General Instructions:

- The submission due date of this assignment is **Saturday (2nd April 2022) midnight** (before **12:00 am**)
- The address of the email is [**YourName_YourID_AssignmentNumber**]
- Write a report (i.e. in a word file) that illustrates your main solution steps including the best fitness values and the average values, plotted over generations (with and without elitism).
- Zip your code and the report in a file entitled [**YourName_YourID_AssignmentNumber**] and submit it on Blackboard.
- This assignment should be delivered and discussed INDIVIDUALLY.

Requirements:

In your previous assignment, you wrote a Python program that implements the simple GA to solve the OneMax problem. You used binary representation (genes either 0 or 1), randomly initialized population of size 20 chromosomes, roulette wheel selection, one-point crossover with pCross = 0.6, bit-flip mutation with pMut = 0.05 and elitism of size 2. You made 10 runs with different random seeds, each for 100 generations. For each run, you had a vector saving the highest fitness and the average fitness in the population at each generation.

Part I:

Building upon your previous assignment (the same implementation details as mentioned above), solve the following optimization problem:

Maximize $F(x_1, x_2) = 8 – (x_1 + 0.0317)^2 + (x_2)^2$, where $-2 \leq x_1, x_2 \leq 2$.

Increase the population size to 100 chromosomes and compare the performance of your GA under the following:

1- Standard decoding and gray decoding.
2- Different precisions (different number of bits to encode the variables).

Part II:

Extend your implementation of Part I to solve the same optimization problem under the following constraint: $x_1 + x_2 = 1$.

The fitness after penalty is computed as $F(x_1, x_2) - | x_1 + x_2 - 1|$.

BEST OF LUCK!