

# Summer Task #1

Create your own version of the Computational Thinking poster.

- Include brief definitions of each of the 4 steps
- How could you represent the 4 steps

*Plenty of ideas on the internet!*

## COMPUTATIONAL THINKING

### DECOMPOSITION

Breaking big problems into smaller, easier to manage problems



### PATTERN RECOGNITION

Analyze & look for a repeating sequence



Remove parts of a problem that are unnecessary and make one solution work for multiple problems

### ABSTRACTION



Step-by-Step instructions on how to do something

### ALGORITHM DESIGN

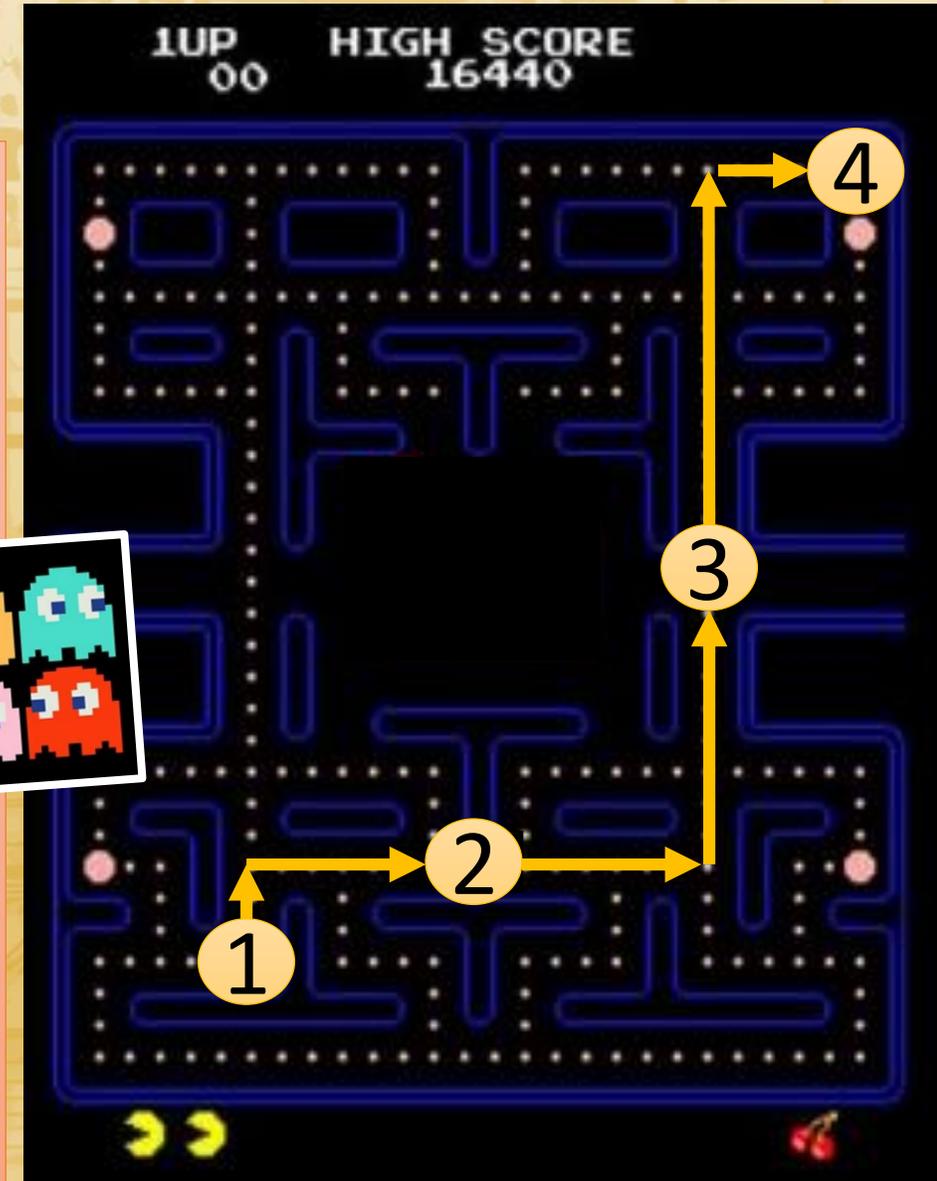
# Summer Task #2

When people played the classic Pacman game, the ghosts were given names and were said to have *personality*; this was because each ghost was programmed to respond differently to the player's movements. On the image to the left you can see the player's movement, going from location 1 → 2 → 3 → 4.

Using each ghost's algorithm you should be able to predict roughly where each ghost will be when the player is at each location.

## Task:

- Research each ghost's algorithm. Google: **Pacman Ghost Behaviour**
- Find a simple Pacman board similar to the one to the right
- Mark out 4 locations similar to the diagram
- For each location provide an approximate location for each of the four ghosts – you obviously can't know for sure but you should be able to make educated guesses. For position 1 feel free to set up the ghost's location to make it easier to track for the following locations. 😊
- Briefly justify your placements for the ghosts basing it on the ghost's algorithm



# Summer Task #3 for students new to CS

## Install Python

1. Google Python → Download Python
2. Click the big yellow download button (*currently 3.8.2*)
3. Save and run (*or just run it*).
4. Click recommended (*or change where you want to save it*)  
*Python will install but might not say confirm it has installed.*
5. Look for Python in the start menu – *there might be two versions, it doesn't matter.*
6. Use the Python IDLE

# Summer Task #3 for existing CS students

Attempt **three** programs from the **GCSE Code Challenge** booklet. Take your time and thoroughly understand the program and what your solution will need e.g. variables, variables for holding temporary values, inputs, outputs etc.

*Bonus challenge, pick another one from the **GCSE and A-Level coding challenges booklet** collection. 😊*