Genetic programming report

Authors:
- Krzysztof Czechowicz
- Mateusz Woźniak

Grammar

```
grammar gramatyka;

// Following grammar reflects in some way Golang programming language. Reflected Golang features is
// for example 'for' loop.
main: (statement | NL*) (NL* statement)* NL*;
statement:
 printStatement ';'
 | inputStatement ';'
 | conditionalStatement
 | loopStatement
 | variableAssignmentStatement ';';
name: STRING;
printStatement: 'print ' (expression);
inputStatement: 'read';
conditionalStatement:
 'if ' comparison codeBlock (' else ' codeBlock)?;
loopStatement: 'for ' comparison codeBlock;
variableAssignmentStatement:
 name '=' (expression | inputStatement);
comparison:
 expression ('==' | '!=' | '<' | '>' | '<=' | '>=') expression
 | notComparison
 | comparison (' and ' | ' or ') comparison;

notComparison: 'not' comparison;

expression:
 term
 | expression ('+' | '-') expression
 | inputStatement;

term:
 INTEGER
 | name
 | inputStatement
 | '(' expression ')'
 | term ('*' | '/') term;

codeBlock: '{' NL* main NL* '}';

INTEGER: '-'? [0-9]* '.'? [0-9]*;
STRING: [a-zA-Z][a-zA-Z0-9_]*;
NL: [\r\n]*;
WS: [ \t]+ -> skip;
```

Test: 11a

Problem solved!

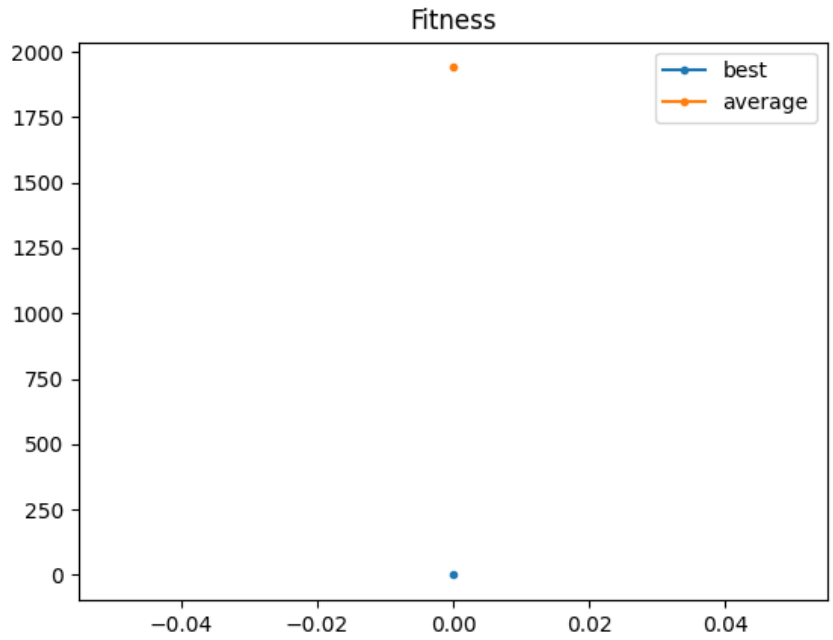
Fitness function:

```
    def test_11a_fitness(input_array, y):
        return 0 if 1 in y else 1000
```

Best fitness: 0

Best program:

```
{
 h = read;
 r = read;
 print 1;
}
```



Test: 11b

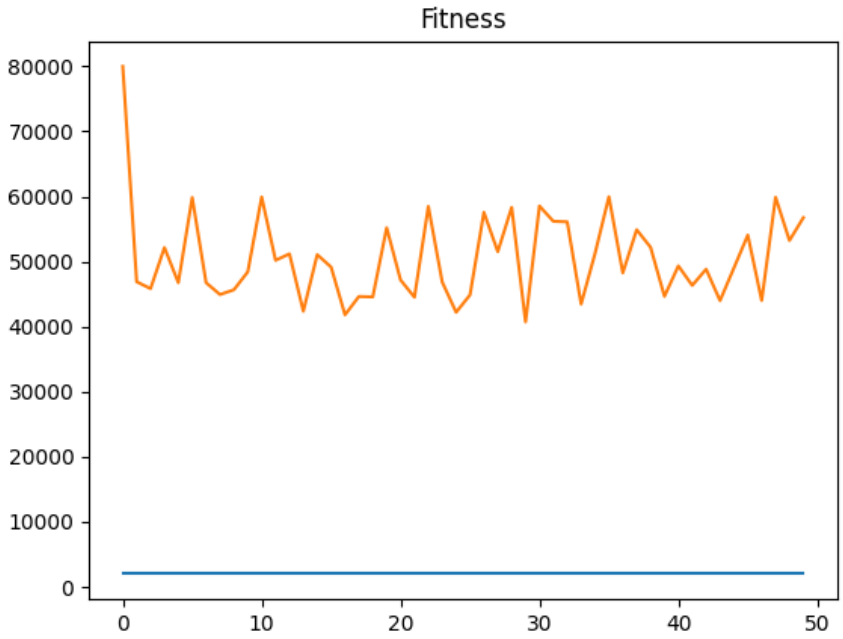
Problem not solved.

Fitness function:

```
def test_11b_fitness(input_array, y):
    return 0 if 789 in y else 1000
```

Best fitness: 99999

Best program:



Test: 11c

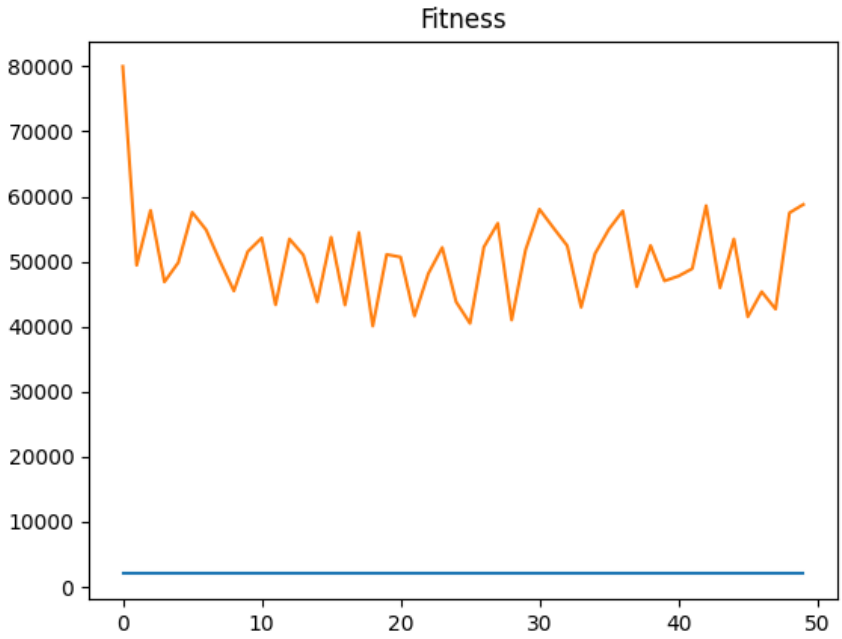
Problem not solved.

Fitness function:

```
    def test_11c_fitness(input_array, y):
        return 0 if 31415 in y else 1000
```

Best fitness: 99999

Best program:



Test: 11d

Problem solved!

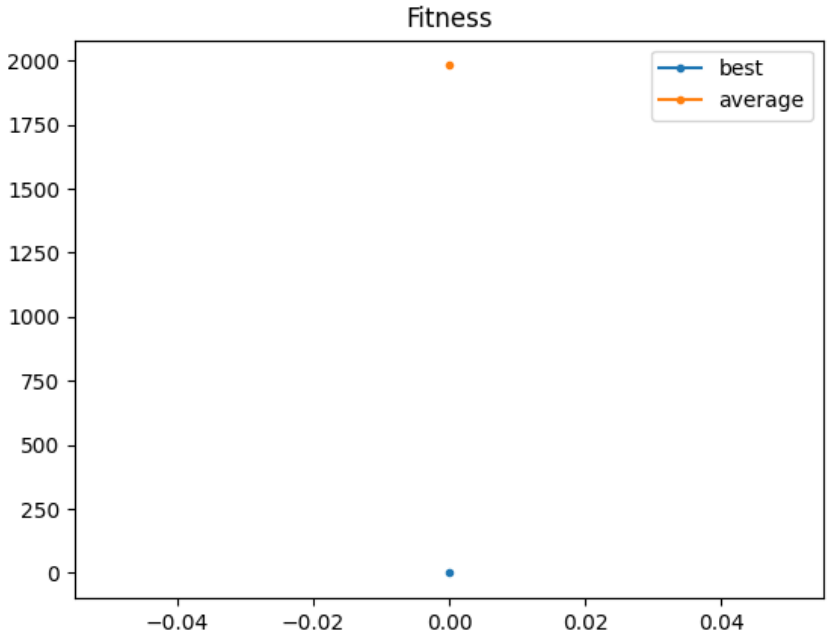
Fitness function:

```
    def test_11d_fitness(input_array, y):
        if len(y) == 0:
            return 1000
        return 0 if y[0] == 1 else 1000
```

Best fitness: 0

Best program:

```
{
 c = read;
 q = read;
 print 1;
 print -2;
}
```



Test: 11e

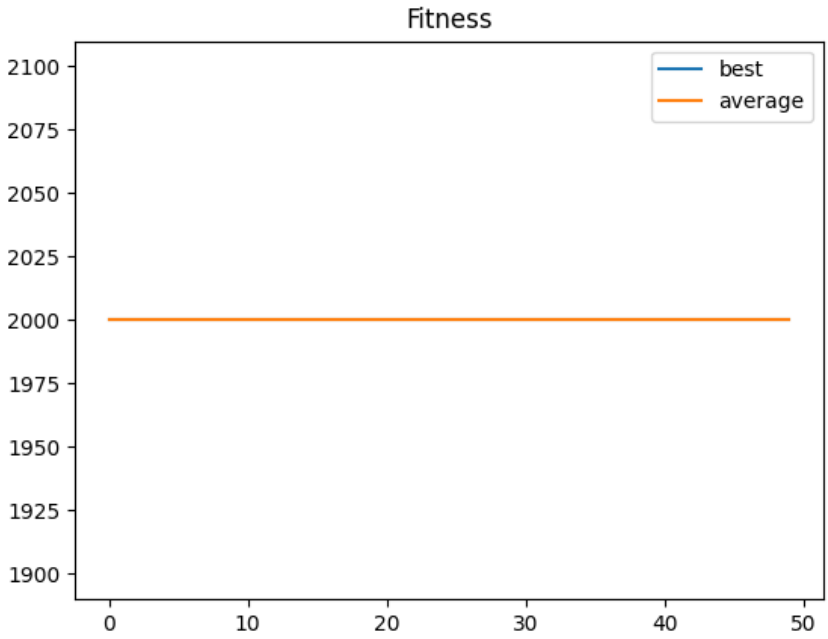
Problem not solved.

Fitness function:

```
def test_11e_fitness(input_array, y):
    if len(y) == 0:
        return 1000
    return 0 if y[0] == 789 else 1000
```

Best fitness: 2000

Best program:



Test: 11f

Problem solved!

