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!

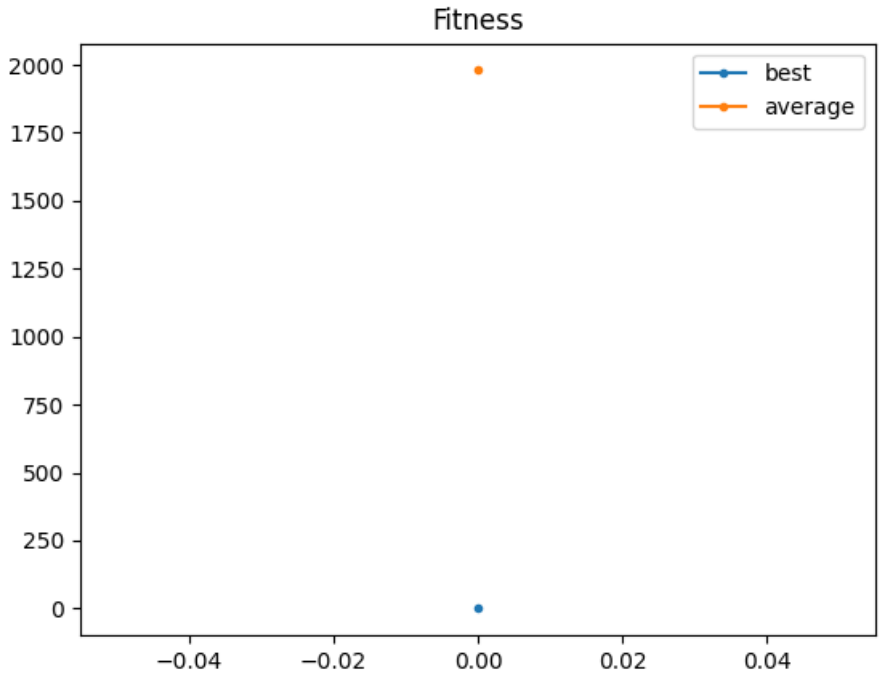
Fitness function:

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

Best fitness: 0

Best program:

```
{
 a = read;
 print 16;
 print 19;
 print 8;
 print 3;
 print 0;
 print 3;
 print 17;
 print 1;
}
```



Test: 11b

Problem not solved.

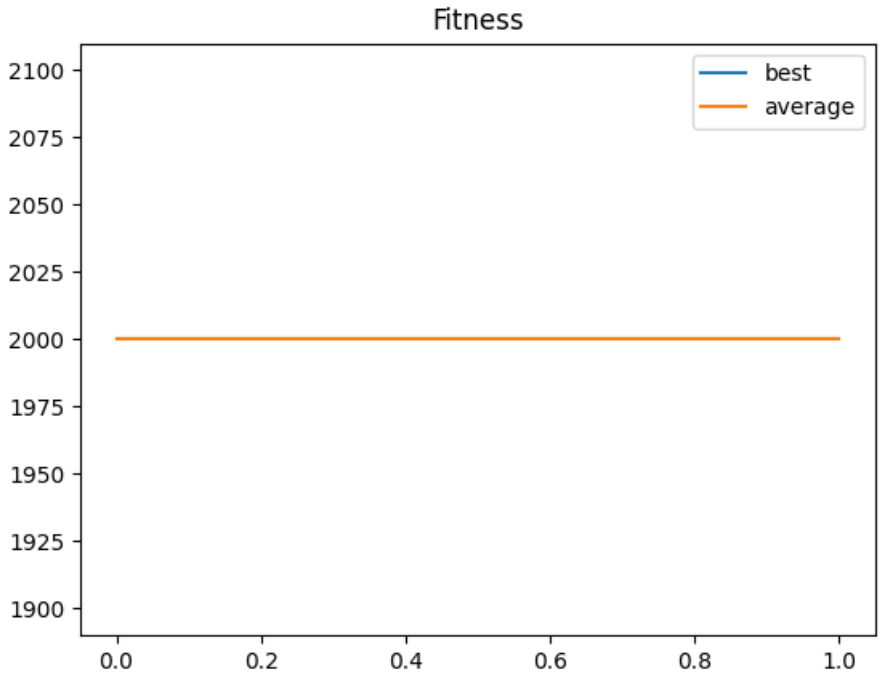
Fitness function:

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

Best fitness: 2000

Best program:

```
{
 r = read;
 print 15;
 print 0;
 print 8;
 print 15;
 print print p;;
 print r;
}
```



Test: 11c

Problem not solved.

