done by ruling out dresses that do not meet any one of the requirements. After doing this, one can see that the dress on the bottom left sold by B&B is Kate's dream dress.

**It's Computational Thinking:**

*Skills - Evaluation (EV)*

The task involves statements (conditions/requirements) that must be evaluated (determined to be true or false) for a set of objects (coats). Conditions and their evaluation is an important part of programming and algorithmic thinking.

Conditions can be simple statements. However, more complex statements can be formed using logical operators such as AND, OR, NOT, ... This task uses the AND operator.
Emily has broken her favourite bracelet. The broken bracelet now looks like this:

Question:
Which of the following four bracelets shows what the bracelet looked like when it was whole?

Answer:
The correct answer is C

Explanation:
Bracelet C follows the beads in the same order as the broken bracelets.

It’s Computational Thinking:
Skills - Evaluation (EV)

In computational thinking it is important to be able to recognise patterns which may be useful to us. Recognising patterns helps us find similarities in things that may look different at first, but have something in common.

This task also deals with verifying a proposed solution: the possible answers need to be checked against the original bracelet to see if they meet the required order of the shapes. The same process of verifying a solution is important in computing in order to determine if the output of a program is correct.
The train lines in Beaver City all have their own number. 
Unfortunately the numbers are only shown on this map. 
When you are on a train you cannot see the line number anywhere! 
You get onto a train at the main station where all the lines begin. 

After three stations your train makes a turn. 
At the next station it makes another turn. 
Four stations later you have arrived at your destination and the train goes no further. 

**Question:**
Which train line were you on?

**Answer:**
Line 6

**Explanation:**
This problem is solved by following the instructions carefully and ruling out routes that do not allow the train to complete its journey. So after three stations a turn needs to be made ruling out line 1 but the other lines still need to be checked.

**It's Computational Thinking:**
*Skills - Algorithmic Thinking (AL), Evaluation (EV)*

It is quite common when writing programs to find that your code produces unexpected results. Careful stepping through instructions is the first part of identifying where the problem is.
The Birchtree family needs to water their fields. Only fields with flowers need to be watered. The other fields must remain dry.

If a sluice gate is open, water will flow from the lake in the middle to the fields below.

**Question:**

Find a solution where water only flows to the fields with flowers in.

Help the Birchtree family produce a plan to open the correct sluice gates and leave the others closed so that water **only** flows to the fields with flowers in them.

Which gates need to be open? Write down the letter of the correct solution in the appropriate block on your answer sheet.

A: Open 1, 2, 3, 4
B: Open 4, 5, 6, 8
C: Open 1, 2, 3, 7
D: Open 1, 2, 6, 7, 9
Answer: C

It’s Computational Thinking:

*Skills - Abstraction (AB), Evaluation (EV)*

The main difficulties here are the following:

- Some fields are watered by more than one canal
- Some canals water more than one field
- Our plan must be exact: you cannot water too many fields or too few

Although this particular problem is maybe not so hard to solve, if you have more fields, more canals and more connections, the task may become extremely difficult - even for a computer!

Knowing (or computing) how hard a task is even before you try to solve it is a very important part of computer science.
Taro is planning an animation of a face that is made from a sequence of pictures. To make the animation run smoothly, only one feature of the face should change from one picture to the next.

Unfortunately, the pictures got mixed up. Now Taro must find the correct order again. Luckily, he knows which picture is last.

**Question:**
Put the pictures in the correct order by dragging them onto the squares.

**Answer:**

**Explanation:**
The ears change from large to small. The whiskers change from curly to straight.
The nose changes from small to large. The mouth changes from plain to smile.
The number of teeth changes from 3 to 2. It is best to work backwards to solve this!

**It’s Computational Thinking:**

*Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)*

In order to find the differences between the pictures, you have to find about about the essential attributes of the faces first. The list of attributes and their possible values is:

- ears: small, large
- mouth: plain, smile
- nose: small, large
- number of teeth: 2, 3
- whiskers: curly, straight

Face A can now be described as:

(ears: small; mouth: plain; nose: large; number of teeth: 3; whiskers: straight)

In computing, it is very usual to model things from the real world as “objects” that have attributes and values.
The beavers and dogs had a competition. In total nine animals took part. The nine participants had the following scores: 1, 2, 2, 3, 4, 5, 5, 6, 7.
No dog scored more than any beaver.
One dog tied with a beaver.
There were also two other dogs that tied with each other.

**Question:**
How many dogs took part in the competition?

**Answer:**
6

**Explanation:**
First we arrange the scores in numerical order. Then we look for ties, there are two of these. One must be between two dogs and the other between a dog and beaver. If the two animals that score 2 points are the beaver and dog then the two animals scoring 5 points must be the tied dogs. This cannot be the case though because that would mean two dogs at least scored more points than a beaver. We can now see the boundary between beavers and dogs:
- dogs 1, 2, 3, 4, 5, 5, 6, 7
- beavers Therefore 6 dogs took part in the competition

**It’s Computational Thinking:**
Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)

When working with data, some organisation of them is necessary. This task requires us to understand how the data are ordered and how ordering rules are used. You also need to use some logic to solve this task.

Remember that no mathematics is necessary to solve this task. You can solve it even if there were letters instead of numbers: A, B, C, D, E, E, F, G.
Two friends, Anna and Bob, are searching for treasure. They have a smartphone app that shows them the direction to the treasure they are looking for. The two boxes on the map show where the treasure is. Anna is searching for box 1. Bob is looking for box 2.

Anna and Bob are standing in the same place. The picture shows the map and a screenshot of the smartphones.

**Question:**
Where are Anna and Bob standing?

**Answer:** B
Answer and Explanation:

It’s Computational Thinking:

*Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV)*

Geocaching is a nice game which uses the global navigation system of the Earth. There is a lot of applications of GPS: car navigation, rescue of lost people, traffic control, localisation of stolen cars and bikes.

In this task, you have to discover the role of the arrow in both devices. So you have to understand the description of a situation and to project it into a graph (or map).
Three very fast beavers will compete in a cross-country run.

Mr. Brown will overtake one beaver when running uphill.

