```python
#himmelblau_test.py

from simple_genetic_algorithm import SGA

ALNUM = ["0", "1"]
VAR_STRING_LEN = 16
NUMBER_OF_VARIABLES = 2
VARIABLE_LEN = int(VAR_STRING_LEN / NUMBER_OF_VARIABLES)
PROBABILITY_OF_MUTATION = 0.005
POP_SIZE = 100
NUM_GENERATIONS = 1000
DOMAIN_MIN = -6
DOMAIN_MAX = 6

def himmelblau(*xs):
    """
    The function is defined on the 2-dimensional space.
    The function can be defined on any input domain but it is usually evaluated on
    x_i element of [-6, 6] for i = 1, 2.

    Takes in x, y cartesian values and returns function output of:
        f(x,y) = (x^2 + y - 11)^2 + (x + y^2 - 7)^2

    The function has four local minima at:
        f(x^*) = 0 at x^* = (3, 2)
        f(x^*) = 0 at x^* = (-2.805118, 3.283186)
        f(x^*) = 0 at x^* = (-3.779310, -3.283186)
        f(x^*) = 0 at x^* = (3.584458, -1.848126)

    Args:
        xs (List[num]): [x,y] cartesian coordinates
    Returns:
        (float): f(x,y) = value closer to 0 indicates a coordinate closer to know
        global minimum
    """

    x = xs[0]
    y = xs[1]
    return (x ** 2 + y - 11) ** 2 + (x + y ** 2 - 7) ** 2

print("Running Simple Genetic Algorithm on Himmelblau benchmark function")
SGA(himmelblau, POP_SIZE, ALNUM, VAR_STRING_LEN, VARIABLE_LEN, DOMAIN_MIN, DOMAIN_MAX,
NUM_GENERATIONS,
    PROBABILITY_OF_MUTATION)

print("\nHimmelblau benchmark test complete\n")
```