The aim of this practical work is to solve the Euclidean symmetric traveling salesman problem using a genetic algorithm.
The goal is to assimilate how genetic algorithms work, learn to adjust their parameters and measure their approximations.
The file provided in addition to this pdf, statement of the TP, contains the following elements:

- An executable named "GAforTSP2024.exe" representing the implementation of an GA for Euclidean symmetric TSP.
- 5 xml files representing the TSP instances to be used as dataset.
- A pdf containing the best results obtained for these instances.

Let us first recall the statement of symmetrical TSP:
A traveling salesman is expected to visit n cities, passing through each of them once and only once. Given the distances d[][] between any pair of these cities, we are asked to find the Hamiltonian tour whose length is minimal.

Part I. Theoretical reminder.

1. Write the mathematical formulation.
2. Determine the size of the search space.
3. What is the type of this problem and what complexity class does it belong to?

Part II. Discover the program.

1. Choose a small number of cities by clicking in the “cities” area then on “start”.
2. Add new cities and run the program as you go and observe the result.
3. Change the values of the various parameters and observe their effect.

Part III. Measuring the GA approximation.

1. Run the program 5 times for each instance provided then complete the following tables:
2. Lengths obtained:

Instance	1st run	2nd run	3rd run	4th run	5th run	Average	Best length	Approximation rate.	Approximation factor.
Att48									
Bayg29									
Eil51									
Berlin52									
Ch130									
Average									

Run time

Instance	1st run	2nd run	3rd run	4th run	5th run	Average
Att48						
Bayg29						
Eil51						
Berlin52						
Ch130						
Average						

What can we deduce?

Part IV. Adjusting GA parameters.

1. Run the program with the eil51 instance, varying the generation factor by one notch of 100. Plot the curve representing the length of the tour as a function of the number of generations.
   What can you deduce?
2. Run the program with the eil51 instance, varying the population size factor by a factor of 10. Plot the curve representing the length of the tour as a function of the population size.
   What can you deduce?