Mrs. Pink will overtake one beaver when running downhill.

Mrs. Green will overtake one beaver when running across rocks.

The course is shown in the picture: uphill, followed by some rocks, downhill and then some more rocks.

Mrs. Pink starts in the first position, followed by Mr. Brown and then Mrs. Green.

**Question:**

In which order will the beavers finish the race?

A. Mrs Pink, Mr Brown, Mrs Green
B. Mr Brown, Mrs Pink, Mrs Green
C. Mr Brown, Mrs Green, Mrs Pink
D. Mrs Green, Mrs Pink, Mr Brown
Answer:
(C) Mr. Brown, Mrs. Green, Mrs. Pink

Explanation:

It’s Computational Thinking:

*Skills - Algorithmic Thinking (AL), Evaluation (EV)*

Programmers must often look closely at how their programs run. This is especially true when the programs do not work well: in this case, programmers carefully go through and check the effect of each line of the program.

This task is similar. You are given some data – the sequence of runners. You have to “step through the program” - where the “steps” are uphill – rocks – downhill – rocks. You have to observe the effect of each step on the sequence and thus discover the “output” of the program, that is, the order at the end.
Hamid has a 4 litre beaker full of a hazardous chemical.
Kazim has an empty 3 litre beaker and another empty 1 litre beaker.

Hamid and Kazim want to share the chemical between them equally and use a machine to do this safely.

The machine can pour one beaker into another. It stops pouring when a beaker is completely emptied or filled, whichever happens first.

Question:

Find the sequence of pours that produces equal shares of the chemical for Hamid and Kazim. Pouring from the 4 litre beaker to the 3 litre beaker is shown as 4→3 etc.

Your sequence must use the minimum number of pours possible. Write down the letter of the correct pour in the appropriate block on your answer sheet.

A. 4→1, 4→3, 1→4, 3→1, 1→4
B. 4→1, 1→3, 4→1, 1→3
C. 4→1, 4→3, 1→4,
D. 4→3, 3→1, 1→4

Answer: D

It’s Computational Thinking:

Skills - Algorithmic Thinking (AL), Evaluation (EV)

This task involves a classical solution search. It is a relatively easy example of a classic puzzle. There are a number of computational thinking strategies that could be employed: For a simple puzzle like this a brute force strategy works (try everything systematically until the correct answer is achieved.) This is a common technique used by computers which takes advantage of their speed. With very complex problems though other techniques might be better employed such as backtracking, breadth-first search, etc.
Gerald was playing in the woods. He used nuts and sticks to create four nice animals.

<table>
<thead>
<tr>
<th>Starfish</th>
<th>Dog</th>
<th>Sea lion</th>
<th>Giraffe</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image1" alt="Starfish" /></td>
<td><img src="image2" alt="Dog" /></td>
<td><img src="image3" alt="Sea lion" /></td>
<td><img src="image4" alt="Giraffe" /></td>
</tr>
</tbody>
</table>

His sister managed to bend the animals around without removing any of the sticks.

Gerald was very upset because he really loved the figure of a dog.

**Question:**

Which of the following figures can be bent back to make the figure of the dog again? Write down the letter of the answer in the appropriate block on your answer sheet.

<table>
<thead>
<tr>
<th>C</th>
<th>D</th>
<th>B</th>
<th>A</th>
</tr>
</thead>
</table>
Answer: A

Explanation:
Each animal can be described by the connections between its parts. The specific positions of parts and angles of connections may change while being played with, but that does not change the animal itself. So we need to determine pairs of pictures with the same structure on them. Let's start with the starfish. It has a regular structure and is therefore easiest to spot: one central part and five arms. There is only one possibility among transformed animals.

It's Computational Thinking:
*Skills - Abstraction (AB), Algorithmic Thinking (AL), Generalisation (GE)*

With walnut animals, we abstract from features like fur and size. We represent the animal only by the structure of its body, the rest is unimportant. This structure is preserved even when the animals are transformed. A computer scientist must recognise what is important, what can be left out, and how structures are similar.

If the animals in our task were just a bit bigger and more complicated, it would be extremely difficult to identify them. That is why computer scientists try to invent efficient algorithms to solve such problems.
Your job is to select the sequence of instructions that draws the image shown at the above.

Write the letter of the correct answer in the appropriate block on your answer sheet.

**Answer:**  C

**It's Computational Thinking:**

*Concepts - Algorithmic Thinking (AL)*

This problem asks us to write a program, including a sequence of instructions within a simple loop.
The beavers want to encode numbers. They developed the Quick-Beaver-Code (QB-Code).

This is a code consisting of squares. Every square has a certain value. The squares are filled line by line from the bottom to the top and from RIGHT to LEFT.

The value of the bottom right square is 1. The other squares have double the value of the square before them.

**Example:**

Here is a 3x3 QB-Code. The beavers have encoded a number by darkening some squares.

The number encoded is the sum of the values of the dark squares, so the number encoded in this QB-Code is $2 + 32 + 64 = 98$.

**Question:**

Of the following 4x4 QB-Codes, which one encodes the highest number? Write down the letter of the answer in the appropriate block on your answer sheet.

**Answer:**

Answer C

**Explanation:**

Did you notice that you could get the answer without doing any complicated calculation? The square in the top-left corner has the highest value (256). Note that the sum of all the other squares, i.e. $128+64+32+16+8+4+2+1 = 255$ which is less than 256. Hence, the highest possible number encoded is the QB-Code with the top-left corner darkened.

**It's Computational Thinking:**

*Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV)*

QR codes are used in a wide range of applications. This includes commercial tracking, entertainment and transport ticketing, product/loyalty marketing and in-store product labelling. Users of mobile phones may receive text, add a vCard contact to their device, open a URL, or compose an e-mail or text message after scanning QR codes.
The Stack Computer is loaded with boxes from a conveyor belt. The boxes are marked with a Number or an Operator (+, -, * or /).

The computer is loaded until the top box is a box marked with an operator. This operator is then used on the two boxes below it. The three boxes are then fused into one single box and marked with the outcome of the calculation.

In the Stack Computer, calculations are entered in a different way to a normal calculator.