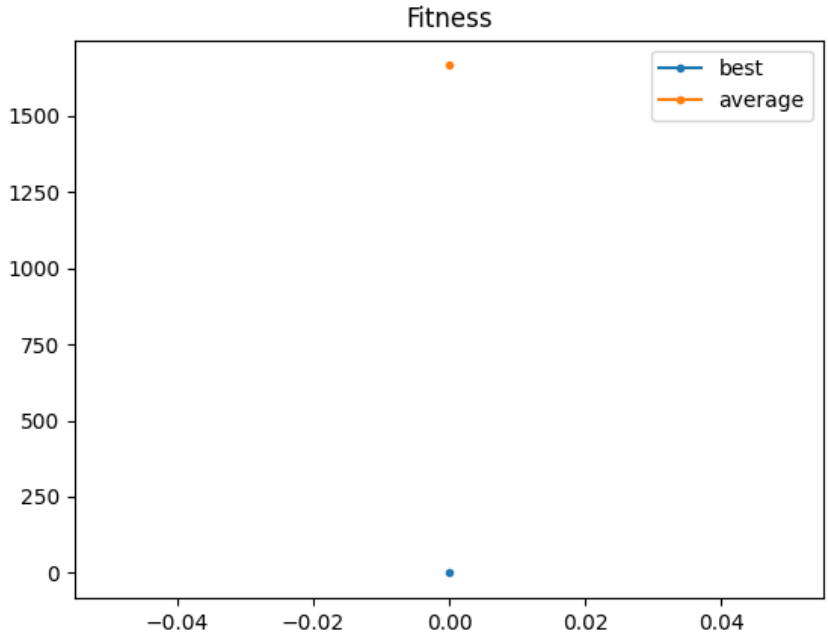
Fitness function:

```
    def test_11f_fitness(input_array, y):
        if len(y) == 0:
            return 1000
        if len(y) > 1:
            return len(y)
        return 0 if y[0] == 1 else 1000
```

Best fitness: 0

Best program:

```
{
 a = read;
 print 1;
 l = read;
 i = read;
}
```



Test: 12a

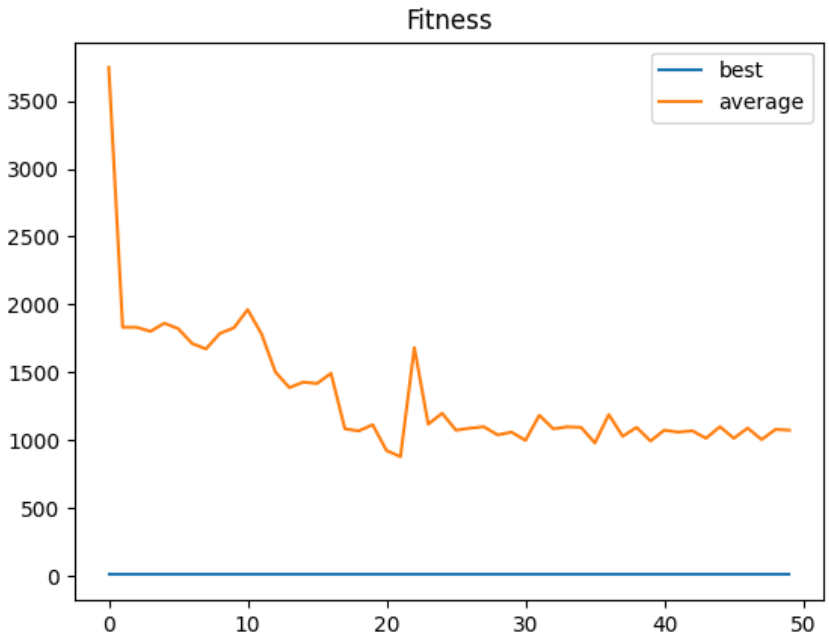
Problem not solved.

Fitness function:

```
    def test_12a_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000
        s = input_array[0] + input_array[1]
        return abs(s - y[0])
```

Best fitness: 10

Best program:



Test: 12b

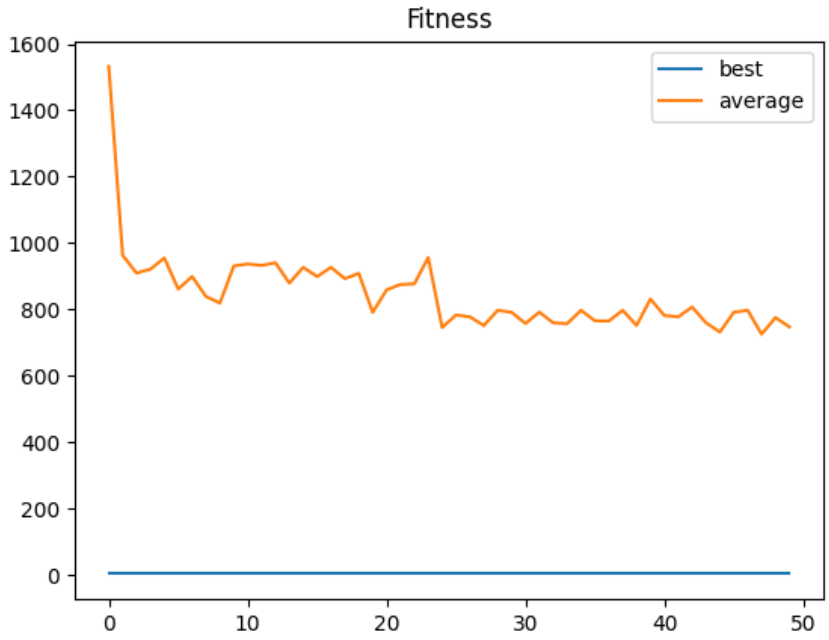
Problem not solved.

Fitness function:

```
def test_12b_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    s = input_array[0] + input_array[1]
    return abs(s - y[0])
```

Best fitness: 4

Best program:



Test: 12c

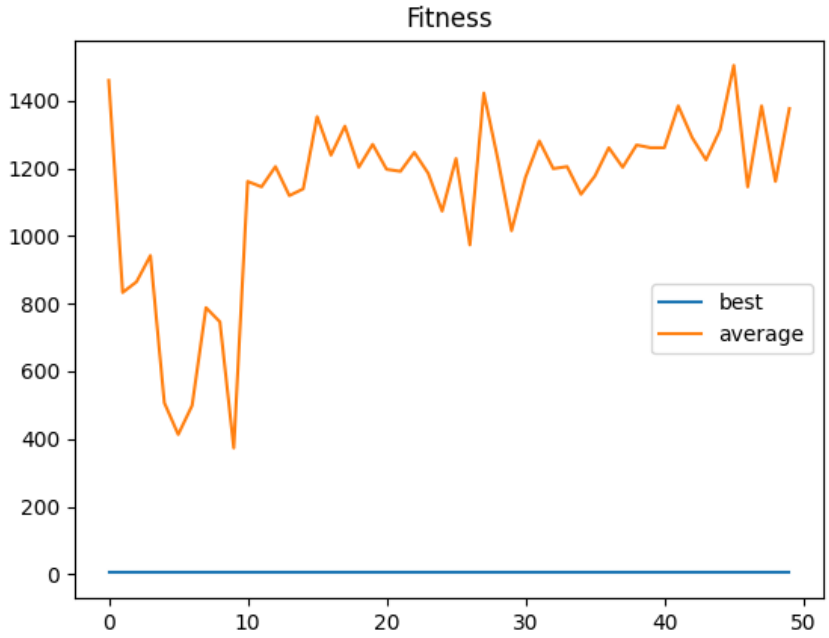
Problem not solved.

Fitness function:

```
def test_12c_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    s = input_array[0] + input_array[1]
    return abs(s - y[0])
```

Best fitness: 4.0

Best program:



Test: 12d

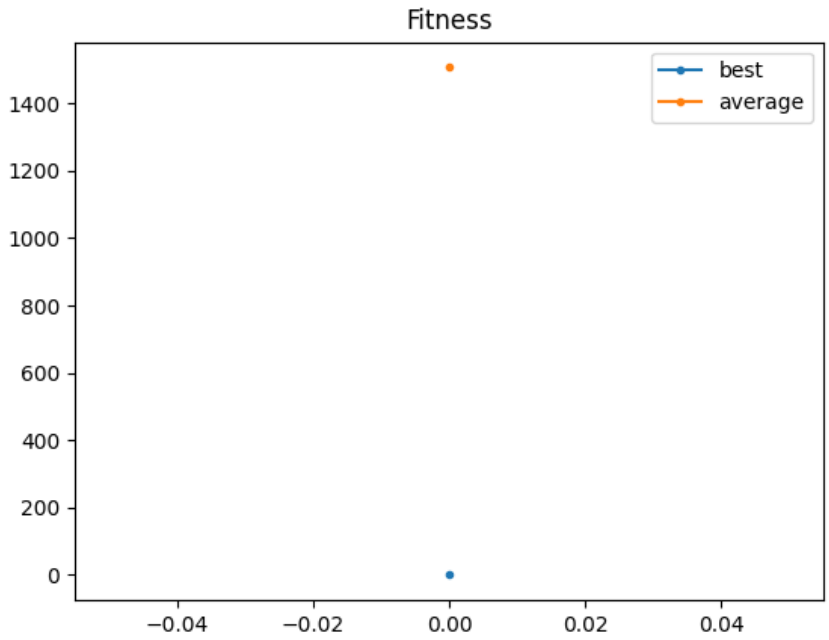
Problem not solved.

Fitness function:

```
def test_12d_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    s = input_array[0] - input_array[1]
    return abs(s - y[0])
```

Best fitness: 9999

Best program:



Test: 12e

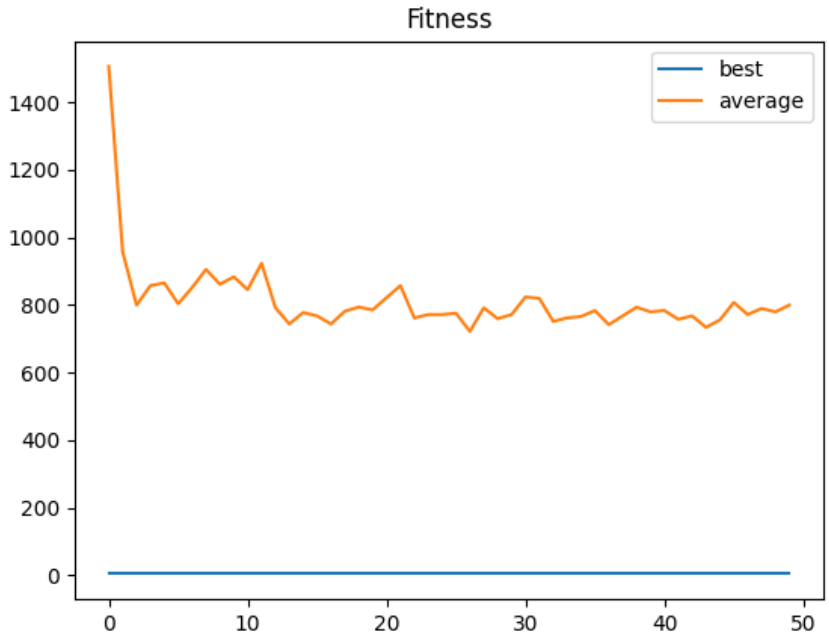
Problem not solved.

