Assignment 2: Accompaniment Generation Report

- ## Problem:

We are given an input midi file, we need to create a chord accompaniment to be played with it that sound pleasant using an evolutionary algorithm.

- ## Approach:

A genetic algorithm was chosen which is a problem-solving method inspired by evolution. Starting with a set of random solutions is the core concept. The best solutions are then chosen from the pool, cross-bred, and mutated. This procedure, which is repeated a set number of times (generations) or until a perfect solution is found, results in better results over time.

First, the key of the melody was found using a library called `music21`. You can see how to install it below. Then a dictionary of some keys with their chords is stored in the code. Then, the genome or the genotype is represented as a string where each character is the chord that is played at a beat. The number of beats is calculated using the formula:

$$n_{beats} = sum(delta\ times)\ /\ ticks\ per\ beat$$

So, the length of the string is n_{beats}. Each character is between 0 and 6 which represent the scale's steps. Moreover, a list of the notes with were played at the same time as the chords were stored to be used in the fitness function.

- ## Algorithm Specifications:

1. Parent Selection: a fitness proportional selection was used which is stochastic universal sampling. It means that the probability of an individual becoming a parent depends directly on their fitness.
2. Variation Operators: single point crossover with mutation was used. The single point crossover is combining the genotypes of two individuals (parents) into a new pair after choosing a point at random and then swapping one half of the two genotypes. Mutation is choosing a random position of the genome and changing it into some random chord with low probability (1/ genotype length) for each position.
3. Fitness Function: the fitness function depends on 3 points. The first one is to avoid dissonances between the chord and melody note in the moment of the beginning of both. The function penalizes dissonance with value of -1. The second one is the root of the given chord matches the note played at the same time and it rewards it with 2. The last one is if the chord is "close" to the next chord to be played and it gets rewarded with 1.

4. Number of generation: 3000. It was enough to produce good results with a population size of 100.
5. Stopping Criteria: the evolution stops win the maximum number of generations is reached only.

- **Libraries:**

  Music21 installation:

  ```
  pip install music21
  ```

  Mido installation:

  ```
  pip install mido
  ```

- **Acknowledgment:**

  The solution was based on the genetic algorithm explained in the following article:
  https://towardsdatascience.com/simple-genetic-algorithm-by-a-simple-developer-in-python-272d58ad3d19