**Examples:**

- $2+3$ must be entered as $2 \ 3 +$
- $10-2$ must be entered as $10 \ 2 -$ 
- $5*2+3$ must be entered as $5 \ 2 * 3 +$
- $5+2*3$ must be entered as $5 \ 2 3 * +$
- $(8-2)*(3+4)$ must be entered as $8 \ 2 - 3 4 *+$

**Question:**

How should the following computation be entered: $4*(8+3)-2$? Enter your answer in the appropriate block on your answer sheet.
Answer:
The correct answer is (A) $8 \ 3 + 4 \ * \ 2$ - 
However, the following answers would also have been acceptable as they all produce the correct output.

\[
\begin{align*}
4 \ 3 \ 8 + \ * \ 2 - \\
4 \ 8 \ 3 + \ * \ 2 - \ \text{(is also a very sensible answer)}
3 \ 8 + 4 \ * \ 2 - 
\end{align*}
\]

These inputs all lead to the same result, even though the order of the operators and operations are not the same as intended in the given expression.
However they were not given as options in the pen-and-paper version.

Explanation:
From left to right, we first have $4*(8+3)$, so we need 4 and the result of 8+3 on the stack. We achieve that by writing

\[
4 \ 8 \ 3 +
\]

We then have 4 and 11 on the stack, so we add a * to multiply the two numbers. We now have 4 4 on the stack, we add a 2 and a – for the final subtraction.

It’s Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

The usual notation for arithmetic expressions is not the easiest to understand for a computer, or rather, it takes a more complicated program (algorithm) to process such expressions. However writing a program to analyse expressions in postfix notation (as done by this machine) is much much easier. This is one reason why on some early handheld calculators this notation had to be used. Another reason is, that the postfix notation does not require any brackets, no matter how complex the expression.
After school the young beavers often play together.
To avoid quarrels about where to play, they throw a normal six sided die.
The decision is found according to this rule:

```
1 IF  the first throw is greater than the second throw
2 THEN  we go to play in the woods
3 ELSE
4 IF  the third throw is less than the first throw
5 THEN  we go to play at the river
6 ELSE  we go to play on the sports field
```

**Question:**
Which sequence of throws will send the young beavers to the sports field? Write down the letter of the answer in the appropriate block on your answer sheet.

A  
B  
C  
D  

**Answer:** B

**Explanation:**
The first throw of 3 is not greater than the second throw of 3, so the ELSE-IF branch decides. The third throw of 3 is not greater than the first 3 thrown so the rule sends the young beavers to the sporting field.

**It’s Computational Thinking:**

*Skills - Algorithmic Thinking (AL), Evaluation (EV)*

"IF-THEN-ELSE" is a form of selection widely used in programming languages. It decides, depending on the current situation, about a program's next action. The "IF-THEN-ELSE" can suggest that being able to make a dual decision is the standard case in life. This tempts beginners programmers to use far too simple world models in their apps. Using nested "IF-THEN-ELSE" constructs or "CASE" (or in Python IF-ELIF-ELIF- …) constructs enables more complex scenarios to be handled.
Stella loves to draw stars. She has devised a system for labelling her stars according to their shape. She uses two numbers:

- A number of points of the star.
- A number indicating if a line from a point is drawn to the nearest point (the number is 1), the second closest point (the number is 2), etc.

Here are four examples of Stella's labelling system:

Question:

How would Stella label the following star? Write down the letter of the correct answer in the appropriate block on your answer sheet.

A 9:3  B 9:4  C 10:4 or D 10:5

Answer:

C 10:4

Explanation:

Computers need simple representations of objects to be able to work with them. The fact that a complex and beautiful object such as a regular star polygon can be described by only two integers is an example of a simple representation.

It’s Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)
Hm, what to take for lunch today?
The cafeteria gives instructions on how to choose a good lunch.

This is shown as a diagram:
Below the tray you see different types of food containers.
The numbers above the containers indicate how many containers of this type can be added to a tray.
Each container can only have food items put in it that are shown below it.
The numbers above the food items indicate how many food items of this type can be added to the containers.

**Question:**
Which of the following lunches is **NOT** a proper lunch? Write down the letter of the correct solution in the appropriate block on your answer sheet.
Answer:
D is not a Beaver Lunch.

Explanation:
D is not a beaver lunch, because there is no container of the third type on the tray. In the picture there are the numbers 1 and 2 written above the image of this container indicating that a beaver meal should have one or two of these.

It’s Computational Thinking:

Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)

The image defining the recommendation is an example of a certain type of diagram, called a tree. It looks a little bit like a tree upside down with the root on top. Programmers use diagrams like these to define the structure of aggregates. An aggregate is a complex object which consists of simpler parts. And each part may be composed of even simpler parts.
Sergo loves to cook. His favourite meal is Chakhokhbili.

When cooking in the garden he uses a single gas burner. He performs the following actions after each other:

1. Cook an onion 10 minutes
2. Cook a bell pepper 10 minutes
3. Combine the cooked onion and cooked bell pepper, add a tomato and cook this together 20 minutes
4. Cook a chicken 30 minutes
5. Combine everything from steps 3 and 4, add some spices, and cook it all. 20 minutes

In total Sergo needs 90 minutes to prepare his Chakhokhbili on a single gas burner.

**Question:**

When Sergo cooks at home he has many gas burners available. He is able to use more burners so his meal is ready sooner.

Which of the following statements is **NOT** correct?

A. Sergo can reduce the cooking time by 10 minutes when using 2 burners
B. Sergo can reduce the cooking time by 30 minutes when using 2 burners
C. Sergo can reduce the cooking time by 40 minutes when using 3 burners
D. Sergo can reduce the cooking time by 50 minutes when using 4 burners

Write down the letter of the **NOT** correct statement in the appropriate block on your answer sheet.
Answer:

D. Sergio can reduce the cooking time by 50 minutes when using 4 burners - False

Explanation:

The left picture illustrates how to decrease the cooking time by 30 minutes (for answers A and B)

The right picture illustrates how to decrease the cooking time by 40 minutes (for answer C)

It’s Computational Thinking:

Skills - Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

In this task gas burners are computer resources such as processors. If you have only one resource, you should process your work sequentially and if you have more resources you can sometimes process the tasks in parallel.

