**AP Computer Science I**

Performance Task

# **Create — Steganography Lab**

## **Overview**

In the Steganography Lab, students will expand on activities from the Picture Lab to

practice manipulating 2D arrays and exploring steganography techniques. Steganography

is the practice of concealing messages or information within other non-secret text or

data. Because images can be represented as 2D arrays of pixels, it makes sense to

use images as our non-secret data.

This lab is an introduction to steganography and includes concepts related to

encryption and information security. Whenever items are purchased online, or sensitive

information such as banking data is accessed from a phone or computer, it’s these

concepts that keep the information safe and secure. Understanding how information

is represented, as well as how it can be manipulated, is an important aspect of

programming. This lab will help provide a foundation for some of these topics.

## **Assessment**

You will be provided with 12 hours of class time to complete and submit the following:

* A video of your program running
* Written responses about your program and development process
* Program Code

Your teacher will share submission guidelines that include suggestions for creating video and PDF files.

## **General Requirements**

You are required to:

* Iteratively design, implement, and test your program.
* Independently create at least one significant part of your program.
* Create a video that displays the running of your program and demonstrates its functionality.
* Write responses to questions about your program.
* Include your entire program code.

## **Program Requirements**

Your program must demonstrate:

* Students are able to write simple iteration statements that operate like counting loops but have difficulty writing programs where each iterative step triggers a separate iterative process.
* Students are able to write single branching program code but have difficulty writing programs that require more than two options or pathways through the program
* Students are able to traverse and manipulate the elements of a 2D array when asked to process the 2D array in row major order but have difficulty traversing the 2D array using a different order (i.e., column major or back and forth).

## **Submission Requirements**

### 1. **Video**

Submit one video in .mp4, .wmv, .avi, or .mov format that demonstrates the running of at least one significant feature of your program. Your video must not exceed 1 minute in length and must not exceed 30MB in size.

### 2**. Written Responses**

Submit one PDF document in which you respond directly to each prompt. Clearly label your responses 2a – 2e in order. Your response to all prompts combined must not exceed 750 words, exclusive of the Program Code.

## **Program Purpose and Development**

1. Provide a written response or audio narration in your video that:

Identifies the programming language.

* Identifies the purpose of your program.
* Explains what the video illustrates.

(Approximately 150 words)

1. Describe the development process used in the completion of the project.

(Approximately 200 words)

1. Provide the method header for one method that you implemented that takes at least one parameter. Explain why you chose the given parameters, including type, and why you made the method static or non-static. How would your code have been affected if you had made a different decision?

(Approximately 200 words)

1. Provide a code segment where the elements in a data structure are traversed. Other than specific syntax, explain how using a different data structure would change the complexity of your code. Provide an equivalent code segment to the one included above that uses a different data structure.

(Approximately 200 words)

1. Capture and paste your entire program code into the PDF.

* Include comments or citations for program code that has been written by someone else.

## Tasks

### **Activity 1 - Explore**

**Description**

Steganography is not part of the curriculum for this course, and is used as a way

to motivate 2D array traversals in an engaging and interesting way. Additional

information about steganography is available through the following links and can be

provided to students as a pre-reading assignment or as additional information for

students who are interested.

* <https://www.wired.com/story/steganography-hacker-lexicon/>
* <https://learncryptography.com/steganography/what-is-steganography>
* Binary Representations

Time To Complete: 1-2 Hours

### **Activity 2 - Research**

**Description**

In this lesson, students will explore and use 2D arrays. A 2D array is an array that stores arrays! They will represent collections of related primitive or object reference data using two-dimensional (2D) array objects. Students will extend their learning on 2D arrays by traversing through them. When attempting to access all elements in a 2D array, we can do so in two different ways. Row-major order traverses the 2D array by accessing each value in a row before moving to the next row and column-major order traverses each column down all rows before moving to the next column. They will Traverse 2D arrays using nested for loops and enhancement for loops.

* Unit 7
* Exploring Color

Time To Complete: 6 Hours

### **Activity 3 - Ideate**

**Description**

When students are determining the algorithm to remove the rightmost two bits,

the table on the subsequent page will give them the answer. Consider giving them

only the first two pages of Activity 1 to give them the opportunity to determine the

algorithm on their own.

If students have trouble determining the algorithm for clearLow or revealPicture,

walk them through the provided pseudocode in the activity solutions, either in pairs or as an entire class.

Activity 1 focused on color values and manipulating individual pixels. This activity will step students through the process of hiding and revealing a picture within another picture. It is important for students to understand the concept of height and width, and the first few questions are intended to review these terms and also what it means for one image to “fit” within another.

It might be beneficial to have prints of the images to place on top of each other to

demonstrate the fit.

* Hiding and Revealing Picture

Time To Complete: 3-5 Hours

### **Activity 4 - Evaluate**

**Description**

Students will peer review and test each other’s code. They will work through any problems that might arise and fix anything based on the feedback received. Lastly students will plan the next part of the project which is identifying a hidden picture.

At the end of the previous activity, there was a discussion about how the code would need to change in order to allow the secret image to be smaller than the source image. As a class, modify the canHide method in Steganography.java to return true in the appropriate cases where the secret image is the same size or smaller than the source image.

* Identifying a Hidden Picture

Time To Complete: 1-3 Hours

### **Activity 5 - Construct a Prototype**

**Description**

At this point all students have seen one way that an image can be stored as a 2D array. Although this activity is not using ASCII representation of characters because of the bits required, ASCII representation could be introduced to students at this time.

This activity involves hiding and then revealing a text message in a picture. The ideas are the same as those involved in hiding and revealing a picture, but the text message must be split up and reassembled more carefully. If students have trouble determining the algorithm for hideText or revealText, walk them through the provided pseudocode in the activity solutions, either in pairs or as an entire class.

* Hiding and Revealing a Text Message

Time To Complete: 3-5 Hours

### **Activity 6 - Improve the Design**

**Description**

Students will peer review and test each other’s code. They will work through any problems that might arise and fix anything based on the feedback received. As a class, spend a few minutes reviewing the requirements of the open-ended activity.

The goal of the activity is to allow students to demonstrate their knowledge of 2D

array access, traversal, and manipulation in a way that’s interesting and engaging to

them. While it’s possible to create additional constraints or requirements, it’s best to provide students with as much freedom as possible. The use of pictures, pixels, and colors are intentionally not included in the list of requirements; however, it’s possible that students will continue to work with the given classes from this lab.

* Open Ended Activity

Time To Complete: 3-5 Hours

### **Activity 7 - Share Solutions**

**Description**

Students share their solutions and how they differentiate from the original design.

Time To Complete:1-3 Hours

### **Activity 8 - Reflect**

**Description**

Students answer the reflections questions and submit all evidence.

Time To Complete: 1 Hour