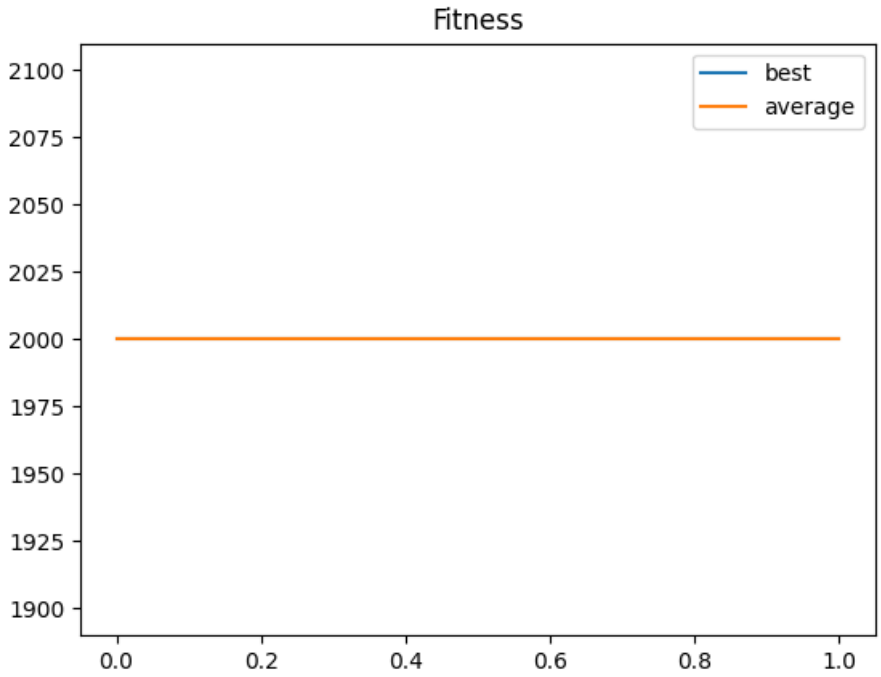
Fitness function:

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

Best fitness: 2000

Best program:

```
{
 x = read;
 f = read;
 print 20;
}
```



Test: 11d

Problem not solved.

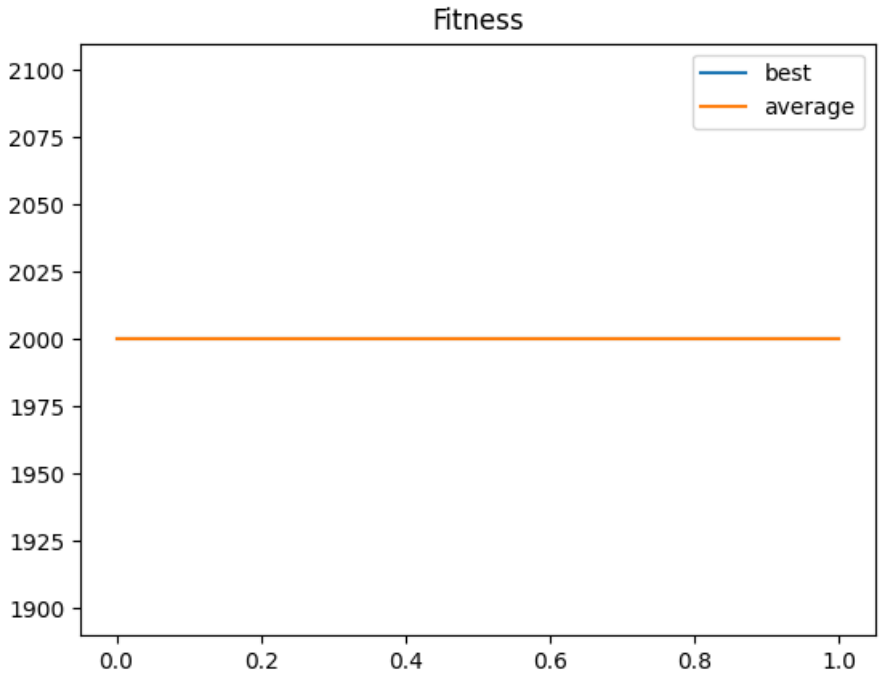
Fitness function:

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

Best fitness: 2000

Best program:

```
{
 p = read;
 print 7;
 print 3;
 print 6;
 print p;
 print p;
 p = p / p;
 print {
 v = read;
 l = read;
 print 10;
 print 17;
 print 11;
 print v;
 print 13;
 print 15;
};
 print 20;
 print 19;
}
```



Test: 11e

Problem not solved.