Fitness function:

```
def test_12e_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    s = input_array[0] * input_array[1]
    return abs(s - y[0])
```

Best fitness: 4

Best program:



Test: 13a

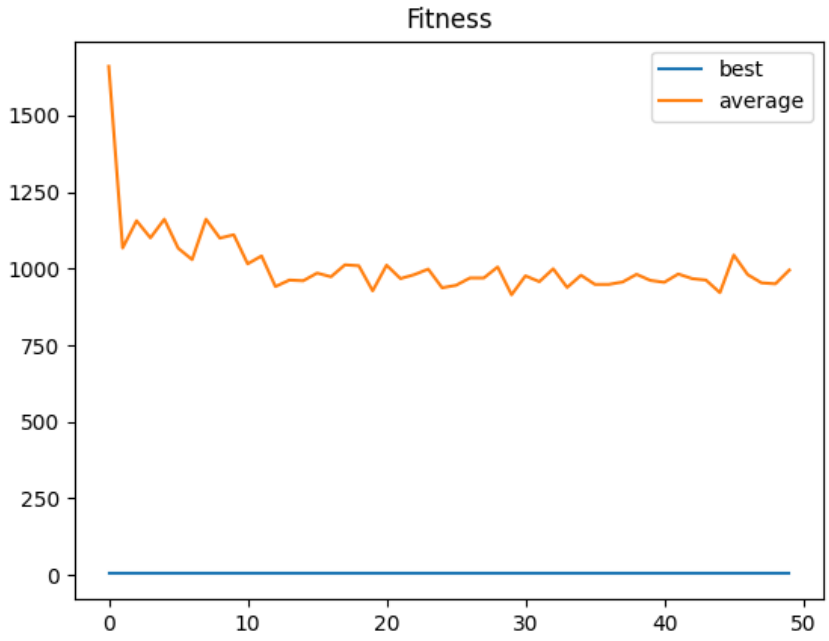
Problem not solved.

Fitness function:

```
def test_13a_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    if input_array[0] > input_array[1]:
        return 0 if y[0] == input_array[0] else 1000
    else:
        return 0 if y[0] == input_array[1] else 1000
```

Best fitness: 4

Best program:



Test: 13b

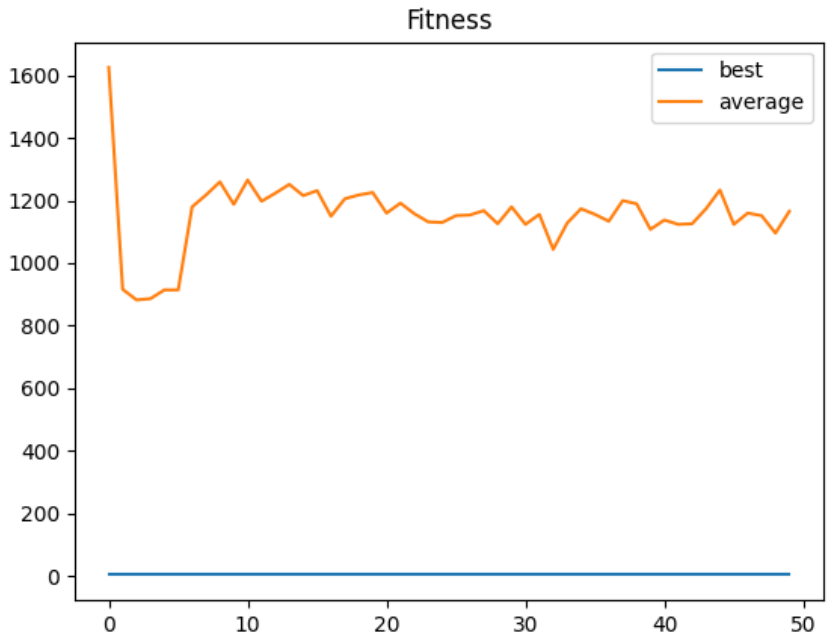
Problem not solved.

Fitness function:

```
def test_13b_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000
    if input_array[0] > input_array[1]:
        return 0 if y[0] == input_array[0] else 1000
    else:
        return 0 if y[0] == input_array[1] else 1000
```

Best fitness: 4

Best program:



Test: 14a

Problem not solved.

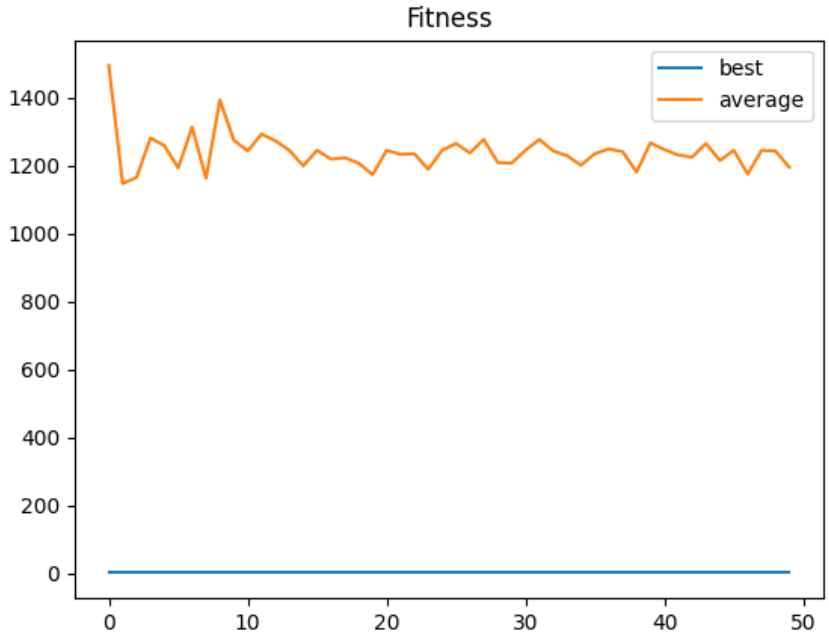
Fitness function:

```
def test_14a_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000

    avg = sum(input_array) / len(input_array)
    return abs(avg - y[0])
```

Best fitness: 2.0

Best program:



Test: 14b

Problem not solved.

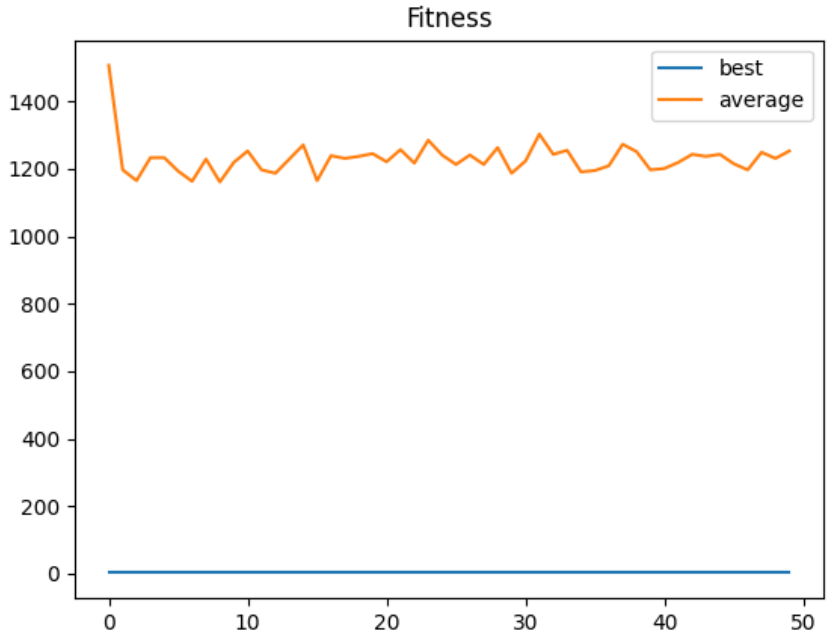
Fitness function:

```
def test_14b_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000

    avg = sum(input_array[1:]) / len(input_array[1:])
    return abs(avg - y[0])
```

Best fitness: 2.0

Best program:



Test: benchmark_1

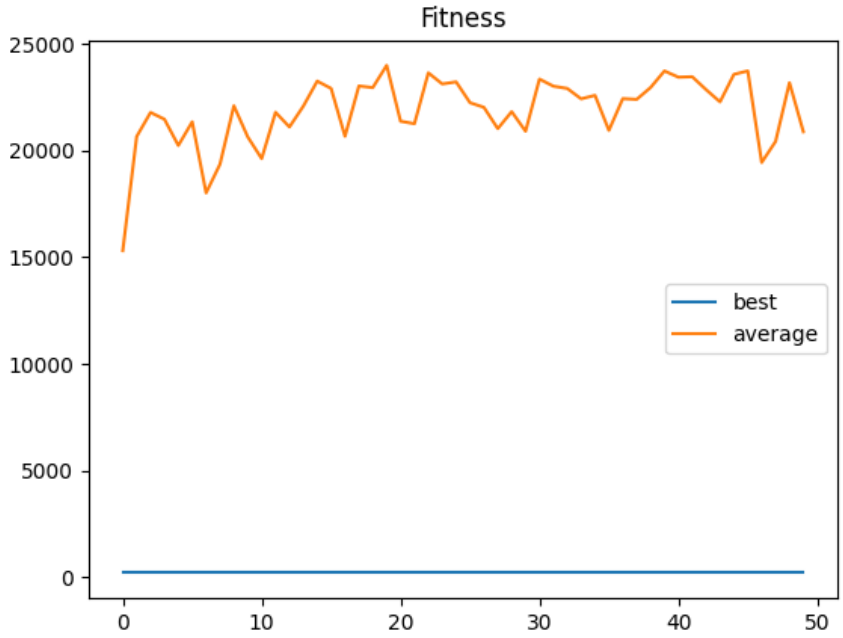
Problem not solved.

Fitness function:

```
def benchmark_1_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == input_array[0] + input_array[1] else 1000
```

Best fitness: 200

Best program:



Test: benchmark_17

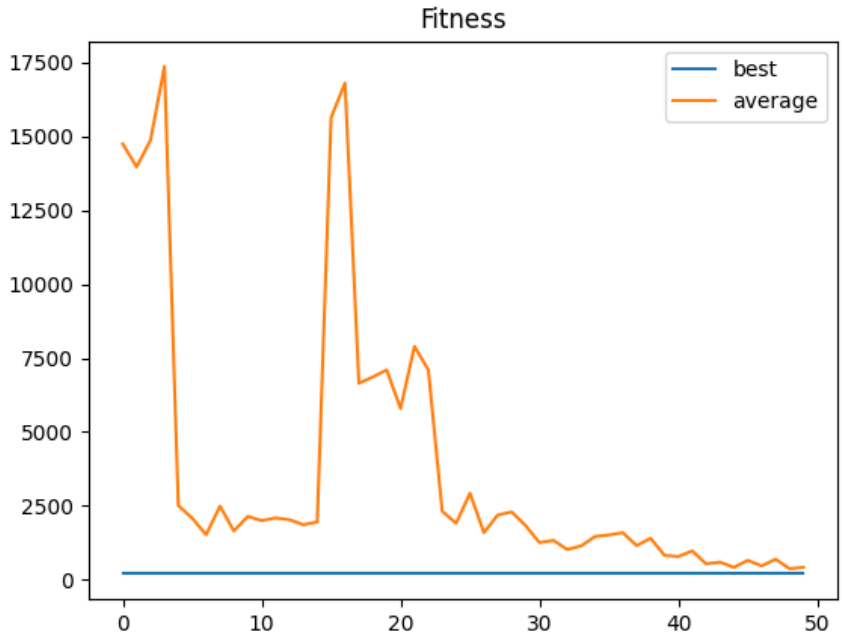
Problem not solved.