Shortening the time is like structuring programming code to run as fast as possible given the amount of processors available. It is good practice to optimise code to be able to run as fast as possible. Parallel computing is a big field of research in computer science.
Alex and Betty send each other messages using the following sequence of transformations on every word.

For example, the word "BEAVER" is transformed to "WBFCSF".

Betty receives the encoded message "PMGEP" from Alex.

**Question:**

What did Alex want to say? Write down the word in the appropriate block on your answer sheet.

A. LODGE
B. FLOOD
C. RIVER
D. KNOCK

**Answer:**

B. FLOOD

**Explanation:**

The steps of the transformation, applied in the reverse order, are:

"PMGEP" → "OLFDO" → "DOOLF" → "FLOOD" That is:

- Replace each letter with the previous in the alphabet;
- Shift letters by 2 to the right;
- Reverse word.

The other answers are not correct.

**It’s Computational Thinking:**

*Skills - Algorithmic Thinking (AL), Evaluation (EV)*

The image in this task is part of a simple flowchart, that explains how to change a word step by step. Flowcharts are a way to describe algorithms. In this task the algorithm changes text so that nobody can understand it. This is called ciphering.
Three spotlights are used to light the theatre stage in the beavers' forest, a red one, a green one and a blue one.

The colour of the stage depends on which of the three spotlights are turned on.

This table shows the possible combinations of colours.

<table>
<thead>
<tr>
<th>Red light</th>
<th>Green light</th>
<th>Blue light</th>
<th>Stage colour</th>
</tr>
</thead>
<tbody>
<tr>
<td>off</td>
<td>off</td>
<td>off</td>
<td>Black</td>
</tr>
<tr>
<td>off</td>
<td>off</td>
<td>on</td>
<td>Blue</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>off</td>
<td>Green</td>
</tr>
<tr>
<td>off</td>
<td>on</td>
<td>on</td>
<td>Cyan</td>
</tr>
<tr>
<td>on</td>
<td>off</td>
<td>off</td>
<td>Red</td>
</tr>
<tr>
<td>on</td>
<td>off</td>
<td>on</td>
<td>Magenta</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>off</td>
<td>Yellow</td>
</tr>
<tr>
<td>on</td>
<td>on</td>
<td>on</td>
<td>White</td>
</tr>
</tbody>
</table>

From the beginning of the show, the lights will be switched on and off in this pattern:

- The red light repeats the sequence: two minutes off, two minutes on.
- The green light repeats the sequence: one minute off, one minute on.
- The blue light repeats the sequence: four minutes on, four minutes off.

**Question:**

What will the colour of the stage be in the first 4 minutes of the show?
Write down the minute and the colour.


You need all the answers correct to get marks. No part marks are allocated.
**Theatre**

**Answer:**

<table>
<thead>
<tr>
<th></th>
<th>Minute 1</th>
<th>Minute 2</th>
<th>Minute 3</th>
<th>Minute 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Blue</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cyan</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Magenta</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>White</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Explanation:**
The best way to approach this is to produce a table showing which lights were on when and what the result would be:

<table>
<thead>
<tr>
<th>Time (mins):</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Red light</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Green light</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Blue light</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Stage colour:</td>
<td>Blue</td>
<td>Cyan</td>
<td>Magenta</td>
<td>White</td>
</tr>
</tbody>
</table>

**It's Computational Thinking:**

*Skills - Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)*

The way colours can be constructed from three basic colours of lights is the RGB colour model. It is used, among many other places, in computer monitors and TV sets to show pictures. Pictures are made up of pixels, which are coloured as a combination of the three basic colours. It is an important principle of computer hardware and computer graphics.

The task also involves an understanding of sequences of actions – in this case the sequence of switching particular light(s) on and off. It is an important part of algorithmic thinking. This is also an example of parallel execution of three independent sequences.
A factory produces sets of 6 bowls of different sizes. A long conveyor belt moves the bowls one by one, from left to right. Bowl production places the 6 bowls of each set onto the conveyor belt in a random order. Before packing the bowls, they need to be sorted to look like this:

To help with the sorting, the factory places workers along the conveyor belt. When a set of bowls passes a worker, the beaver will swap any two neighbouring bowls which are in the wrong order. The worker will keep doing this until the set of 6 bowls has finished passing.

See how the order of a set of bowls changes as it passes one worker:

Three workers are therefore required to put this set in proper order.

**Question:**
How many workers should be put along the line to sort the set of bowls on the right? Write down the correct answer in the appropriate block in your answer sheet.
Answer:

4

Explanation:

As shown in the question, the original order of the set of bowls is: 5 6 3 2 1 4
Remember that the swapping of neighbouring bowls happens from right to left.

After passing a first worker, the order of the bowls is: 1 5 6 3 2 4 (4 swaps, all with bowl 1)
After passing a second worker, the order is: 1 2 5 6 3 4 (3 swaps, all with bowl 2)
After passing a third worker: 1 2 3 5 6 4 (2 swaps, all with bowl 3)
After passing a fourth worker, the set of bowls is sorted: 1 2 3 4 5 6 (2 swaps, all with bowl 4)

It's Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

Typically, automatic processing of data (which is what computational thinking is mostly about) is much easier when data is arranged according to some criteria – when it is sorted. Much effort has been spent by Computer Scientists on investigating sorting algorithms. The method for sorting sets of bowls that is described in this task is called a “bubble sort”. This sorting algorithm steps through a list of objects again and again, swapping any neighbouring objects which are in the wrong order. The list is sorted when no swap occurs during a pass through the list.

Bubble sorting is quite easy to understand compared to other sorting algorithms. Unfortunately, it is not very efficient. For sorting 1000 items, a bubble sort may use up to half a million steps in the worst case. Better sorting algorithms would use only about 10 000 steps.
Edgar is looking for a new home to live in.

He searched the internet and found a perfect flat for a very good price.

He has sent an e-mail to Francis, who is selling the flat, and received a quick reply:

Hi,

Thank you for your interest in my flat.