Fitness function:

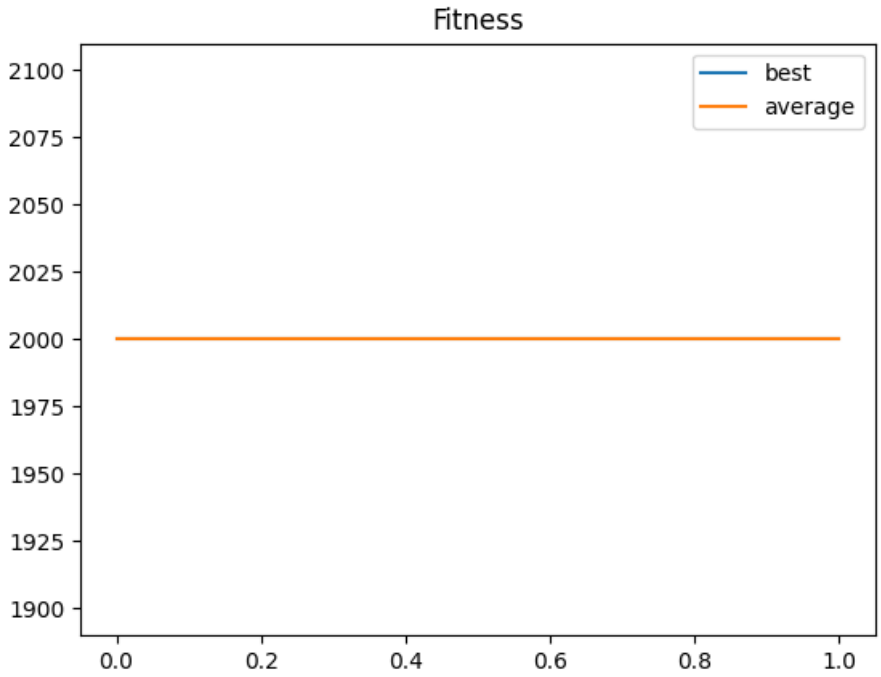
```
    def test_11e_fitness(input_array, y):
        if len(y) == 0:
            return 1000
        return 0 if y[0] == 789 else 1000
```

Best fitness: 2000

Best program:

```
{
 m = read;
 if m >= m {
 print 11;
 for m == m {
 print 16;
 print -1;
 print 20;
 print m;
 6 = m - m;
}
 if m < m {
 print m;
 if m == m {
 print 10;
 print 14;
 print 20;
 print m;
}
 print 19;
 print 5;
 print 1;
 print 4;
 print 9;
 p = read;
 print 6;
 print 17;
}
 print 6;
}
 print 7;
 if p >= m {
 for p < p {
 m = p / m;
```

```
 print 1;
 z = read;
 s = read;
 print -1;
 if z > s {
 print 18;
 b = read;
 print 1;
 print z;
 print 16;
 print 5;
 print 10;
 print 5;
}
 print 18;
 c = read;
 print 13;
 print 19;
}
 print -1;
 o = read;
 for c < b {
 print 3;
 r = read;
 print 15;
 print 1;
 print 2;
 print 10;
 print 18;
 e = read;
 print 12;
 print 20;
}
 print 20;
 print 15;
 print 4;
 for z <= z {
 for o < b {
 print 10;
 print 15;
 print z;
 print 7;
 print 13;
}
 print 3;
 print r;
 if e > r {
 print 0;
 print 1;
 print 4;
 print 3;
 print 11;
}
 print 12;
}
 print 16;
 print 17;
}
 print 0;
 d = read;
 print 10;
 print 20;
}
```



Test: 11f

Problem not solved.

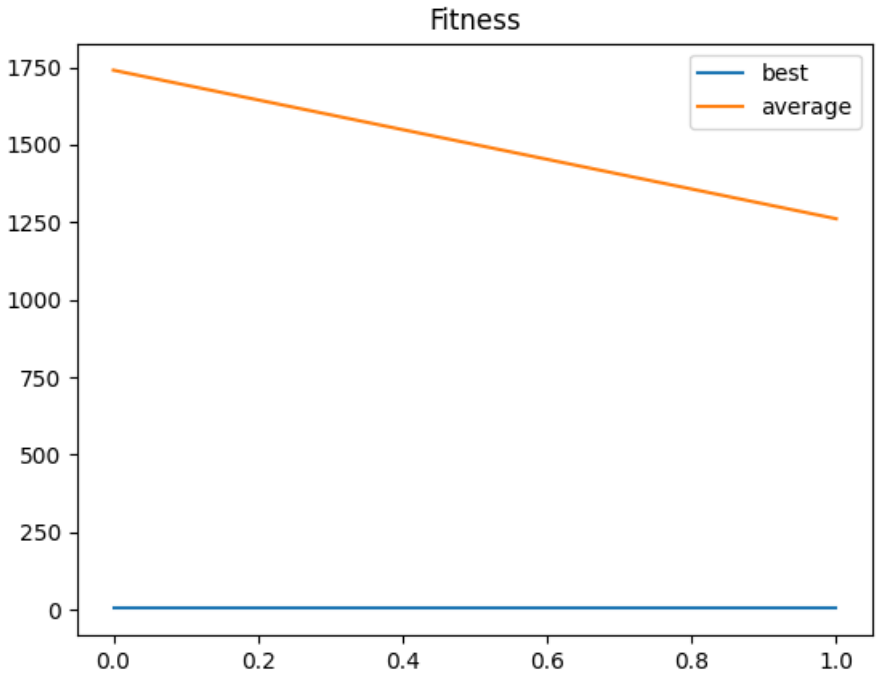
Fitness function:

```
    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: 4

Best program:

```
{
 p = read;
 print 5;
 k = read;
 print 17;
}
```



Test: 12a

Problem not solved.

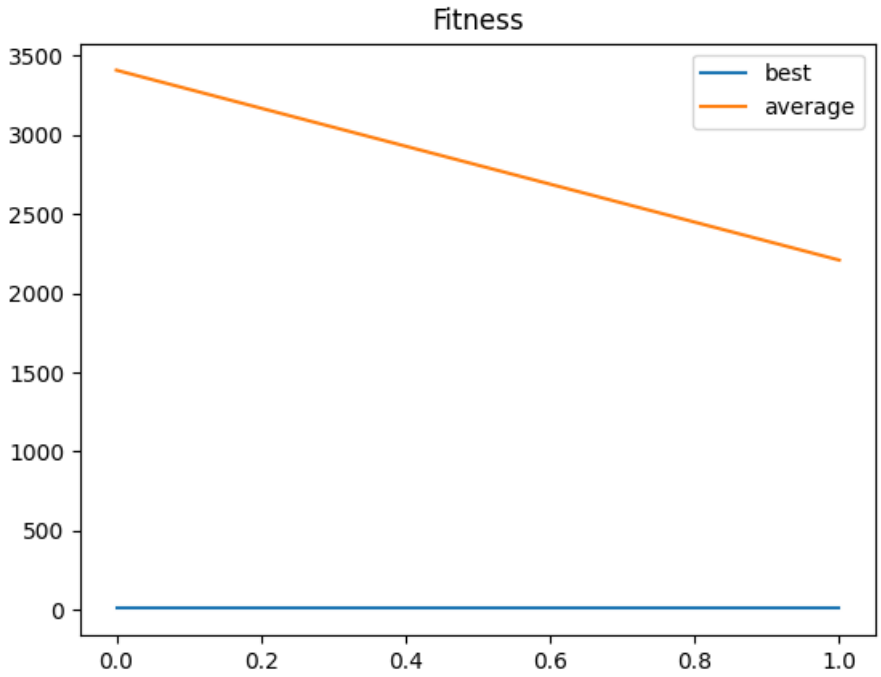
Fitness function:

```
    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:

```
{
 u = read;
 print 2;
 print 15;
}
```



Test: 12b

Problem not solved.

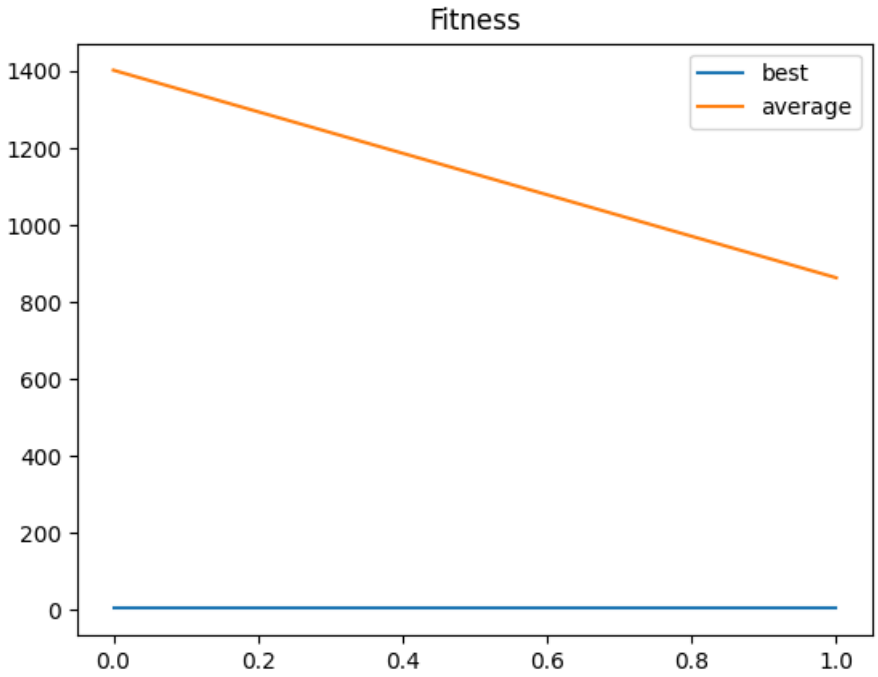
Fitness function:

```
    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: 4.0

Best program:

