# Genetic Algorithm Comparison with Beam Search

## Problem Setup

1. **Vocabulary:**

   ```
   V={′A′,′B′,′C′,′D′}
   ```

2. **Sequence Length:**

   Generate sequences of fixed length **5**.

3. **Fitness Function:**

   Define fitness of a sequence as the **sum of ASCII values** of its characters.

   - Example: `"ABCAA"` → `65 + 66 + 67 + 65 + 65 = 328`.

---

## Approach 1 – Beam Search

- Use **beam width k=3k=3k=3**.
- At each step, expand candidate sequences by adding one character from VVV.
- Keep only the **top 3 sequences** ranked by fitness.
- At the end, return the best sequence and its fitness score.

---

## Approach 2 – Genetic Algorithm (GA)

- Initialize a population of **6 random sequences** of length 5.
- Perform the following for **10 generations**:
  1. **Selection:** Choose parents based on fitness (higher fitness = higher chance).
  2. **Crossover:** Combine two parents by swapping part of the sequence.

3. **Mutation:** With small probability (e.g., 10%), randomly replace a character.

- After 10 generations, return the best sequence and its fitness score.

---

## Tasks

(a) Implement both **Beam Search** and **Genetic Algorithm** in Python.

(b) Print the **best sequence and fitness score** found by each method.

(c) In your code comments, discuss:

- Which method found more **diverse solutions**?
- Which is more **deterministic** and why?
- Which might be preferable for **language generation tasks** where diversity matters?

---