Joseph Sumner:

Prefect Rota Generator
Programming language: Python.
Purpose of code: to automatically create a prefect rota/timetable, taking into account their timetables in order to allocate them to the place that is nearest to the lesson they have prior to the duty. It will give each person the same number of duties. It will ensure no-one has more than one duty on a single day. It will display the rota in a table format and export it as a CSV file.
Scale of project: 196 lines, 8 functions.
Interesting data structures / implementations of algorithms:

- Use of nested dictionaries for timetable structure.
- Function which checks that all the necessary folders are in the program's directory. If they do exist it collects all of the CSV files in the "prefectTimetables" folder and if not it will create them.
- Dictionary containing names of all the prefects along with their location at school during the lesson prior to break or lunch (the times duties are allocated to) - uses Pandas to read the CSV files of the timetables. Also stored in this dictionary are the remaining number of duties each person needs to be assigned.
- Contains an algorithm which collects all prefects with unassigned duties and, using a weighted system, assigns them duties as close to their previous lesson as possible: the closest locations have the lowest weight etc. It creates a list of prefects with the lowest weighting and randomly assigns one to the duty.
- Function which formats the timetable dictionary so it can be used as a data frame for Pandas. This then allows it to be displayed in a table format and exported as a CSV file.

Specific File Hacker
Programming language: Python.
Purpose of code: made in order to expand understanding of existing malware to expand my cyber-security knowledge. The code searches the whole computer for the target file, then emails it to you and deletes the file afterwards so that there is no trace of it.
Alongside the development of this program, I also wanted to understand more about the Raspberry Pi Pico and its ability to act as an input device. An easy way of doing this was by using Pico Ducky. By using this, the program can be easily run on any computer simply by plugging in the Raspberry Pi Pico which will automatically type in the relevant commands and execute the program. I used this project to gain insight into how easy it is to be targeted by these sorts of malicious programs.
Scale of project: 123 lines, 6 functions.
Interesting data structures / implementations of algorithms:

- Algorithm that collects all available drives on a computer such as C-Drive, D-Drive etc.
- Main algorithm - searches entire computer for the target file based off of the following criteria:
    - File name
    - File type
    - File size limit

- Algorithm that sends target file to the user via email.
- Algorithm that deletes the target file afterwards.

Educake Completer
Programming language: Python.
Purpose of code: Automatically solve Educake (online quizzes for school subjects) questions using ChromeDriver, web scraping, image to text converters and ChatGPT.
Scale of project: 163 lines, 6 functions.
Interesting data structures / implementations of algorithms:

- Uses ChromeDriver to navigate and input responses into the Educake website.
- Algorithm which searches for any images on the displayed page. If it finds any, Pytesseract is used to convert the images into text which can then be inputted into ChatGPT.
- Algorithm to search for any buttons on the page and read the text associated with each button.
- It then uses the pyChatGPT library to input the question, along with other relevant data (such as text from buttons or images), into ChatGPT. The answer that ChatGPT gives is used by selenium to enter the answer into educake.

BBC Bitesize Notes Maker
Programming language: Python.
Purpose of code: to automatically create notes using ChatGPT from Bitesize webpages.
Scale of project: 86 lines, 4 functions.
Interesting data structures / implementations of algorithms:

- Function which correctly formats URL entered by the user.
- Algorithm which uses web scraping to gather all the text from all of the Bitesize pages of a specific topic.
- It then uses pyChatGPT to input this text into ChatGPT, asking it to make notes in bullet points with headings.
    - This is all stored to a text file.

Youtube Notes Maker
Programming language: Python.
Purpose of code: to automatically create notes using ChatGPT from YouTube playlists.
Scale of project: 113 lines, 5 functions.
Interesting data structures / implementations of algorithms:

- Function which uses pytube to collect all video URLs in the playlist that the user entered.
- Another algorithm is then used to get the ID, name and transcript of all videos.
- It then uses pyChatGPT to input this text into ChatGPT, asking it to make notes in bullet points with headings.
    - This is all stored to a text file.

AA Home Improvements Website
Programming language: HTML, CSS, JavaScript.
Purpose of code: portfolio website to be used by Andre for his up-and-coming handyman business.
Scale of project:

- HTML - 446 lines.
- CSS - 759 lines.
- JavaScript - 101 lines, 5 functions.

Interesting data structures / implementations of algorithms:

- Uses Typed.Js and Swiper Js.
- Function which changes the logo used in the header when you scroll down the page (only for the 'Projects' page).
- Various functions which, when you press on specific buttons, scroll the page down to the relevant section.
- Algorithm that checks if an element is in view with an offset.
- If you scroll it changes the styling of the header.
- The contact page uses formspree so a user can directly send messages to the website owner.

Bank Management System

Programming language: Python.

Purpose of code: can create account and log in, and then can deposit, withdraw and transfer 'money'. Replicates how a bank functions.

Scale of project: 532 lines, 1 class, 17 functions.

Use of object-oriented concepts: used classes to store text colours so they could easily be used later on in the program.

Interesting data structures / implementations of algorithms:

- Algorithm that collects account details by reading a text file which provides the username, password and balance.
- Register function - ensures the username is not already taken and that the password is valid. These are stored in the personal details text document.
    - Passwords are hashed using a hashing function.
- Login function - when a user enters their username and password, it then hashes the password and compares this to the hashed password stored in the text file for this username.
- File checker ensures all necessary text files are created.
- Use of various dictionaries to store account details, balances, and transactions.
- Algorithm that allows the transfer of cash from one registered account to another and then both of the account's balances are updated accordingly. The transfer can be named and it is recorded and stored in the transfer history.
- Algorithms to withdraw, submit, view transfer history, and view balance.

OTHER - (code not included):

1. **Fitness App**
    - For my A-Level Computer Science project, I am currently developing a fitness app for mobile using React Native.
    - It will allow a user to create an account (passwords are hashed) and then all of a user's data is stored.
    - When an account is created, the user is asked various different questions which are also stored to the profile.

- It will provide educational videos, create personal workout plans, track workouts, macros and certain metrics which will be displayed in a graph format.
- There will be a built-in AI Chat Bot which uses ChatGPT.

2. **Leetcode:**
    - I have completed various different Leetcode problems to help expand my problem solving abilities.
    - My most memorable one was the 01 matrix problem, when I had to return the distance of the nearest ‘0’ for each cell.
    - I did some problems with binary trees, including inverting one.
    - Additionally, I completed one titled ‘longest substring without repeating characters” and another to find the “longest palindromic substring”.

3. **My C# and Java Experience:**
    - To gain understanding of both of these languages, I learnt some basic programming constructs such as loops, functions, dictionaries and classes.
    - I made replicas of hang-man and Wordle using each of these languages, but in C# I made a GUI for the Wordle replica.
    - Also with C# I developed some basic Terraria mods.