```
{
 e = read;
 print 5;
}
```



Test: 12c

Problem not solved.

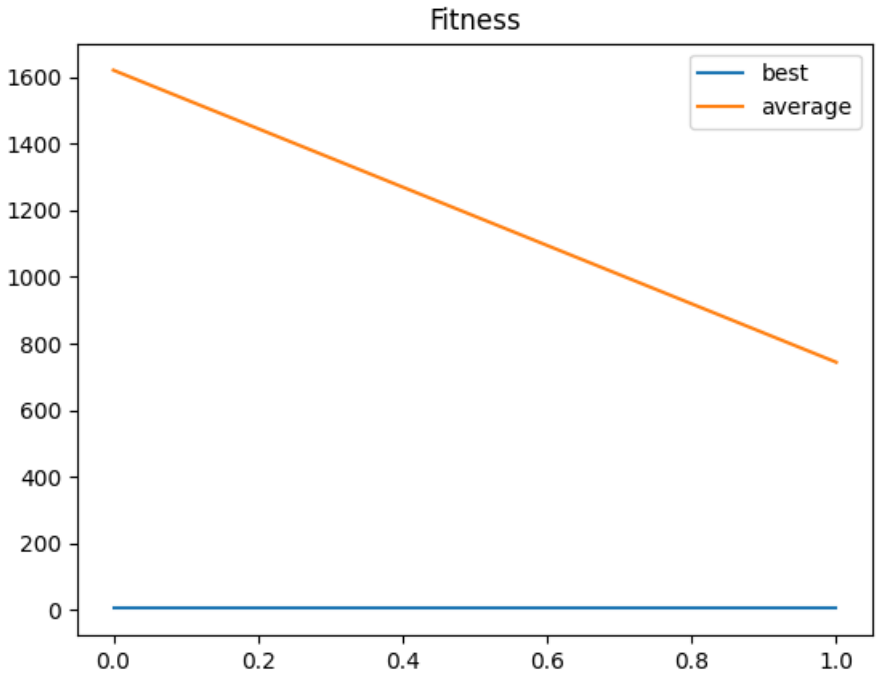
Fitness function:

```
    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

Best program:

```
{
 c = read;
 print -1;
 print 9;
}
```



Test: 12d

Problem not solved.

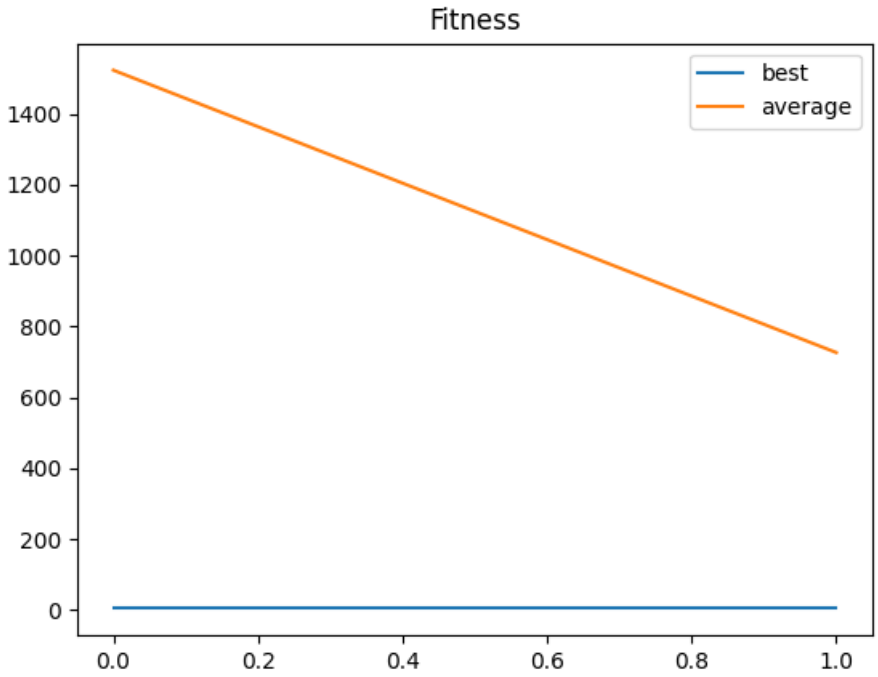
Fitness function:

```
    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: 4

Best program:

```
{
 c = read;
 print 11;
 print 8;
}
```



Test: 12e

Problem not solved.

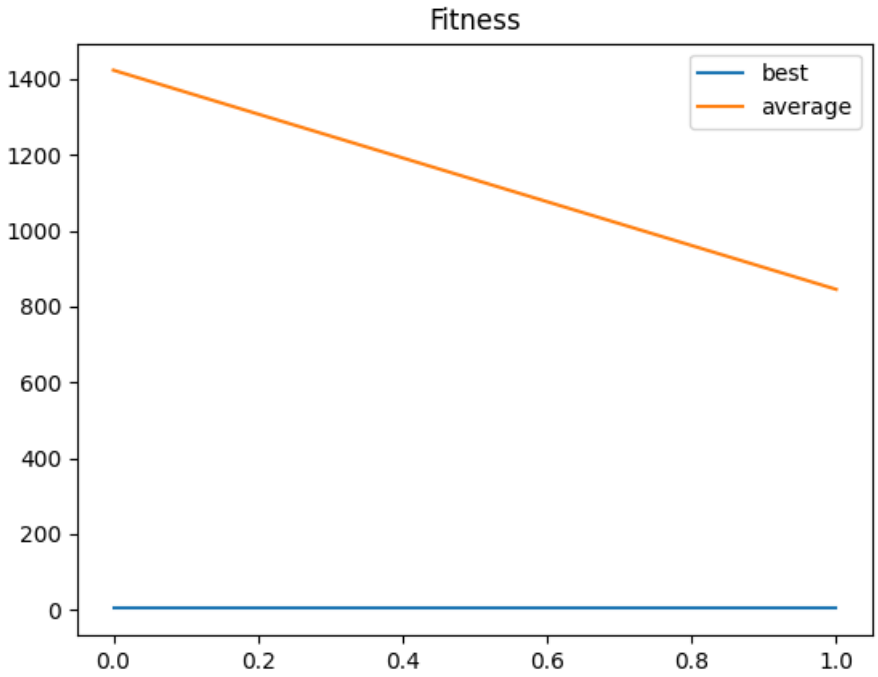
Fitness function:

```
    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:

```
{
 v = read;
 print 19;
 print 3;
}
```



Test: 13a

Problem not solved.

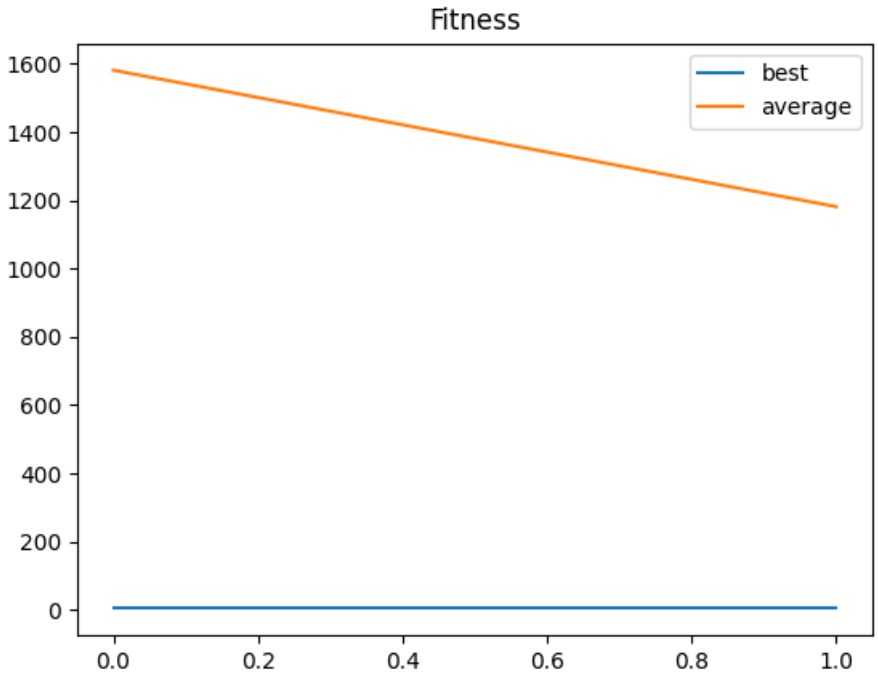
Fitness function:

```
    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:

```
{
 i = read;
 print 9;
 print 12;
}
```



Test: 13b

Problem not solved.

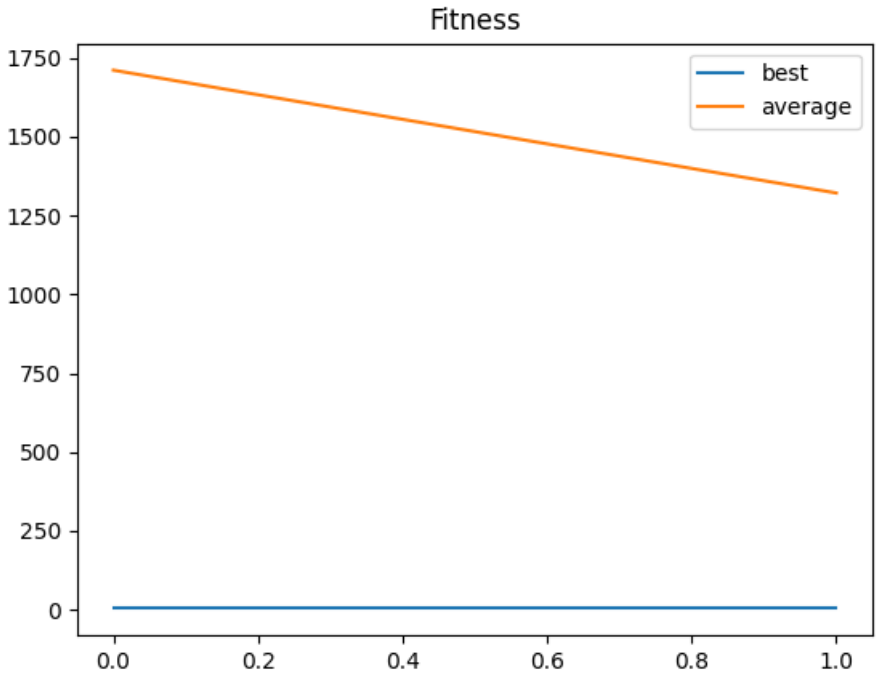
Fitness function:

```
    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:

```
{
 y = read;
 print 17;
 print -1;
}
```



Test: 14a

Problem not solved.