Although I am not in town, I can send you the key to the flat so you can inspect it, but I need a security deposit of R50 000,- beforehand.

To show my trustworthiness, I attach a copy of my ID.

Cheers,

Francis

Edgar is unsure what to do and is asking for your help.

**Question:**

What would be your best advice? Write down the letter of the best advice in the appropriate block on your answer sheet.

A. Pay the deposit. With the ID you can always go to the police if you don’t get the deposit back.
B. That is perfect. If you like the flat, you can keep the key right away.
C. Don’t pay the deposit, there is a high chance that this is a mail fraud.
D. Pay the deposit, go and have a look and decide later on.

**Answer:**

Response C

**Explanation:**

Response C would be the best advice. The copy of the ID could be "Photoshopped". You will not be able to meet the person to verify the ID. Statements 1 and 2 are not good as there is a high chance of not even receiving the key. Statement 4 is not good since the authenticity of the ID can’t be proven.

**It’s Computational Thinking:**

*Skills - Evaluation (EV)*

The Internet can be used to hide ones true identity and provides anonymity. Criminal people use this mechanism to get money from naive people. Very often spelling mistakes or high money values are within such emails and should raise awareness from the user.
The beavers have created a clever irrigation system for their fields. The water flows from a lake at the top of the hill all the way down to the fields numbered 1 to 6 at the bottom.

Along the water canals, the beavers have installed four water gates A, B, C and D, where the water can only flow either to the left or to the right.

**Question:**

Click on the arrows, so that only the fields numbered 2, 4, 5 and 6 are irrigated. Which combination will water the correct fields?
Write down only the letter of the correct answer in the appropriate block on your answer sheet.

A. At A right at B right at C left at D left
B. At A right at B left at C right at D left
C. At A left at B left at C right at D right
D. At A left at B right at C right at D left

**Answer:** B

**Explanation:**

**It’s Computational Thinking:**

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

The irrigation system behaves like a directed graph in graph theory. The graph shape is very similar to a tree with a root node and several leaves but in this graph there are directed connections between several vertices, which would not occur in a tree.

The water source forms the root of the graph and the crop fields are the leaves. If a node is connected to the root, water will flow there. Therefore, fields that need to be irrigated need to have at least one connection to the root node and fields that do not need to be irrigated must not have a connection.
All members of a beaver family have abilities.

- A daughter inherits all her abilities from her mother.
- A son inherits all his abilities from his father.
- Each family member also has one extra ability.

The diagram below shows the relationships between the beavers. It also shows the extra ability for each beaver.

**Examples:**

Mother Jennifer has inherited the ability to sing from Grandmother Maria, and she also has the ability to program.

Lisa inherits two abilities from her mother and also has the ability of writing. This means she can write, program and sing.

**Question:**

Look at the diagram above. Which one of the following statements is true? Write down the letter of the answer in the appropriate block on your answer sheet.

A. Sarah has abilities in reading, programming and singing.
B. Aunt Mary has abilities in dancing and swimming.
C. Tom inherits from Grandmother Margot the ability to calculate.
D. Tom’s abilities are riding, painting and photography.
Answer:
Statement D is correct

Explanation:

Statement D is correct, since Tom inherits painting from his grandfather and photography form his father.
Statement A is not correct, because Sarah does not inherit reading from her brother Charles.
Statement B is not correct, because Aunt Mary does not inherit swimming from her father.
Statement C is not correct, because Tom can not inherit abilities from his grandmother.

It’s Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

Inheritance is an important concept in object oriented programming. Parts of software systems can be reused and expanded. In this case abilities are like parts of software and are inherited and enriched with additional abilities.

In our task inheritance is different from programming because not everything is inherited.

The Elite diagram is in a form that would be familiar to object oriented programmers.
In the Seniors version the diagram is familiar to everyone.
In the game of Pirate Hunters players take turns moving a Pirate or a Policeman.  
When it is the police's turn, the player moves a policeman over to a neighbouring circle.  
The pirate is faster than a policeman, and skips a circle on his turn, moving two circles.  
A policeman cannot move to a circle that is occupied -- either by his colleague policeman, or the pirate.  
The game ends when the pirate is forced to move to a circle occupied by one of the policemen.  
The policeman goes first.

**Question:**

If the pirate plays the best way possible and makes no mistakes, how many moves will it take the police to capture him?  Write down the answer in the appropriate block on your answer sheet.

- A. The police can win in 2 turns.
- B. The police can win in 5 turns.
- C. The police can win in 3 turns.
- D. The police have no chance of winning.
Answer:
D. The police have no chance of winning if the pirate player plays the best way possible.

Explanation:
Let us assume that the police actually forced the pirate into the above situation and it is the pirate's turn (so she loses). What did the board look like before the policemen's last move? The police player moved one or other of the policemen up or down. Since the board is symmetrical, we shall assume that it was the right policeman that moved. The situation for the left one is the same. Thus the previous situation was one of these.

Let us go back one more move. The pirate must have come from the right. Hence the situation before that was one of these two:

The above situation, in which the pirate is captured, can therefore arise only from these two positions (and their mirrored versions). However, if the pirate is indeed a good player and has found himself to be in one of these two positions, he would surely move the pirate to the left and not up.

It’s Computational Thinking:

Skills - Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)

Programs that play board games work by computing the possible path through a “graph” of game states. They usually start from the current state (as opposed to here where we worked from the end state backwards) and compute the possible moves they and their opponent can make. They employ algorithms like Minmax, which score the potential moves that the computer can make by assuming that the opponent will make the best possible move. In complex games, like chess, computers analyse the moves up to a given depth (often around 15 moves) and use approximate methods to evaluate the position.
Beaver the Alchemist can convert objects into new objects. He can convert:

- Two clovers into a coin
- A coin and two clovers into a ruby
- A ruby and a clover into a crown
- A coin, a ruby, and a crown into a kitten.

After an object has been converted into another object, it disappears immediately.

**Question**

How many clovers does Beaver the Alchemist need to create one kitten?

Write down the correct answer in the appropriate block on your answer sheet.

