

# AI Assignment 1

Ganesh Patchipala 20420682

Ben Lynch 20348746

## A1.1

- **Fitness Function (fitness):** Measures the quality of an individual by counting the number of 1s. More 1s mean higher fitness.
- **Selection (select\_parent):** Chooses parents for the next generation using roulette wheel selection, favouring individuals with higher fitness.
- **Crossover (crossover):** Combines two parents to produce offspring. It uses a one-point crossover method with a set probability (crossover\_rate).
- **Mutation (mutate):** Introduces random changes to individuals at a set mutation rate (mutation\_rate), flipping bits to maintain diversity.
- **Evolution Process (evolve):** Runs the algorithm over a series of generations, each involving selection, crossover, and mutation, to evolve the population towards the optimal string.

## A1.2

The code for this part is very similar to 1.1 but the fitness function is different.

- **Fitness Function (fitness\_target):** Measures the quality of an individual by counting the number of bits that match a predefined target string. **Higher match counts mean higher fitness.** This differs from the first implementation where fitness was the count of 1s.

## A1.3

Again, the main difference in the code is the fitness function.

- **Deceptive Fitness Function (fitness\_deceptive):** Measures the quality of an individual by counting the number of 1s, but differently to previous parts. If the individual has any 1s, the fitness is the count of 1s. If the individual has no 1s the fitness is set to twice the length of the string which makes the algorithm consider this a highly fit solution.

## Description of Results:

Configurations	Starting Fitness	Fitness Progression	Final Fitness	Comments
1.1(String of 1s)	15.56	Consistently Increases	28.45	Steadily develops towards a string of all 1s. Comes out with high fitness as expected.
1.2(Specific Target String)	14.93	Progression varies and has some fluctuations	26.9	More complicated due to having to aim for a specific string. Lower fitness values indicate harder optimisation problem.
1.3(Deceptive Fitness)	15.66	Fluctuating Fitness Progression	27.61	Despite initial similarity to 1.1, deceptive fitness introduces complexity, resulting in a more erratic progression and slightly lower final fitness.