Fitness function:

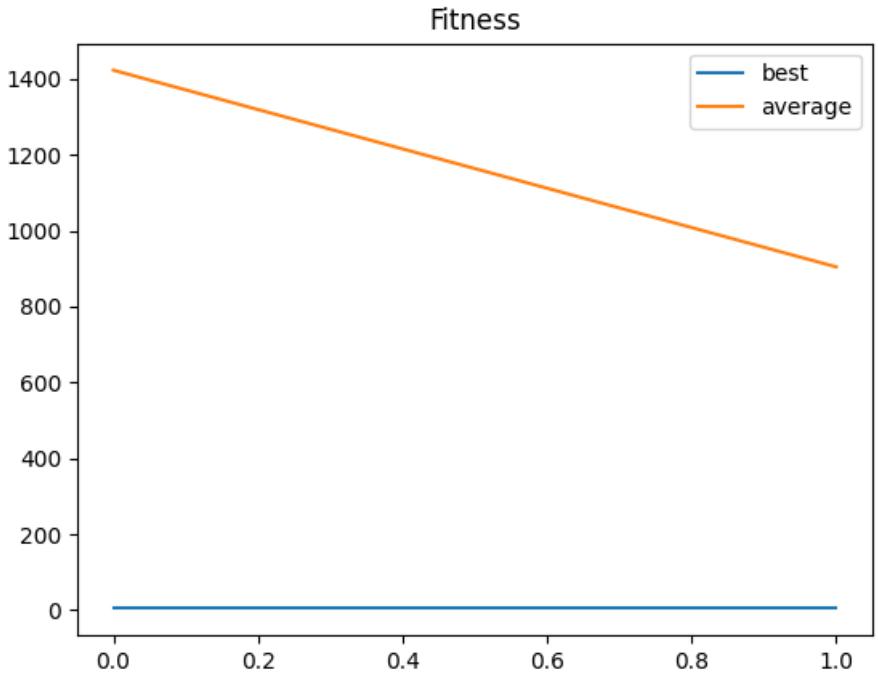
```
    def test_14a_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

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

Best fitness: 4

Best program:

```
{
 i = read;
 print 13;
 print 11;
}
```



Test: 14b

Problem not solved.

Fitness function:

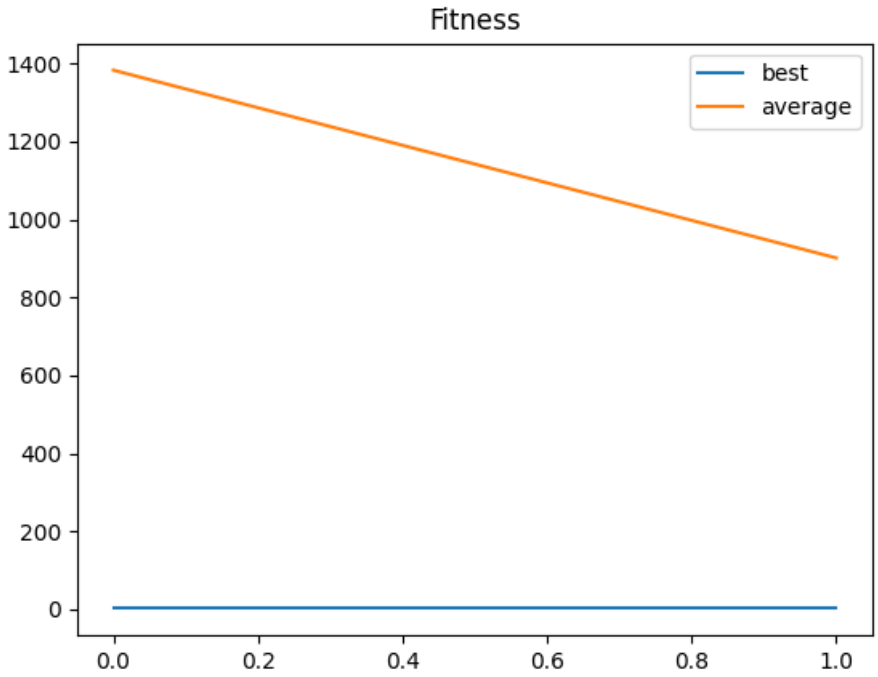
```
    def test_14b_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

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

Best fitness: 2.0

Best program:

```
{
 s = read;
 print 4;
 {
 j = read;
 print 4;
}
}
```



Test: bool_1_AND

Problem not solved.

