

Team 1: Question 1

What is computational thinking?

- a. Giving instructions to a computer
- b. Thinking like a computer - in binary
- c. Using a set of techniques and approaches to help to solve problems

Team 1: Question 2

Which of the following is NOT a computational thinking technique?

- a. Decomposition
- b. Pattern recognition
- c. Coding

Team 1: Question 3

Which of the following is NOT an example of computational thinking?

- a. Planning what to collect and where to exit to complete a video game level
- b. Planning how to beat your enemies in a video game level
- c. Accidentally completing a video game level

Team 1: Question 4

What is a complex problem?

- a. A problem that, at first, is not easy to solve
- b. A problem that, at first, is not easy to understand
- c. A problem that, at first, is not easy to solve or to understand

Team 1: Question 5

To create a successful computer program, how many computational thinking techniques are usually required?

- a. Two
- b. Four
- c. Three

Team 2: Question 1

Why do we need to think computationally?

- a. To help us to program
- b. To help us solve complex problems more easily
- c. To help us to think like a computer

Team 2: Question 2

Which of the following is an example of thinking computationally?

- a. Planning out your route when going to meet a friend
- b. When going to meet a friend, wandering around until you find them
- c. When going to meet a friend, asking a parent to plan your route for you

Team 2: Question 3

Which of the following is NOT an example of computational thinking?

- a. Letting the bossiest friend decide where you should all go
- b. Considering the different options carefully before deciding upon the best one
- c. Discussing with your friends how much time and money you have before choosing from a shortlist of places

Team 2: Question 4

Which computational thinking technique involves breaking a problem down into smaller parts?

- a. Decomposition
- b. Abstraction
- c. Algorithms

Team 2: Question 5

When is a computer most likely to be used when using computational thinking?

- a. During decomposition
- b. At the end, when programming a computer
- c. When writing algorithms