Fitness function:

```
def benchmark_17_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == sum([i ** 2 for i in range(1, input_array[0] + 1)]) else 1000
```

Best fitness: 200

Best program:



Test: benchmark_27

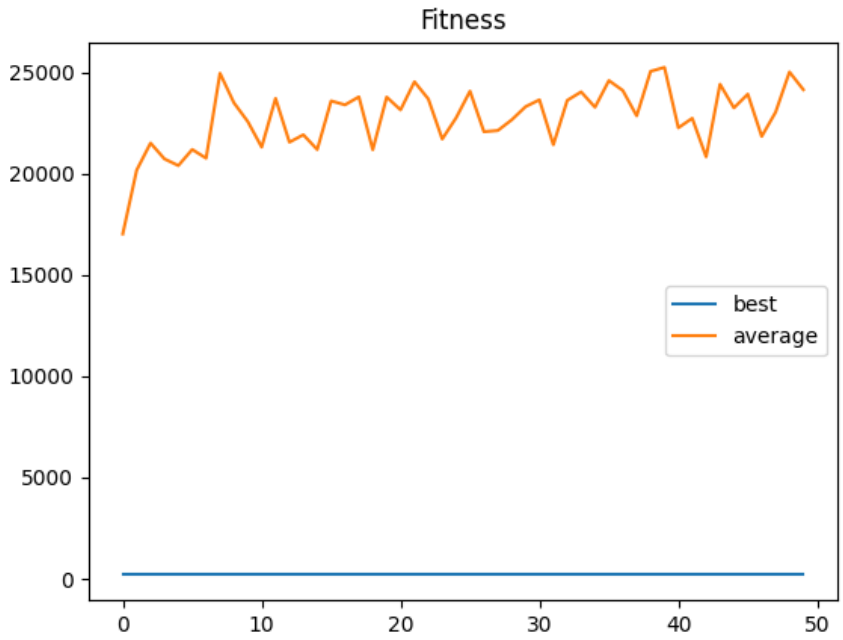
Problem not solved.

Fitness function:

```
def benchmark_27_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == sorted(input_array)[1] else 1000
```

Best fitness: 200

Best program:



Test: bool_10_AND

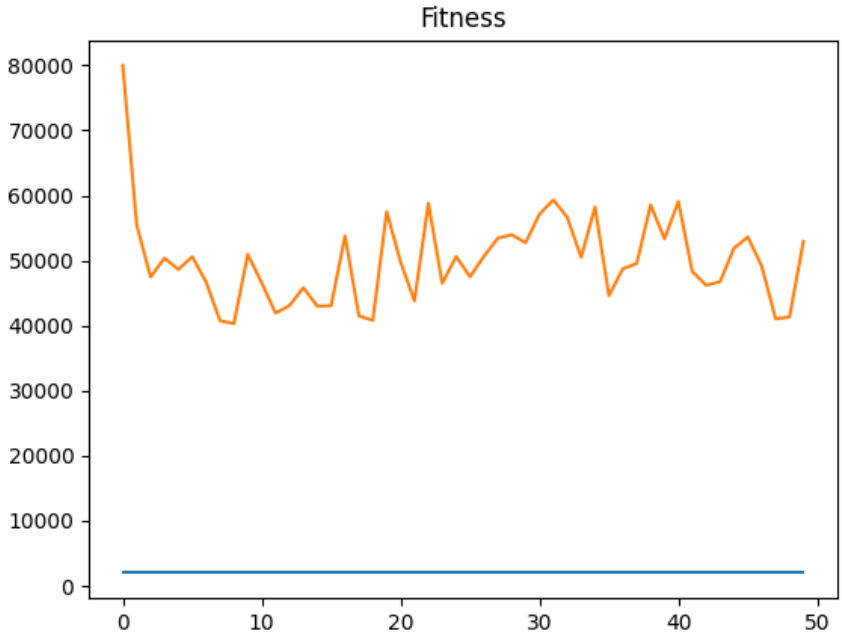
Problem not solved.

Fitness function:

```
def test_bool_10_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] and y[5] == input_array[5] and y[6] == input_array[6] and \
                y[7] == input_array[7] and y[8] == input_array[8] and y[9] == input_array[9] else 1000
```

Best fitness: 99999

Best program:



Test: bool_10_OR

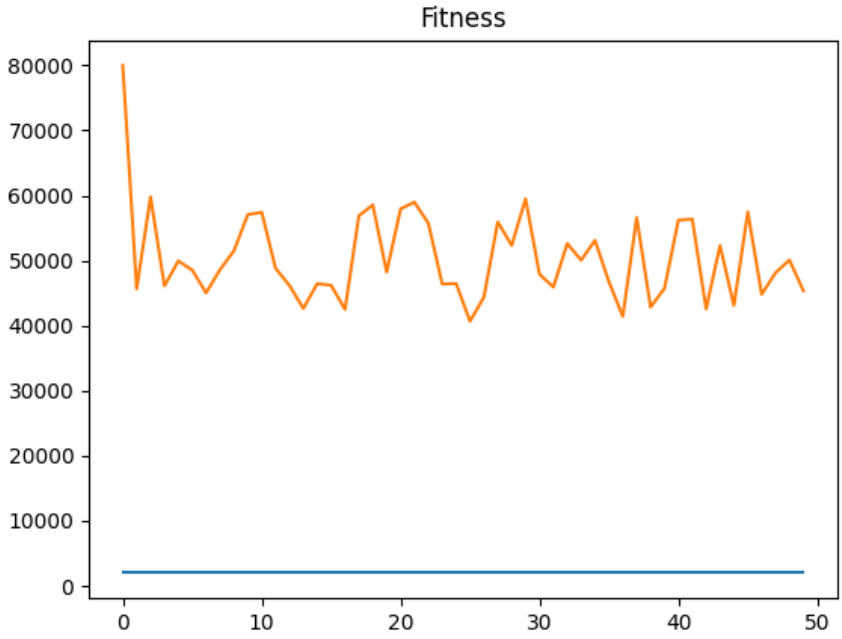
Problem not solved.

Fitness function:

```
def test_bool_10_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] or y[5] == input_array[5] or y[6] == input_array[6] or y[7] == input_array[
                    7] or y[8] == input_array[8] or y[9] == input_array[9] else 1000
```

Best fitness: 99999

Best program:



Test: bool_1_AND

Problem solved!

Fitness function:

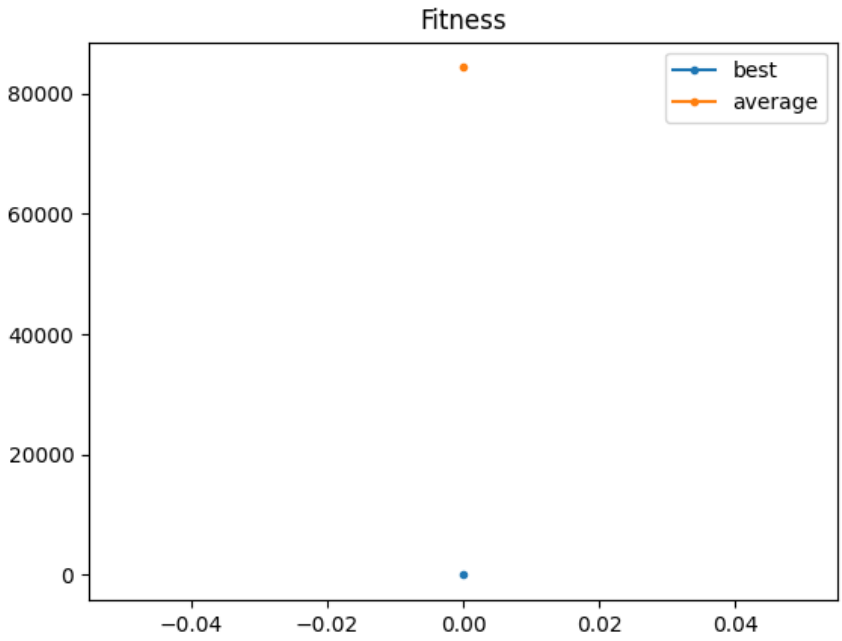
```
    def test_bool_1_AND_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

        return 0 if y[0] == 0 else 1000
```

Best fitness: 0

Best program:

```
{
 z = read;
 print 0;
}
```



Test: bool_1_OR

Problem not solved.

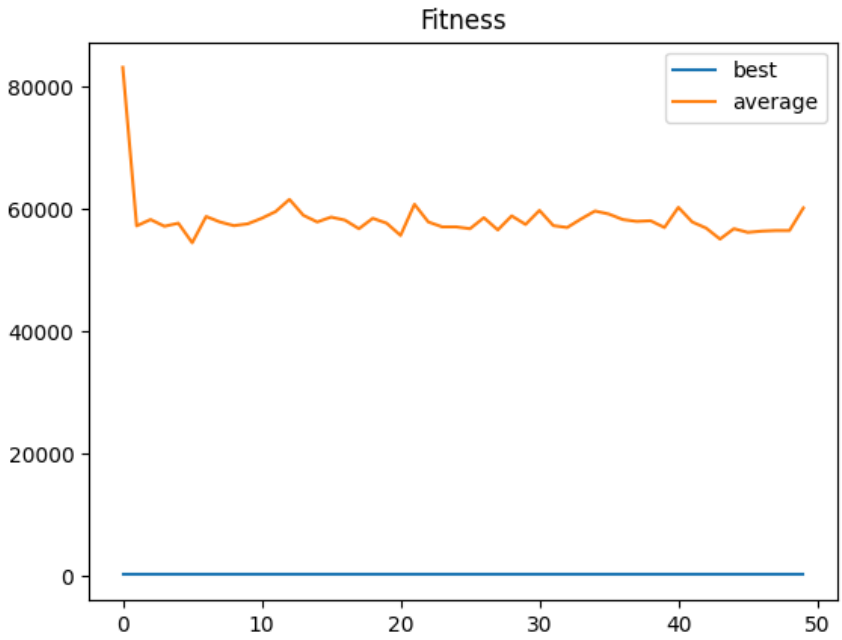
Fitness function:

```
    def test_bool_1_OR_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

        return 0 if y[0] == input_array[0] else 1000
```

Best fitness: 200

Best program:



Test: bool_1_XOR

Problem not solved.

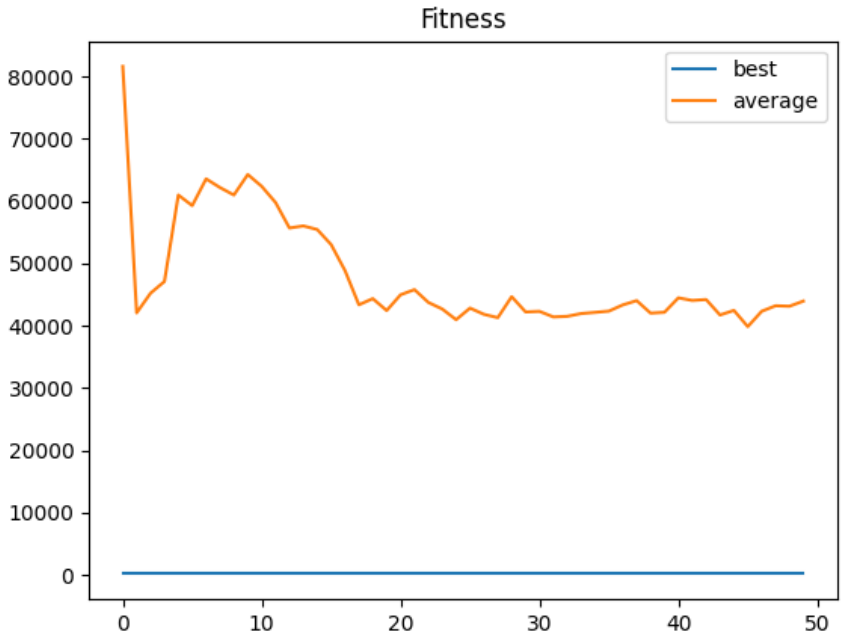
Fitness function:

```
def test_bool_1_XOR_fitness(input_array, y):
    if len(y) > 1:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] else 1000
```

Best fitness: 200

Best program:



Test: bool_2_AND

Problem not solved.