**Answer:**
The answer is 11.

**Explanation:**

We can see the conversion as follows:

- Coin = 2 clovers
- Ruby = 2 clovers + 1 coin = 4 clovers
- Crown = 1 ruby + 1 clover = 4 clovers + 1 clovers = 5 clovers
- Kitten = 1 coin + 1 ruby + 1 crown = 2 clovers + 4 clovers + 5 clovers = 11 clovers

**It’s Computational Thinking:**

*Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)*

This task demonstrates how graphs can be used to represent dependencies between items. A graph is a data structure that is used a lot in computational thinking to demonstrate relationships. Graphs also make it easier to visualise a task compared to just reading the descriptions of the relationships in text.
Seven beavers are in an online social network called Instadam.

Instadam only allows them to see the photos on their own friends’ pages and the friends of friends’ pages. However it does not allow friends of friends of friends to see the photo.

In this diagram, if two beavers are friends they are joined by a line.

After the summer holidays everybody posts a picture of themselves on all of their friends' pages.

**Question:**

Whose picture will be seen the most? Write down the name in the appropriate block on your answer sheet.
Answer:
The correct answer is Chio.

Explanation:
In order to find the beaver whose picture gets seen by most beavers, you have to count the beavers that are at most two steps away. The beavers one step away are those on whose page the pictures will be posted and the beavers two steps away are those who can see these pages. Of course any beaver can only be counted once.

The following table summarises the info and helps us to see whose picture will be seen the most.

<table>
<thead>
<tr>
<th>Beaver</th>
<th>Direct Friends</th>
<th>Friends’ Friends</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ari</td>
<td>Bob, Chio</td>
<td>Ehab, Gerald</td>
<td>4</td>
</tr>
<tr>
<td>Bob</td>
<td>Ari</td>
<td>Chio</td>
<td>2</td>
</tr>
<tr>
<td>Chio</td>
<td>Ari, Ehab, Gerald</td>
<td>Bob, Dmitri, Fritz</td>
<td>6</td>
</tr>
<tr>
<td>Dmitri</td>
<td>Ehab, Gerald</td>
<td>Chio, Fritz</td>
<td>4</td>
</tr>
<tr>
<td>Ehab</td>
<td>Chio, Dmitri</td>
<td>Ari, Gerald, Fritz</td>
<td>5</td>
</tr>
<tr>
<td>Fritz</td>
<td>Gerald</td>
<td>Chio, Dmitri</td>
<td>3</td>
</tr>
<tr>
<td>Gerald</td>
<td>Chio, Dmitri, Fritz</td>
<td>Ari, Ehab</td>
<td>5</td>
</tr>
</tbody>
</table>

It’s Computational Thinking:

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

Many social networks use larger and more complicated versions of this concept. It is not always obvious that by posting something on a friend's page, it might be available to people other than the close friend.

Social networks themselves are incredibly powerful tools in today's world. Computing statistics on their users and their pages is useful to marketing departments and anyone else trying to understand a person or group of people.

Instadam could also be interpreted as a model of a miniature internet, with the beavers being websites and friends as pages “linked to”. Search engines typically rank these websites by some measure of popularity or importance, at least by the number of links to and from the website.

A widely used way to find the result by using a computer is to use the flood fill algorithm which can cope with systems with more than the two iterations in this example.
For his homework, Thomas had to write words on cards and connect them with rubber bands.
The teacher told him to connect any two words that differ by exactly one letter.
Thomas did this, as you can see in the picture on the right.
When Thomas returned from having a break he got a surprise.
Peter, his little brother, had erased all the words!

Also, the cards were completely mixed up, as you can see in the image on the left.
Importantly, the rubber bands still connected them as before.
Thomas was sure he could put the words back in the correct place.

Question:

Which of the pictures below contains the words in exactly the right places? Write down the letter of the answer in the appropriate block on your answer sheet.

Offered choices: A, B, C, D

Answer:
C

Explanation:
We can proceed by counting the edges going from each node. There are 2 nodes with 3 edges, 2 nodes have two edges and 2 nodes have 1 edge. There is only one node with one edge connected to a node that has two edges.
So we have identified the node for “EAR” and “CAR”. We can continue with this method ruling out the wrong answers as we proceed.

It’s Computational Thinking:
Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)

This is a problem about graphs. A graph is a set of objects, where some pairs of objects are connected
Two beavers live in lodges separated by a large forest. They decide to send messages to each other by shooting fireworks into the sky above the trees. Each message is a sequence of words, though the beavers only know five different words. The beavers can shoot two types of fireworks, one after the other, and know the following codes:

<table>
<thead>
<tr>
<th>Word</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td><img src="image1.png" alt="Fireworks Image" /></td>
</tr>
<tr>
<td>Tree</td>
<td><img src="image2.png" alt="Fireworks Image" /></td>
</tr>
<tr>
<td>Rock</td>
<td><img src="image3.png" alt="Fireworks Image" /></td>
</tr>
<tr>
<td>River</td>
<td><img src="image4.png" alt="Fireworks Image" /></td>
</tr>
<tr>
<td>Food</td>
<td><img src="image5.png" alt="Fireworks Image" /></td>
</tr>
</tbody>
</table>

For example, to send the (rather strange) message "food, log, food", a beaver would shoot:

![Fireworks Image](image6.png)

This can also have the meaning “Tree, Food”

**Question**?

How many **different** meanings can the following sequence of fireworks have? Write down the number in the appropriate block on your answer sheet.
**Answer:**
The correct answer is 4.

**Explanation:**
The message could mean any of the following:

- log, rock, food, river
- log, log, log, river
- rock, tree, river
- rock, food, log, river

To convince yourself that there are no more possibilities, you can systematically count them:

- Start with the first firework. It is not a message, so you can write a zero on it.
- The first two fireworks can only mean log. Write number one next to the second firework.
- We are at the third firework. It can have a meaning of any shorter sub-sequence plus one new word. Yet we see that there is no way to prolong the previously examined sequences (of length 1 and 2), so we only have one possible meaning (rock) and write 1 to the third firework.
- The fourth firework is finally somewhat interesting. It can either add the word log to the first two fireworks, or food to the first three fireworks, as shown by the arrows below. So we sum the two numbers at the 2nd and 3rd firework and write it to the 4th (1+1=2).
- We proceed applying the same idea to each firework to the right. We look one, two and three fireworks back. If those shorter messages can be prolonged with a correct word, we mark this fact with an arrow. Then we just sum the numbers “brought” by the arrows to the currently examined firework.
- At the last firework, we will have the number of all possible meanings.