Fitness function:

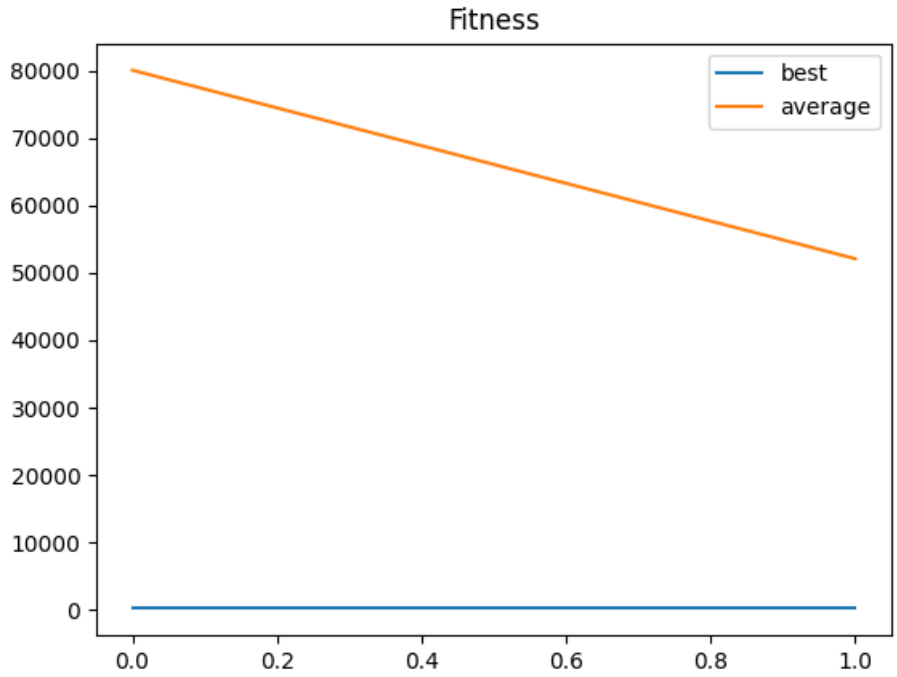
```
    def test_bool_1_AND_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

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

Best fitness: 200

Best program:

```
{
 e = read;
 print 7;
 s = read;
 print 5;
}
```



Test: bool_1_OR

Problem not solved.

Fitness function:

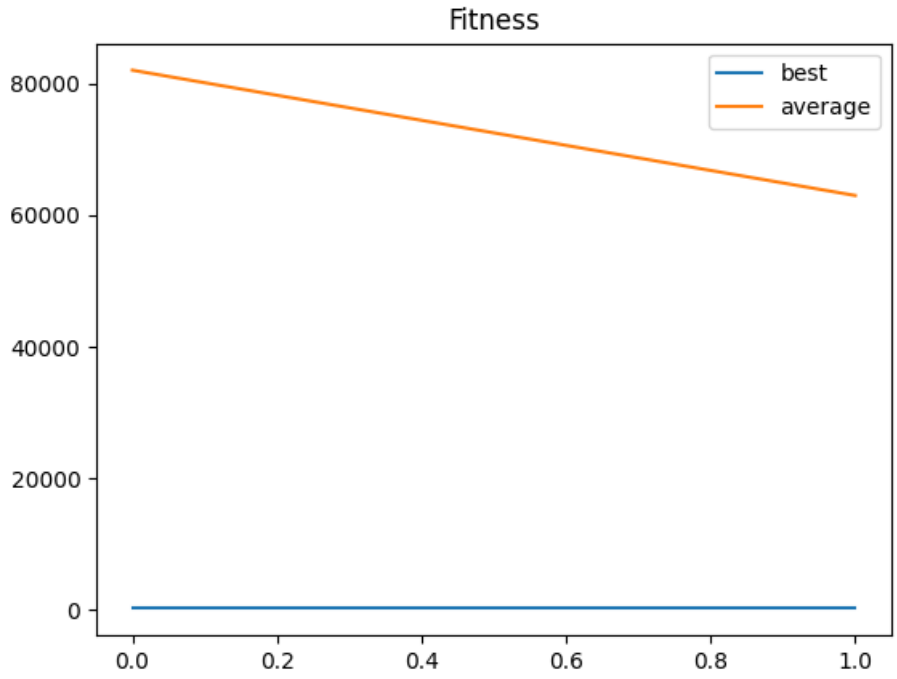
```
    def test_bool_1_OR_fitness(input_array, y):
        if len(y) > 1:
            return len(y)
        if len(y) == 0:
            return 1000

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

Best fitness: 200

Best program:

```
{
 m = read;
 print -1;
 print 2;
}
```


Fitness
best
average
80000
60000
40000
20000
0
0.0
0.2
0.4
0.6
0.8
1.0