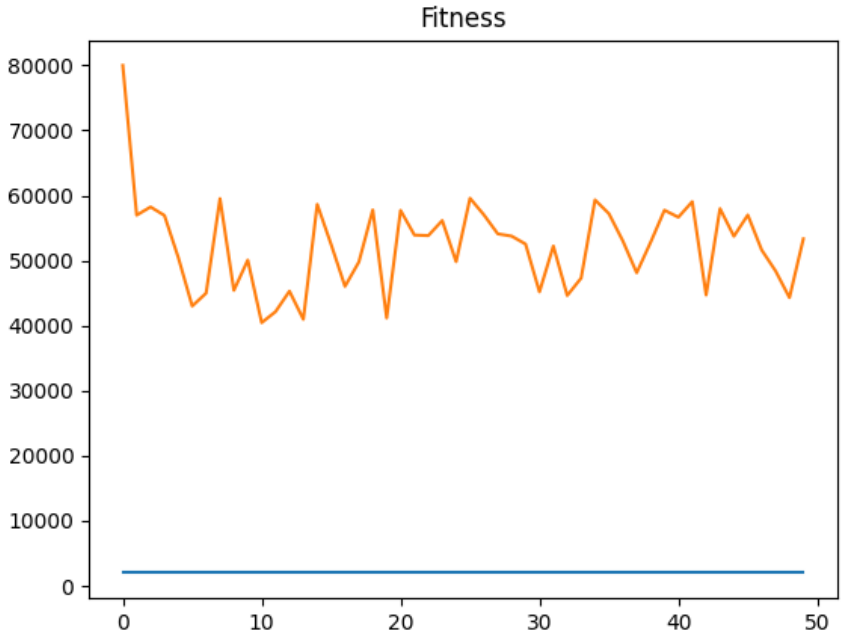
Fitness function:

```
    def test_bool_2_AND_fitness(input_array, y):
        if len(y) > k:
            return len(y)
        if len(y) == 0:
            return 1000

        return 0 if y[0] == input_array[0] and y[1] == input_array[1] else 1000
```

Best fitness: 99999

Best program:



Test: bool_2_OR

Problem not solved.

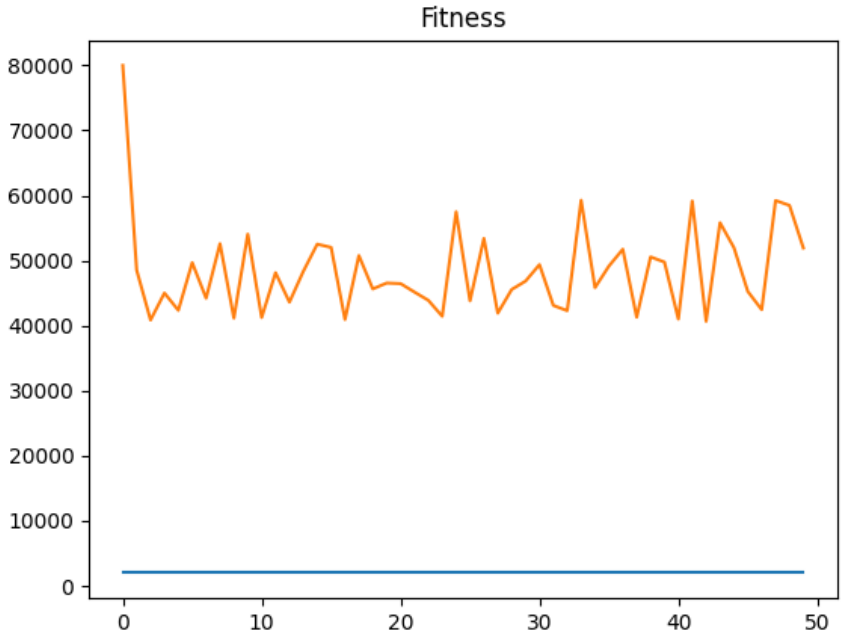
Fitness function:

```
    def test_bool_2_OR_fitness(input_array, y):
        if len(y) > k:
            return len(y)
        if len(y) == 0:
            return 1000

        return 0 if y[0] == input_array[0] or y[1] == input_array[1] else 1000
```

Best fitness: 99999

Best program:



Test: bool_2_XOR

Problem not solved.

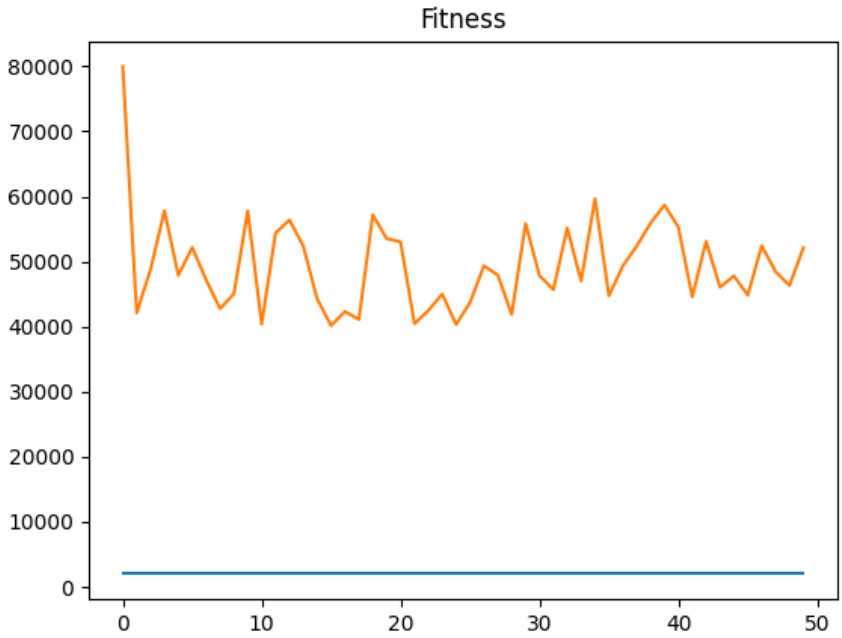
Fitness function:

```
def test_bool_2_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] else 1000
```

Best fitness: 99999

Best program:



Test: bool_3_AND

Problem not solved.

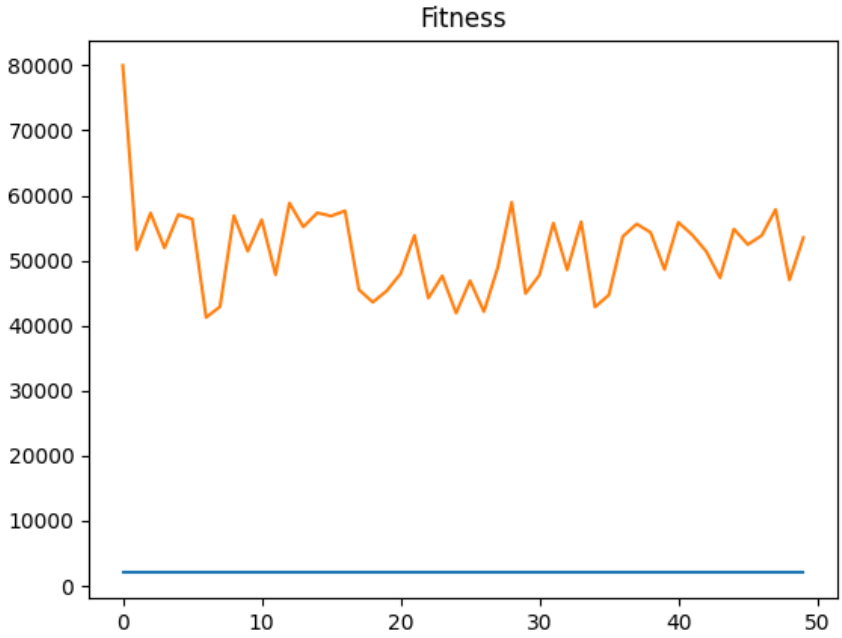
Fitness function:

```
def test_bool_3_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] else 1000
```

Best fitness: 99999

Best program:



Test: bool_3_OR

Problem not solved.

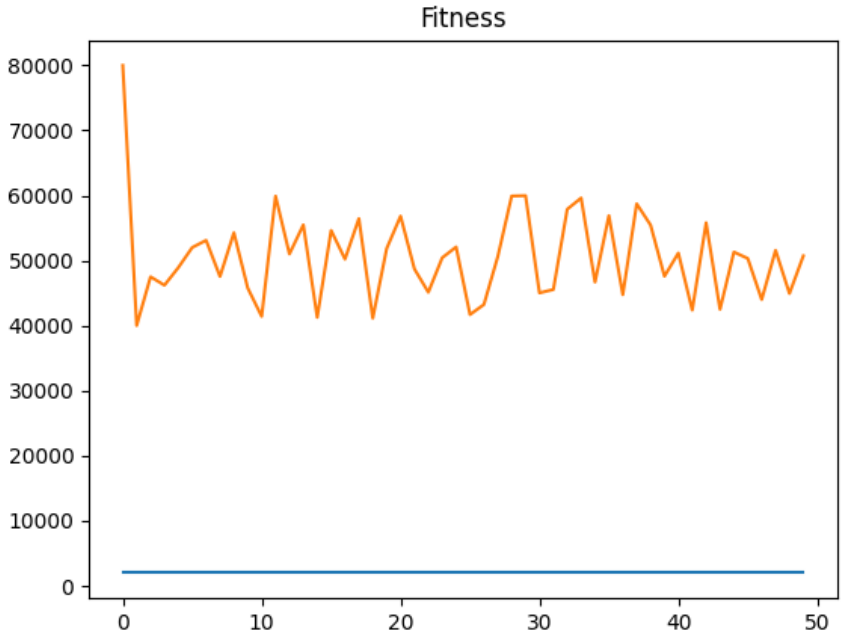
Fitness function:

```
def test_bool_3_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] else 1000
```

Best fitness: 99999

Best program:



Test: bool_3_XOR

Problem not solved.

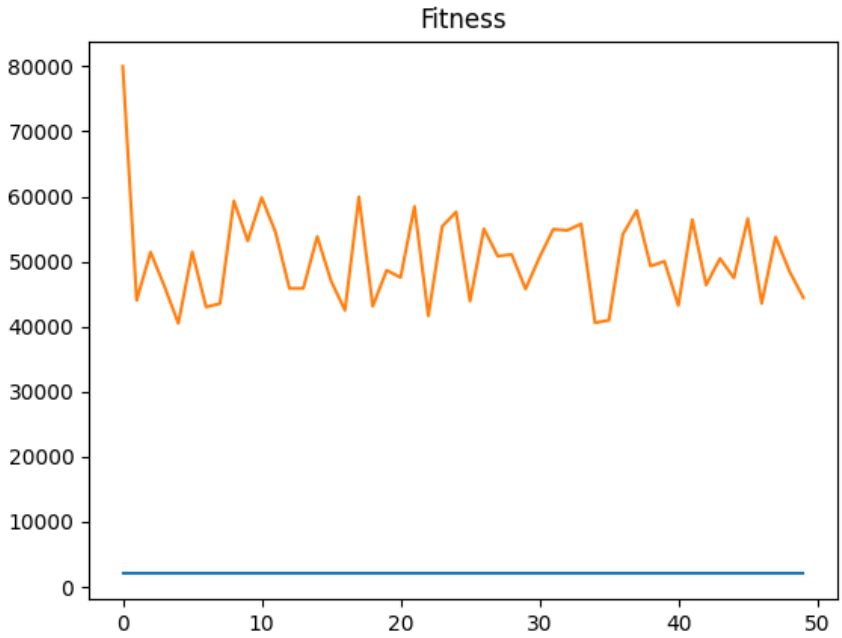
Fitness function:

```
    def test_bool_3_XOR_fitness(input_array, y):
        if len(y) > k:
            return len(y)
        if len(y) == 1:
            return 1000
        if len(y) == 0:
            return 1000

        return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] else 1000
```

Best fitness: 99999

Best program:



Test: bool_4_AND

Problem not solved.