**It’s Computational Thinking:**

*Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV)*

The process of building a solution systematically, step by step, using the previous steps like this is called dynamic programming. It makes the process much easier – just imagine trying to find all the full meanings of the message right away!

All digital information is represented using binary. That is, it consists of only the bits 0 and 1. Only longer combinations of 0 and 1 (“words” in this task) allow us to use more than two different meanings. But we also want to avoid ambiguity in our messages.

Most standard codes use the same number of bits per word, so there is only one meaning to each message. But if some word is used very often and some rarely, such code generates needlessly long messages. It is then useful to have shorter codes for frequent words (like “food”) and longer codes for less frequent words (like “rock”). Of course you can be smarter than the beavers in our task: If you generate a prefix code the messages will only have one meaning. This trick of shortening frequent data chunks without introducing ambiguity is used in data compression.
A mobile is a piece of art that usually hangs from a ceiling. You may remember one hanging from the ceiling in your bedroom.

A mobile consists of sticks and figures. Each stick has a few points to which figures or other sticks may be attached.

Also, each stick has a hanging point, from which it is attached to a stick further above (or to the ceiling).

The following example mobile can be described using these numbers and brackets:

\(-3 (-1 1) (1 1)) (2 3)\)

**Question:**

Which of the following mobiles is described by the following instructions:

\(-3 (-1 4) (2 (-1 1) (1 1))) (2 (-1 6) (2 3))\). Write the letter of the correct answer in the appropriate block on your answer sheet.
Answer:

The answer is mobile A

Explanation:

From the sample mobile, we can conclude the following about how a mobile is described: A structure that is hanging from a stick is described by an opening bracket; the stick position where it is hanging from (with the hanging point of the stick equal to 0); the descriptions of its parts; and a closing bracket.

Sticks (including the uppermost one) are described by describing the structures hanging from it. Figures are described by how many of them are hanging from each point.

It’s Computational Thinking:

Skills - Abstraction (AB), Decomposition (DE), Evaluation (EV), Generalisation (GE)

The structure of a mobile has an interesting property: If you detach a stick (except the uppermost one) from a mobile, you have a mobile again, with the detached stick being the uppermost stick now. That is, the parts of a mobile are constructed in the same way as the full mobile is constructed. If a single figure is considered as a (very basic) mobile, mobiles may be defined very briefly, as follows: A mobile is either (a) a single figure, or (b) a stick with one or more mobiles attached to it.

That is, in order to define a "mobile", we use the term "mobile" itself.

In computational thinking, such structures and their definitions are called recursive. With computer programs, recursive data structures may be assembled and processed with only a few lines of code thanks to the brevity of their definitions.
The teacher in the beaver school wants to give some material to his students. He found a portal with a scanned book which declares on its front page that it should be distributed according to a “Creative Commons License” (CC-BY-ND) that allows everyone to share, make a copy and redistribute the material in any medium or format for any purpose, even commercially, provided that appropriate credit is given.

The license also specifies that if one remixes, translates, or builds upon the book, the modified book may not be distributed.

**Question:**

Which of these actions is **NOT** permitted under the terms of this license?

A. Selling book copies to the students
B. Translating the book, keeping the translated copy for himself
C. Giving the students his translation of one chapter of the book
D. Putting a copy of the scanned book on the school website.

**Write down the letter of the correct answer in the appropriate block on your answer sheet.**

**Answer:**

C is not permitted and so is the answer to this problem.

**Explanation & It's Computational Thinking:**

Skills - Evaluation (EV)

Copyright is a complex issue: sometimes even the experts in local and international laws have difficulties in disentangling regulations and people's rights. It is not easy to decide if something can be uploaded, downloaded, used, distributed. Creative Commons licenses were invented in order to make it easier for authors and users to understand what they can do without breaking laws or contracts. The authors can easily state clearly if they want attribution (a mention of the original author), if they allow commercial use of their creation, if they prohibit derivative works, and if they put restrictions on the license of derivative distributions. These “rights” (known, in order, as BY, NC, ND, SA) can be composed independently in a new license. So, for example, the task describes a BY-ND license, and users should immediately understand that the author requires attribution of her/his original work and no derivatives are permitted. Everything else is allowed (at least according to the license...). Thus:

- A is permitted by the author's license (but perhaps not by school regulations...),
- B is permitted, as long as the derivative work is not distributed to others,
- D is permitted, since Creative Commons do not restrict sharing if the attribution to the original author is preserved.
A small chip is composed of a grid of contacts (marked as dots). Some are already connected (marked as line segments). Connectors are always only between adjacent contacts, horizontally or vertically. We want to connect S and R with a continuous sequence of connectors, which do not touch any already connected contacts.

**Question?**

How many different ways are there to connect S and R with the least possible number of connectors? Write down the correct answer in the appropriate block on your answer sheet.
Building a Chip

**Answer:**
The correct answer is 15

**Explanation:**
It is not difficult to find some shortest paths. You can imagine a wave spreading over the board, one contact at a time, starting at S and moving toward R. When the wave reaches a contact, it takes the least possible number of connectors (“steps” of the wave).

The first table shows the wave midway through filling in. The numbers show the order in which the wave reached them, and the blacked out cells indicate the connectors we cannot connect to. They are also the length of the shortest path to the respective contact. The highest numbers are the current edge of the wave. The table on the right shows the completion of the process.

But if you try to find all the shortest paths, you can easily get lost. Luckily, you do not really need to, you are just interested in the number of paths. So how can we break this task down in to some smaller tasks?

We need to realise this: A shortest path to any given contact must go through one of the adjacent contacts, which is exactly one connector closer to the start. If there are more such contacts, any of them can be used. So if you want to know the number of possibilities, you need to sum the possibilities to get to these adjacent contacts. The number of shortest paths to a certain contact is the sum of the numbers of the shortest paths to the adjacent contacts which are one connector closer to the start.

This allows you to follow a procedure, which will reliably give you the final number of shortest paths. You can proceed similarly as with finding the shortest path, filling in the table like a wave. Start at S. The contacts next to it can be reached in only one way. Then add the next contacts along the wave edge and always sum up the neighbouring numbers from the previous step.

The resulting table is shown below:
Building a Chip

It’s Computational Thinking:

*Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)*

Nowadays, integrated circuits (chips) are used in all electronic equipment and they have revolutionised the world of electronics. A chip can be made very compact, having up to several billion transistors and other components in an area the size of a fingernail. Without them all computation thinking would be theoretical.

The chips are very crowded and hence not easy to design. Computer scientists use many clever algorithms to do this. When deciding where to put two components, one criterion can be the number of paths between them: the more paths available, the less constraint for the rest of the board, because some third component will block these two components easily.

Finding the shortest path is a common problem in computational thinking. The described “wave” procedure is actually called a “breadth-first search”. Here it was adapted to count all the shortest paths. We did not have to check all the possible paths, because we proceeded systematically from the start. This approach is called dynamic programming.
There are 10 plates in a row. There is one apple on each plate.

Thomas the kangaroo loves to jump. First, he jumps onto the leftmost plate with the letter A.

On each single jump after this, he either jumps forward two plates, or backwards three plates.
(An example of the two possible jumps from plate F is shown with arrows in the picture.)

Thomas only jumps onto plates with an apple.
If he jumps onto a plate, he collects the apple from it.

**Question:**

If Thomas collects all 10 apples, which apple does he collect last? Write the letter of the correct answer in the appropriate block on your answer sheet.

A  B  C  D  E  F  G  H  I  J
Answer:
Plate 1

Explanation:
Thomas can collect all ten apples in the order 1, 3, 5, 2, 4, 6, 8, 10, 7, 9.
This is the only sequence of jumps that allows Thomas to collect all the apples. Why? To begin, Thomas must jump on plates 1, 3 and then 5 because otherwise he jumps to the left of the first plate. Next, he must jump to plate 2 because he can only get to plate 2 from plate 5 and he will not return to plate 5 later. The same kind of reasoning can be used to determine the only possible solution.

It’s Computational Thinking:
Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)
One way to solve this problem is to consider all possible sequences of plates and look for one that consists only of valid jumps. Each possible sequence is called a permutation and there are many of them. So this approach, which is called a brute-force or exhaustive search, takes a lot of time.

Another approach is to build a permutation one plate at a time. Once you figure out that a permutation or the start of a permutation is not valid (such as determining that the second plate cannot be anything but plate 3), you can remove the last plate(s) and continue building new permutations. This is called backtracking and if you can rule out many permutations early in your search, you can find a valid permutation much faster. This shortcut is called pruning.

One way to look at this problem is to view it as a graph. The plates are vertices and we join two plates by an edge if Thomas can jump between them. The task involves finding a path moving along edges that visits every vertex exactly once. This is called a Hamiltonian path. In general, it is very hard to find such a path. However, in this case, the graph is small and has special properties.

The general problem of finding a Hamiltonian path is known to be NP-complete which means that it belongs to a collection of very important problems for which we do not have efficient solutions. Interestingly, we know that if somebody finds an efficient solution to one of these important problems, then we instantly have a way to solve every one of these important problems efficiently.
Spies

Every Friday, six spies exchange all the information they have gathered during the week. A spy can never be seen with more than one other spy at the same time. So, they have to have several rounds of meetings where they meet up in pairs and share all the information they have at that point.

The group of 6 spies needs only three rounds to distribute all their secrets:

Before the meetings each spy holds a single piece of information. (spy 1 knows 'a', spy 2 knows 'b', etc.). In the first round spies 1 and 2 meet and exchange information so now both know 'ab'. The diagrams show which spies meet in each round with a line. It also shows which pieces of information they all have. After three rounds all information has been distributed.

Which of the following statements is true?

After an international incident one spy has stopped attending the meetings. What is the minimum number of rounds needed for the five remaining spies to exchange all information? Enter the number in the appropriate block on your answer sheet.
Answer:
The correct answer is 4

Explanation:
This is unexpected! The obvious answer is three (or less?) since we have one spy less. This is even stranger if we consider that four spies would quite obviously exchange the information in two rounds.

However, unsuccessful attempts at solving the task soon show us the root of the problem: since the number of spies is odd, one of them is “inactive” in every round. Say, that spy number 5 does not participate in the first round, but he participates in the second. Thus in the second round, only two spies will know his piece information (e). In the third round, these two spies will meet two other spies, so after three rounds (only) four spies will know e. The fourth round is needed to spread this information to the fifth.

Therefore, we proved that at least four rounds are needed. To show that they also suffice, we construct a scheme with four rounds:

![Diagram of spy interactions](image)

**It’s Computational Thinking:**

Skills - Abstraction (AB), Algorithmic Thinking (AL), Decomposition (DE), Evaluation (EV), Generalisation (GE)

When computers exchange information, they often exchange the data in pairs. Sometimes a problem involving how to share information along a whole network in the shortest time possible could arise. So computer scientists need to solve problems similar to this task. This problem is also known as the gossip problem ([http://mathworld.wolfram.com/Gossiping.html](http://mathworld.wolfram.com/Gossiping.html)). You can try to solve it for different numbers of spies and you might discover an interesting rule.

The solution of the problem was first solved – and the general rule described – in 1975. This and many similar problems occur in different areas of computer science especially those involving exchange of data, communication networks and cryptography.
Sponsors

The SA Talent Search is free to enter thanks to the sponsorship of the following organisations.

Standard Bank
www.standardbank.co.za/

The South African Computer Olympiad Trust.
www.olympiad.org.za

The Institute for IT Professionals South Africa.
www.iitpsa.org.za
Join us again next March for

Talent Search 2017

www.olympiad.org.za
Talent Search
Bebras Computational Thinking Challenge