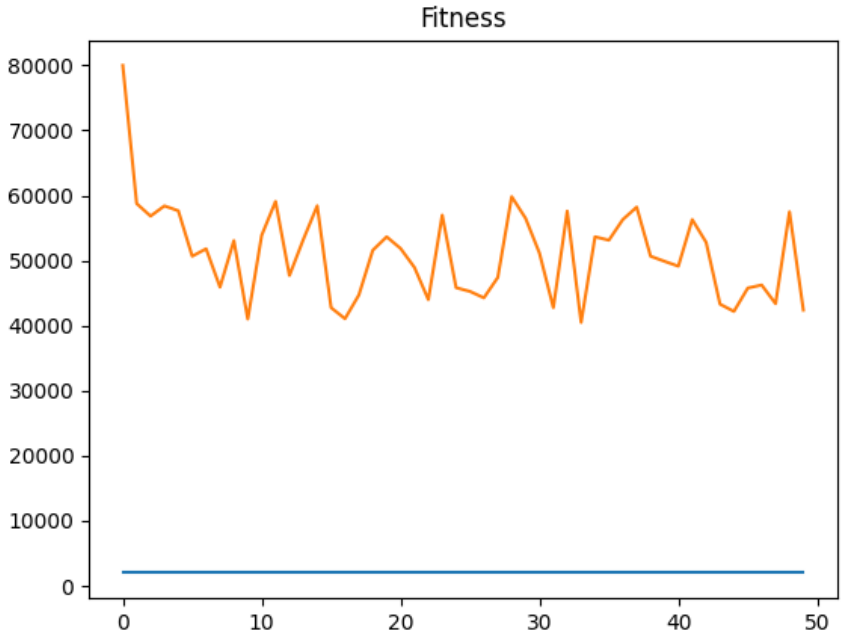
Fitness function:

```
def test_bool_4_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] else 1000
```

Best fitness: 99999

Best program:



Test: bool_4_OR

Problem not solved.

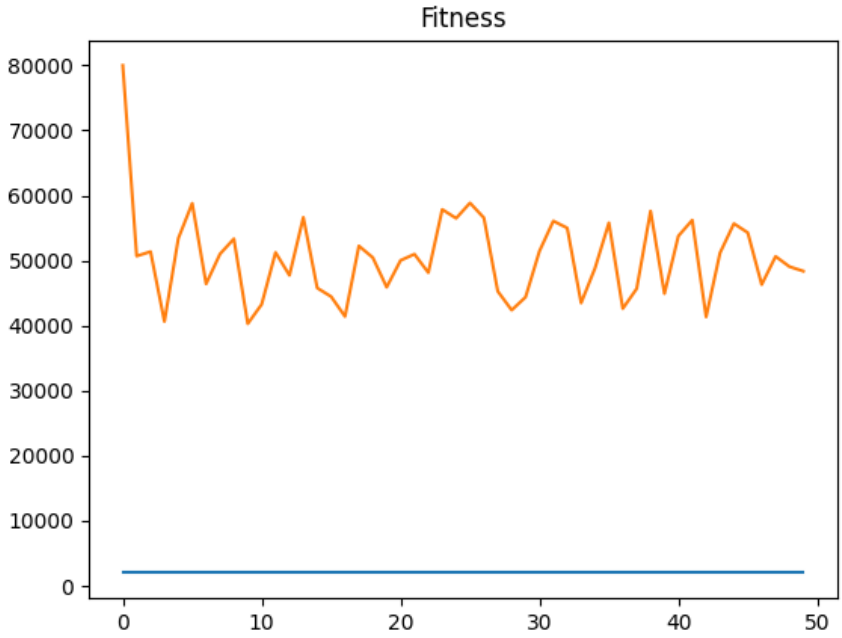
Fitness function:

```
def test_bool_4_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] else 1000
```

Best fitness: 99999

Best program:



Test: bool_4_XOR

Problem not solved.

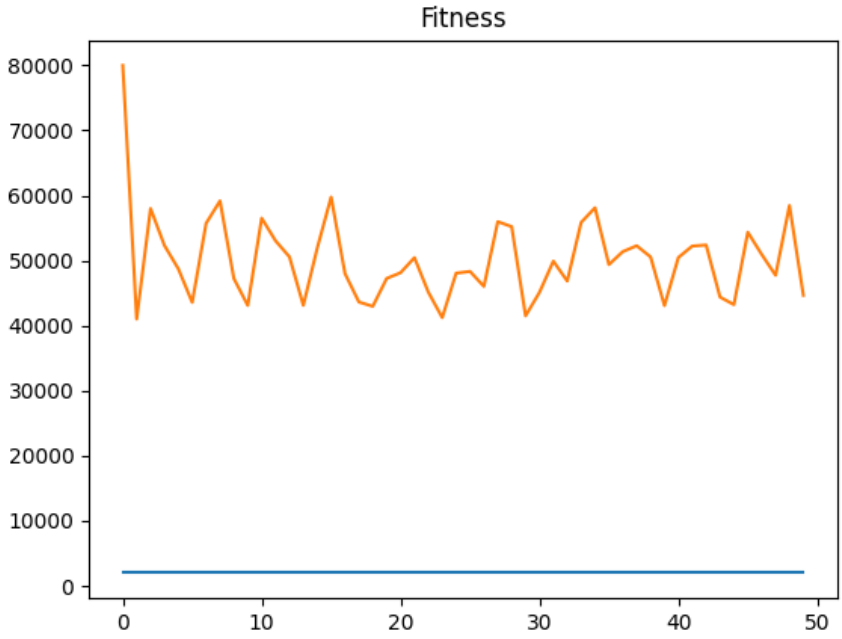
Fitness function:

```
def test_bool_4_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[
        3] else 1000
```

Best fitness: 99999

Best program:



Test: bool_5_AND

Problem not solved.

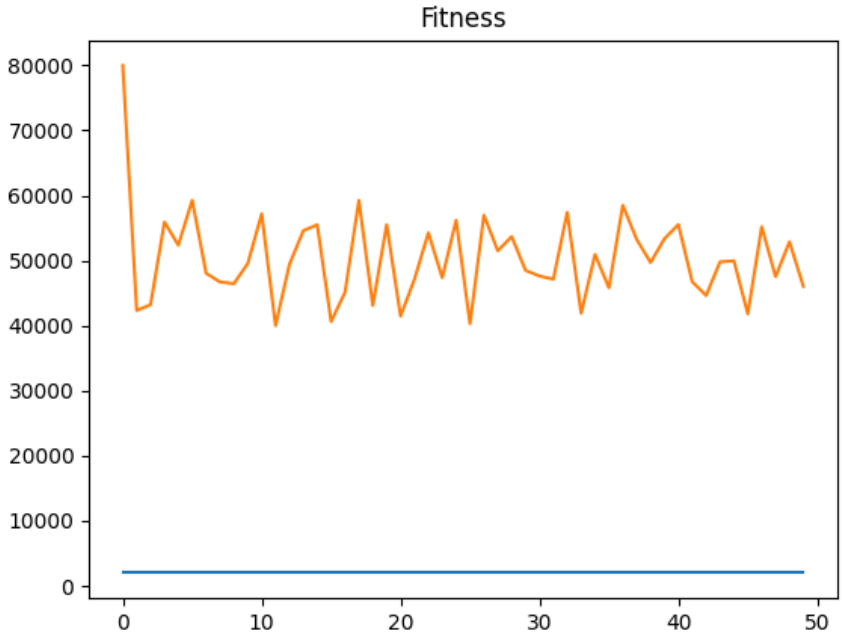
Fitness function:

```
def test_bool_5_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] else 1000
```

Best fitness: 99999

Best program:



Test: bool_5_OR

Problem not solved.

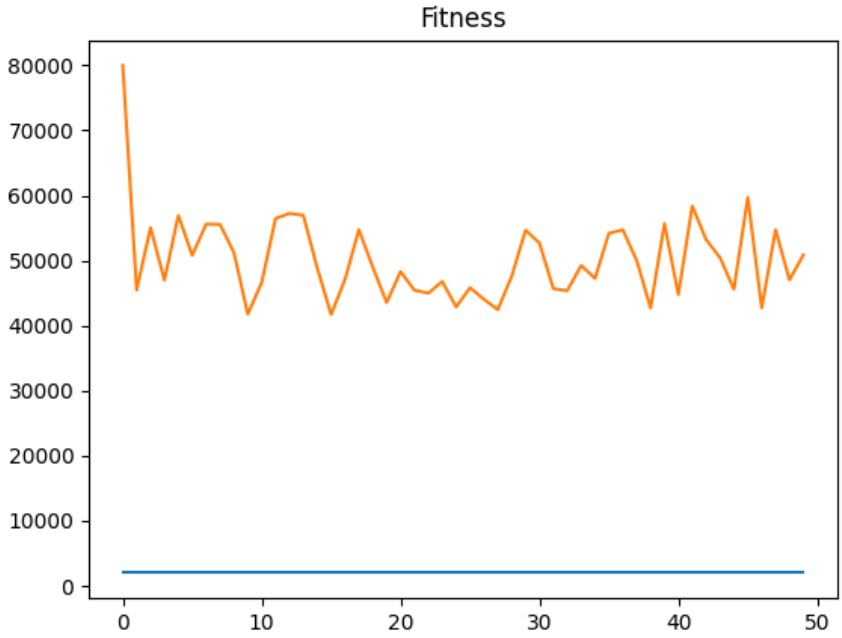
Fitness function:

```
def test_bool_5_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] else 1000
```

Best fitness: 99999

Best program:



Test: bool_5_XOR

Problem not solved.

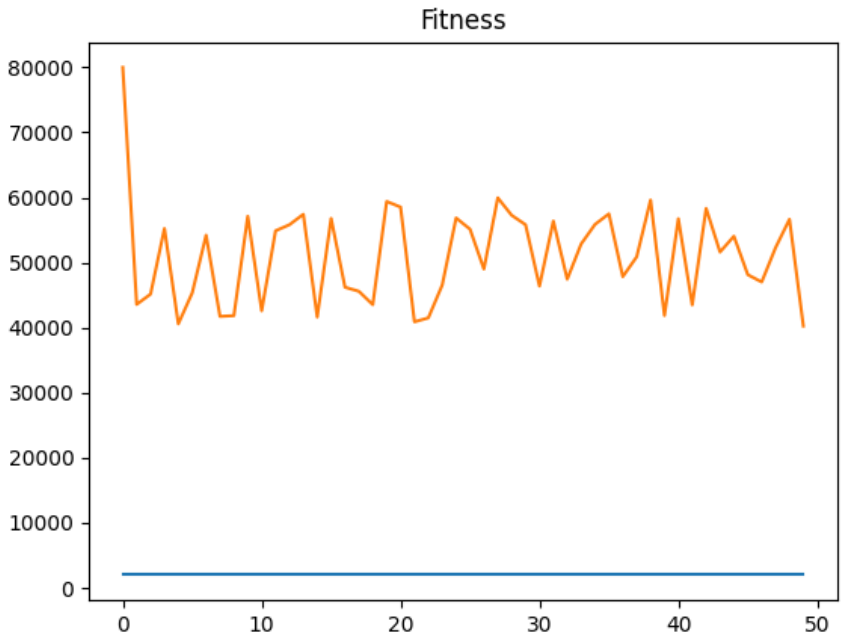
Fitness function:

```
def test_bool_5_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000

    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[3] ^ \
                y[4] == input_array[4] else 1000
```

Best fitness: 99999

Best program:



Test: bool_6_AND

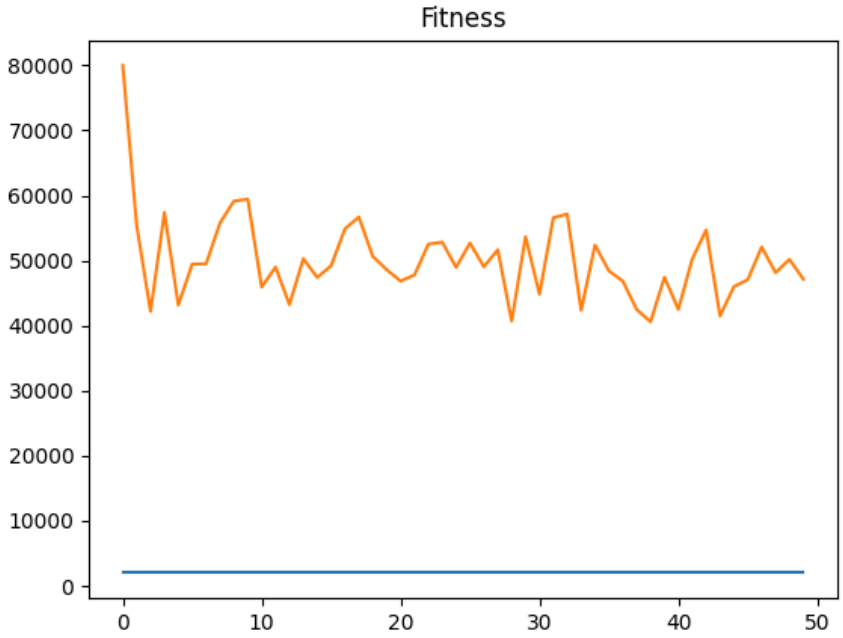
Problem not solved.

Fitness function:

```
def test_bool_6_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] and y[5] == input_array[5] else 1000
```

Best fitness: 99999

Best program:



Test: bool_6_OR

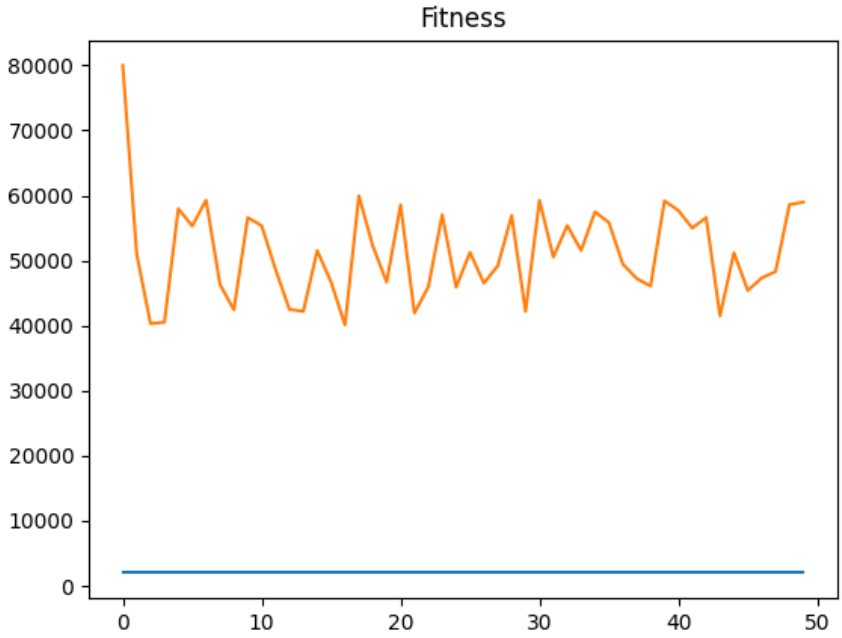
Problem not solved.

Fitness function:

```
def test_bool_6_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] or y[5] == input_array[5] else 1000
```

Best fitness: 99999

Best program:



Test: bool_6_XOR

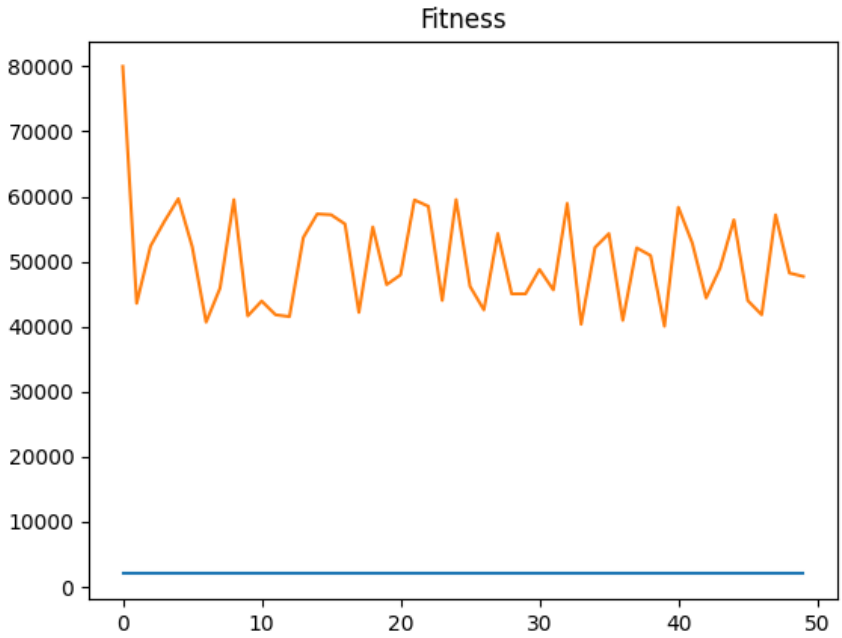
Problem not solved.

Fitness function:

```
def test_bool_6_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[3] ^ \
                y[4] == input_array[4] ^ y[5] == input_array[5] else 1000
```

Best fitness: 99999

Best program:



Test: bool_7_AND

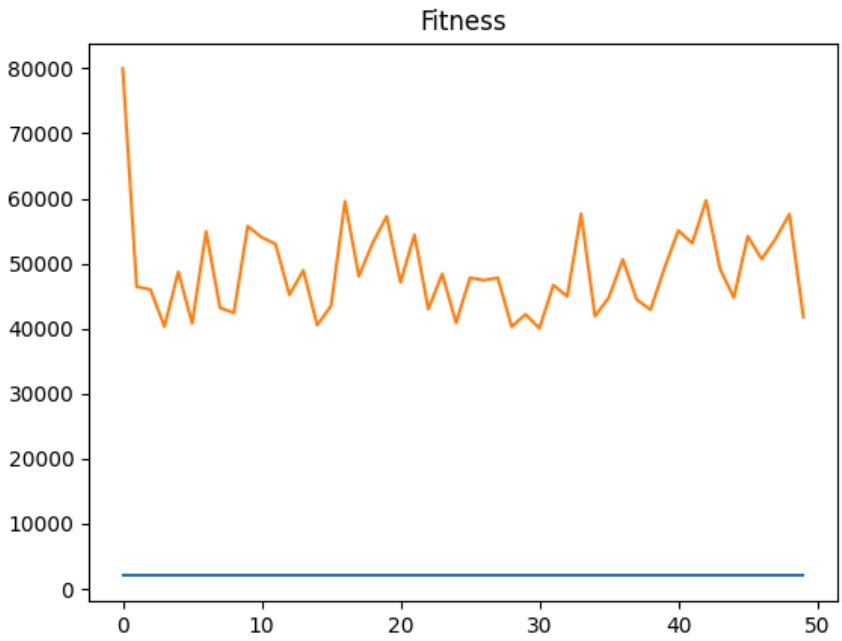
Problem not solved.

Fitness function:

```
def test_bool_7_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] and y[5] == input_array[5] and y[6] == input_array[
                    6] else 1000
```

Best fitness: 99999

Best program:



Test: bool_7_OR

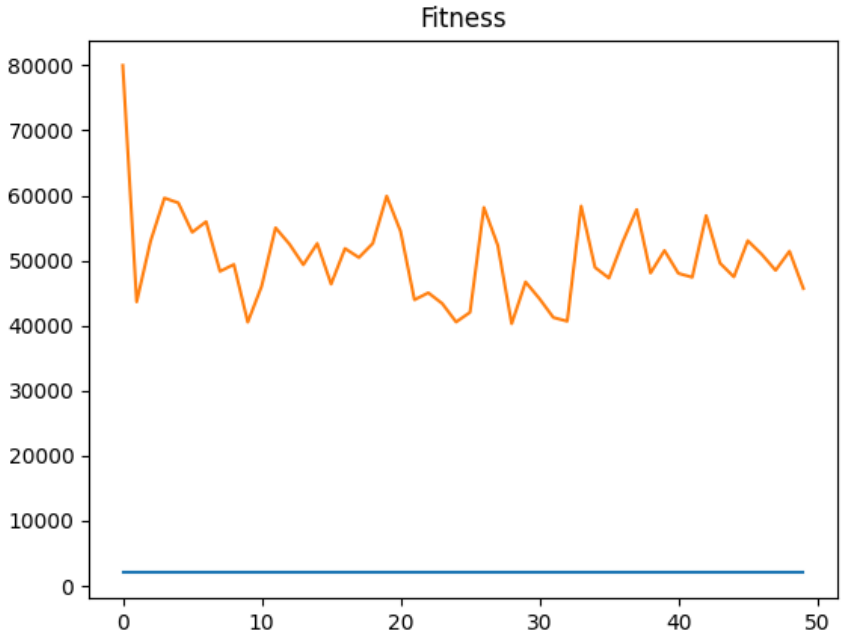
Problem not solved.

Fitness function:

```
def test_bool_7_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] or y[5] == input_array[5] or y[6] == input_array[6] else 1000
```

Best fitness: 99999

Best program:



Test: bool_7_XOR

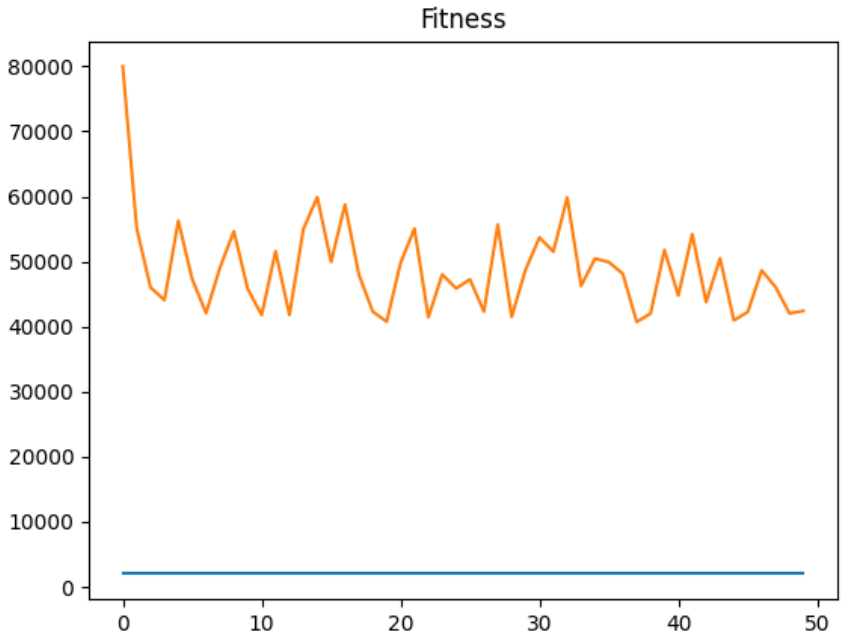
Problem not solved.

Fitness function:

```
def test_bool_7_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[3] ^ \
                y[4] == input_array[4] ^ y[5] == input_array[5] ^ y[6] == input_array[6] else 1000
```

Best fitness: 99999

Best program:



Test: bool_8_AND

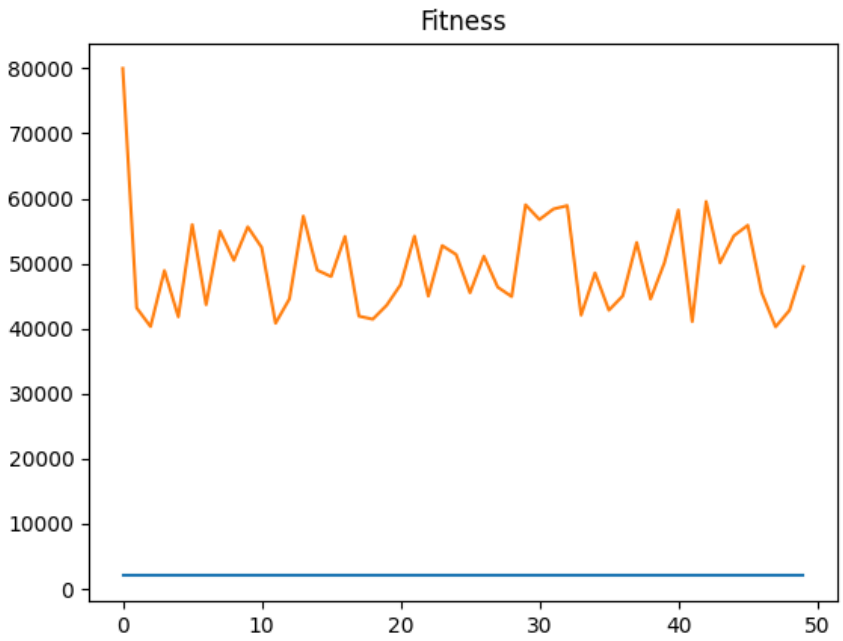
Problem not solved.

Fitness function:

```
def test_bool_8_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] and y[5] == input_array[5] and y[6] == input_array[6] and \
                y[7] == input_array[7] else 1000
```

Best fitness: 99999

Best program:



Test: bool_8_OR

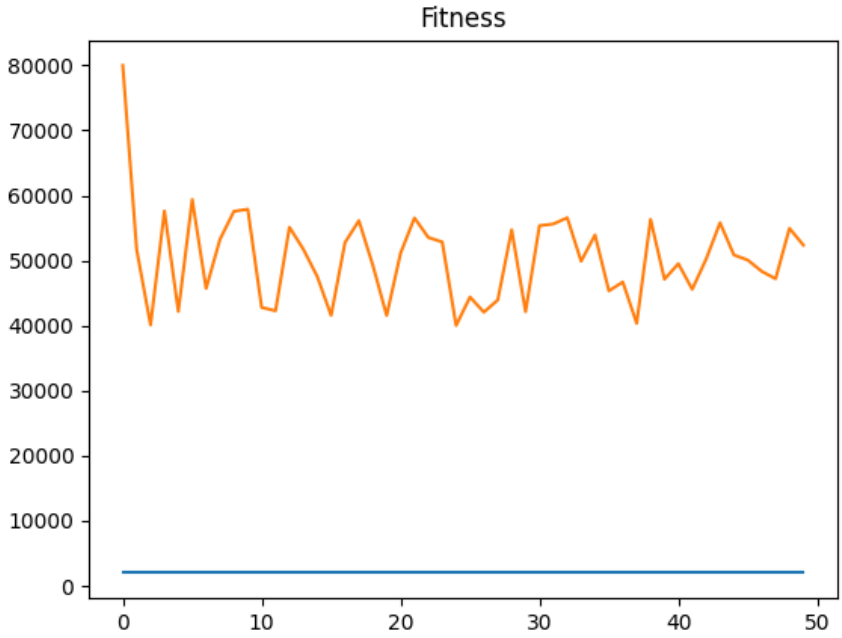
Problem not solved.

Fitness function:

```
def test_bool_8_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] or y[5] == input_array[5] or y[6] == input_array[6] or y[7] == input_array[
                    7] else 1000
```

Best fitness: 99999

Best program:



Test: bool_8_XOR

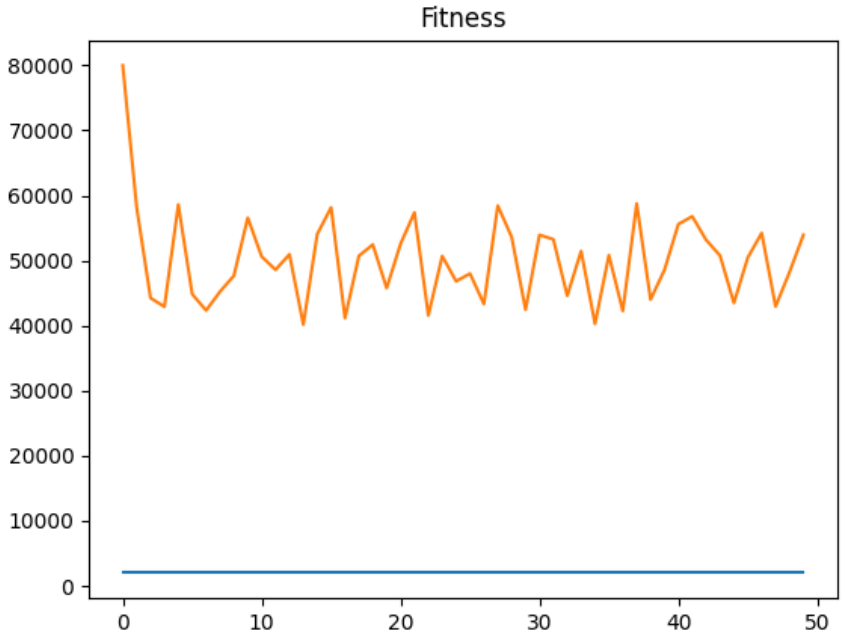
Problem not solved.

Fitness function:

```
def test_bool_8_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[3] ^ \
                y[4] == input_array[4] ^ y[5] == input_array[5] ^ y[6] == input_array[6] ^ y[7] == input_array[
                    7] else 1000
```

Best fitness: 99999

Best program:



Test: bool_9_AND

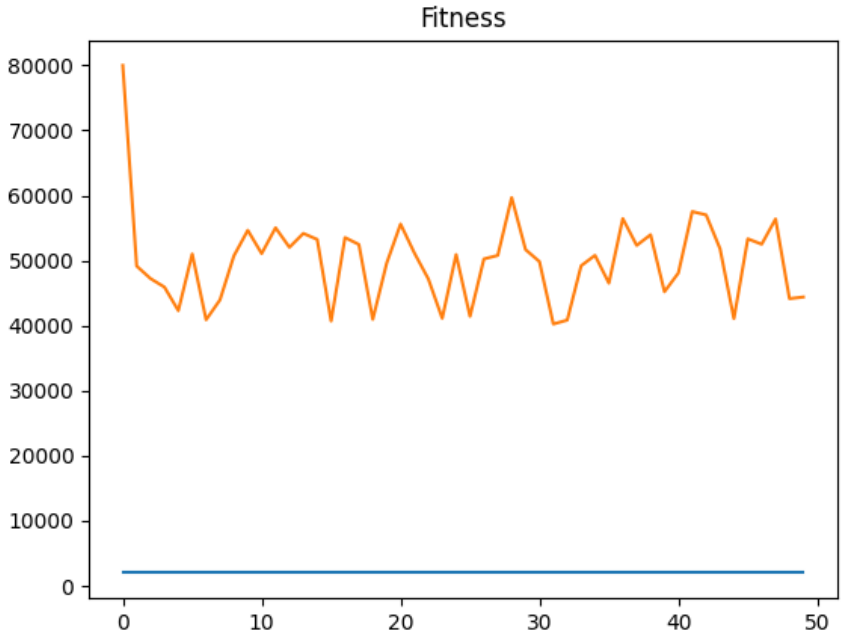
Problem not solved.

Fitness function:

```
def test_bool_9_AND_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] and y[1] == input_array[1] and y[2] == input_array[2] and y[3] == \
                input_array[3] and y[4] == input_array[4] and y[5] == input_array[5] and y[6] == input_array[6] and \
                y[7] == input_array[7] and y[8] == input_array[8] else 1000
```

Best fitness: 99999

Best program:



Test: bool_9_OR

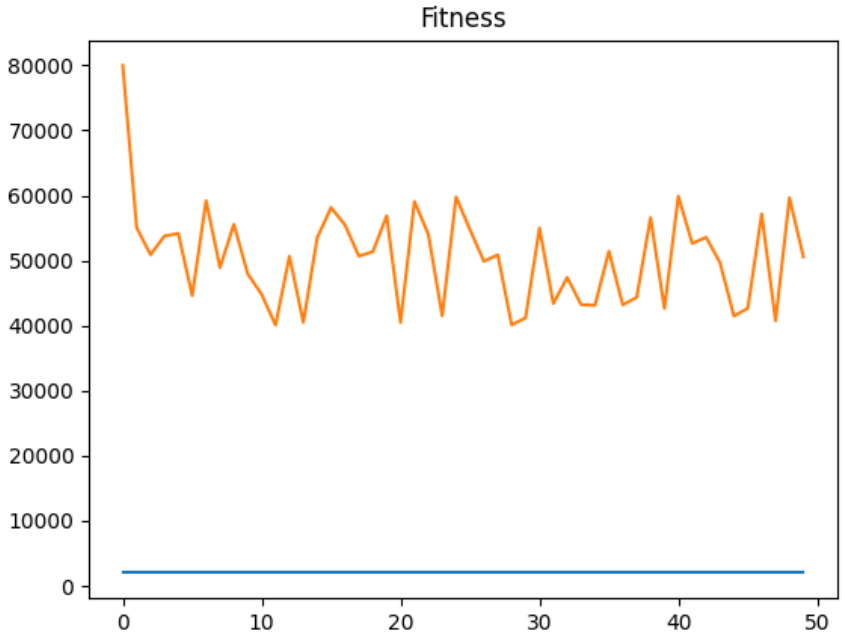
Problem not solved.

Fitness function:

```
def test_bool_9_OR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 0:
        return 100
    return 0 if y[0] == input_array[0] or y[1] == input_array[1] or y[2] == input_array[2] or y[3] == input_array[
        3] or y[4] == input_array[4] or y[5] == input_array[5] or y[6] == input_array[6] or y[7] == input_array[
                    7] or y[8] == input_array[8] else 1000
```

Best fitness: 99999

Best program:



Test: bool_9_XOR

Problem not solved.

Fitness function:

```
def test_bool_9_XOR_fitness(input_array, y):
    if len(y) > k:
        return len(y)
    if len(y) == 1:
        return 1000
    if len(y) == 0:
        return 1000
    return 0 if y[0] == input_array[0] ^ y[1] == input_array[1] ^ y[2] == input_array[2] ^ y[3] == input_array[3] ^ \
                y[4] == input_array[4] ^ y[5] == input_array[5] ^ y[6] == input_array[6] ^ y[7] == input_array[7] ^ \
                y[8] == input_array[8] else 1000
```

Best fitness: 99999

Best program: