Table of Contents

```
function [min_yVal] = PS08_simplex_koike()

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% ENGR 132
% Program Description
% This program is designed to execute the simplex method which finds
% a minimum value for an unknown function with certain conditions.
%
% Function Call
% PS08_simplex_koike(x_value)
%
% Input Arguments
% 1. funX: y = f(x) which is the fucntion we are going to probe
%    to find the min
% 2. x1: the lower limit of the interval including the minimum value
%    of the function
% 3. x4: the higher limit of the interval including the minimum value
%    of the function
%
% Output Arguments
% 1. min_yVal: the minimum y-value of a function of x
%
% Assignment Information
%   Assignment:        PS 08, Problem 3
%   Author:             Tomoki Koike, koike@purdue.edu
%   Team ID:            002-08
%   Contributor:   Name, login@purdue [repeat for each]
%   My contributor(s) helped me:
%     [ ] understand the assignment expectations without
%         telling me how they will approach it.
%     [ ] understand different ways to think about a solution
%         without helping me plan my solution.
%     [ ] think through the meaning of a specific error or
%         bug present in my code without looking at my code.
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

INITIALIZATION

```
% the nameArray for the academic integrity statement
nameArray = "Tomoki Koike";
```

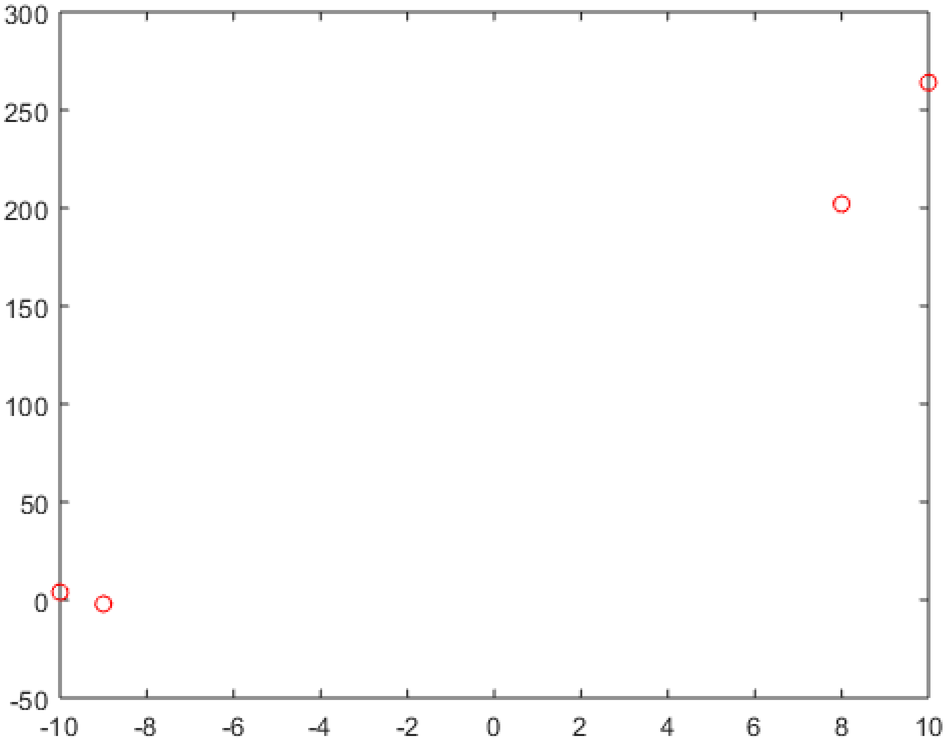
CALCULATIONS

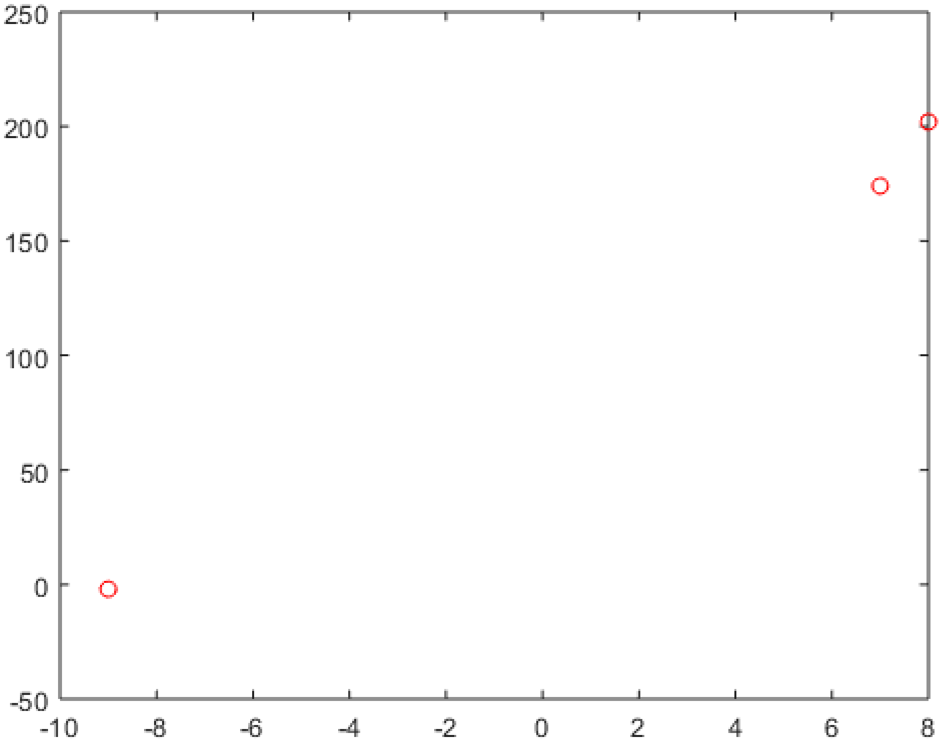
```
% % for this program we assume that the minimum value of the known
% % function is in the interval of [-10,10]
%
% %start
% n = 1;  %the counter (nth-term)
% x1= -10;
% x4 = 10;
% x2 = -9;
% x3 = 9;
% while (x4 - x1)>0.0001
%     %finding the corresponding y-values for the x-values
%     y1 = PS08_sub_function(x1);
%     y4 = PS08_sub_function(x4);
%     y2 = PS08_sub_function(x2);
%     y3= PS08_sub_function(x3);
%     if y2 >= y3
%         x2 = x2 + 1;
%     else
%         x3 = x3 - 1;
%     end
%     %redefining the y-values for the new x2 and x3 values
%     y2 = PS08_sub_function(x2);
%     y3 = PS08_sub_function(x3);
%     %creating the guessed points to plot a graph
%     x = [x1 x2 x3 x4];
%     y = [y1 y2 y3 y4];
%     %plotting a graph for the guessed points
%     figure
%     plot(x, y, "or");
%     %assigning x2 as the new x1 or x3 as the new x4
%     x1 = x2;
%     x4 = x3;
%     %printing out the iterations and the difference of x4 and x1
%     fprintf("iteration %d\n", n);
%     fprintf("x4 - x1 = %f\n", x4-x1);
%     %incrementing the counter
%     n = n + 1;
% end
% fprintf("the location of the minimum value is (%.4f, %.4f)\n\n", x1,
 y2);
```

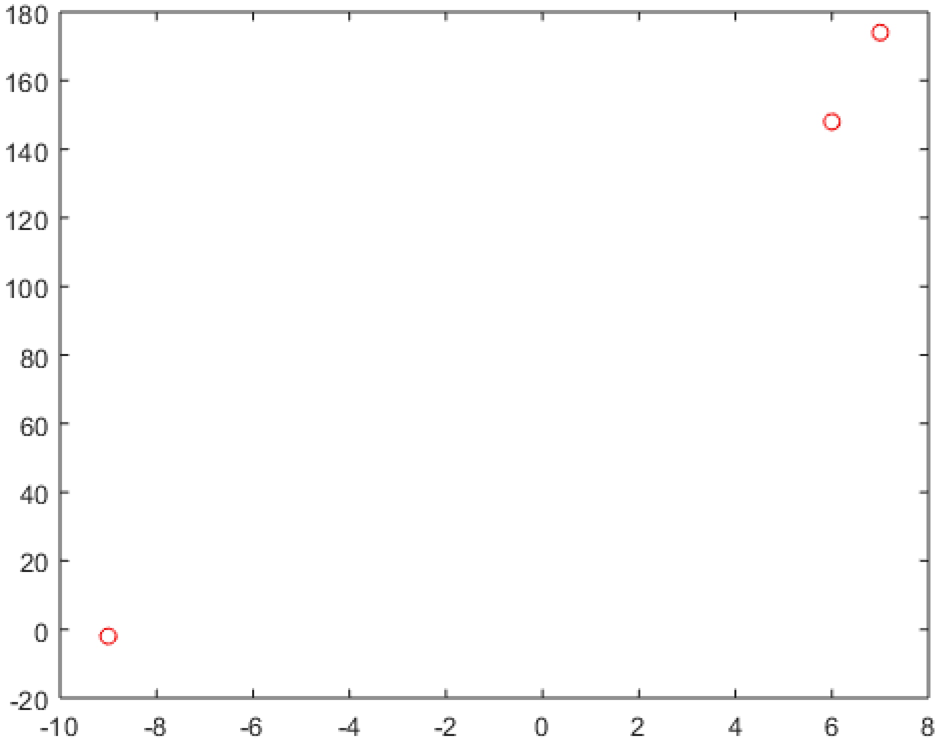
```
% min_yVal = y2;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% for this program we assume that the minimum value of the unknown
% function in PS08_funceval.p
% function is in the interval of [-10,10]

%start
n = 1;  %the counter (nth-term)
x1= -10;
x4 = 10;
x2 = -9;
x3 = 9;
while (x4 - x1)>0.0001
    %finding the corresponding y-values for the x-values
    y1 = PS08_funceval(x1);
    y4 = PS08_funceval(x4);
    y2 = PS08_funceval(x2);
    y3 = PS08_funceval(x3);
    if y2 >= y3
        x2 = x2 + 1;
    else
        x3 = x3 - 1;
    end
    %redefining the y-values for the new x2 and x3 values
    y2 = PS08_funceval(x2);
    y3 = PS08_funceval(x3);
    %creating the guessed points to plot a graph
    x = [x1 x2 x3 x4];
    y = [y1 y2 y3 y4];
    %plotting a graph for the guessed points
    figure
    plot(x, y, "or");
    %assigning x2 as the new x1 or x3 as the new x4
    x1 = x2;
    x4 = x3;
    %printing out the iterations and the difference of x4 and x1
    fprintf("iteration %d\n", n);
    fprintf("x4 - x1 = %f\n", x4-x1);
    %incrementing the counter
    n = n + 1;
end
fprintf("the location of the minimum value is (%.4f, %.4f)\n\n", x1,
 y2);
min_yVal = y2;
```

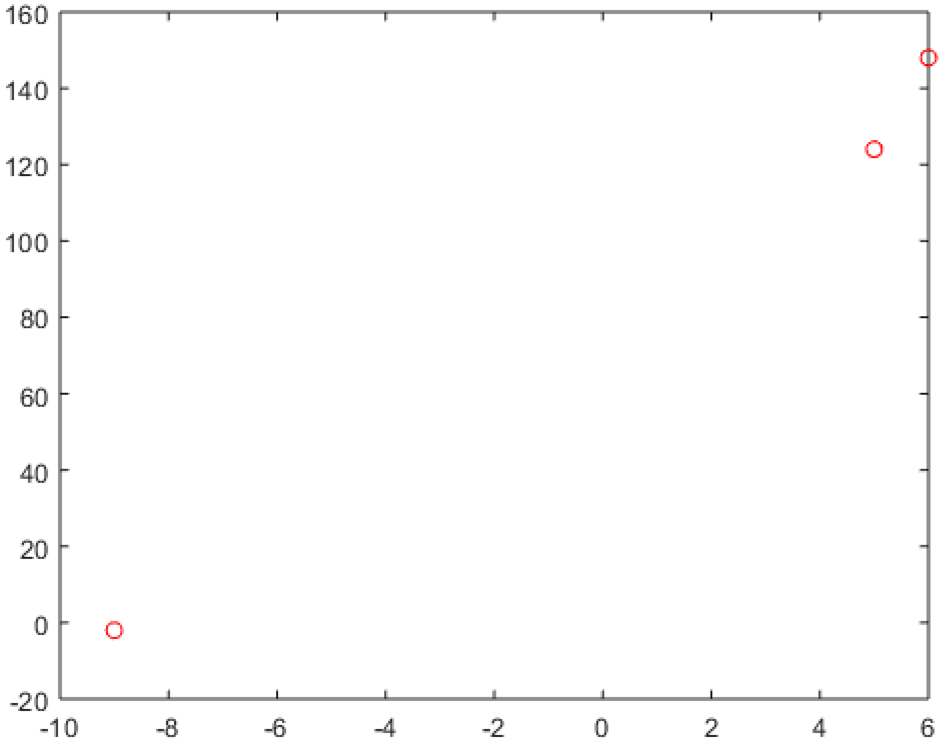
```
iteration 1
x4 - x1 = 17.000000
iteration 2
x4 - x1 = 16.000000
iteration 3
x4 - x1 = 15.000000
iteration 4
x4 - x1 = 14.000000
```

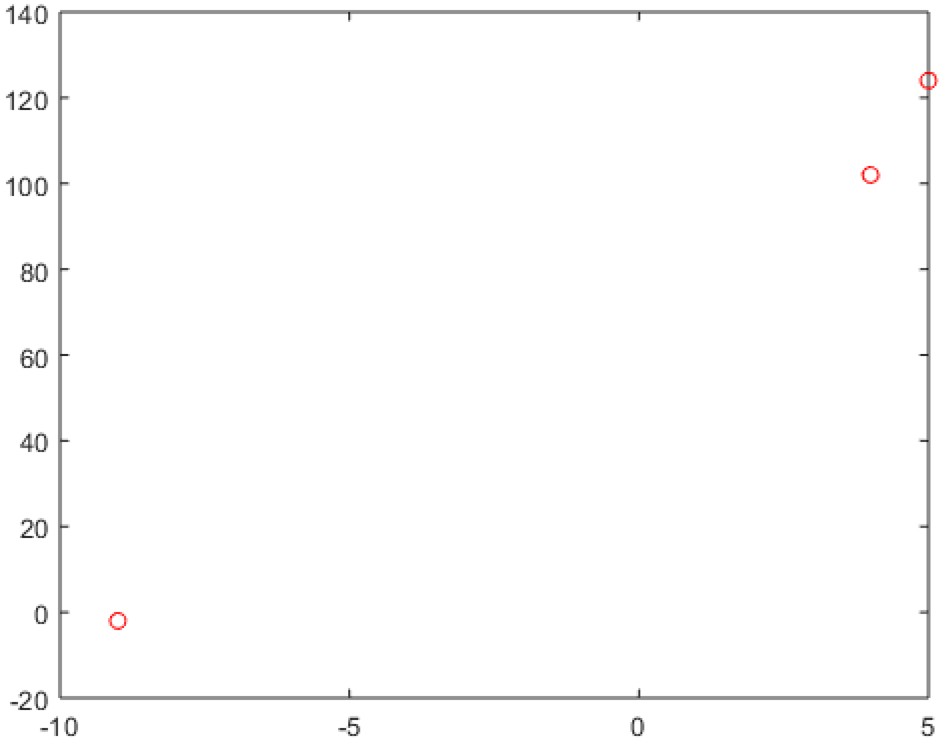
```
iteration 5
x4 - x1 = 13.000000
iteration 6
x4 - x1 = 12.000000
iteration 7
x4 - x1 = 11.000000
iteration 8
x4 - x1 = 10.000000
iteration 9
x4 - x1 = 9.000000
iteration 10
x4 - x1 = 8.000000
iteration 11
x4 - x1 = 7.000000
iteration 12
x4 - x1 = 6.000000
iteration 13
x4 - x1 = 5.000000
iteration 14
x4 - x1 = 4.000000
iteration 15
x4 - x1 = 3.000000
iteration 16
x4 - x1 = 2.000000
iteration 17
x4 - x1 = 1.000000
iteration 18
x4 - x1 = 0.000000
the location of the minimum value is (-6.0000, -8.0000)
```

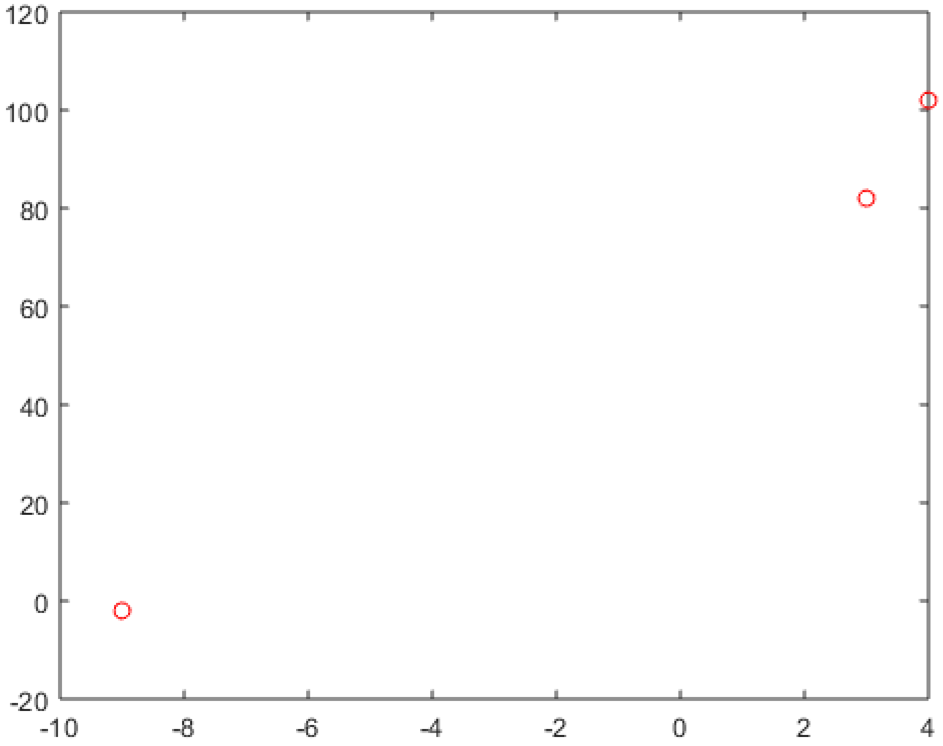

300
250
200
150
100
50
0
-50
-10
-8
-6
-4
-2
0
2
4
6
8
10

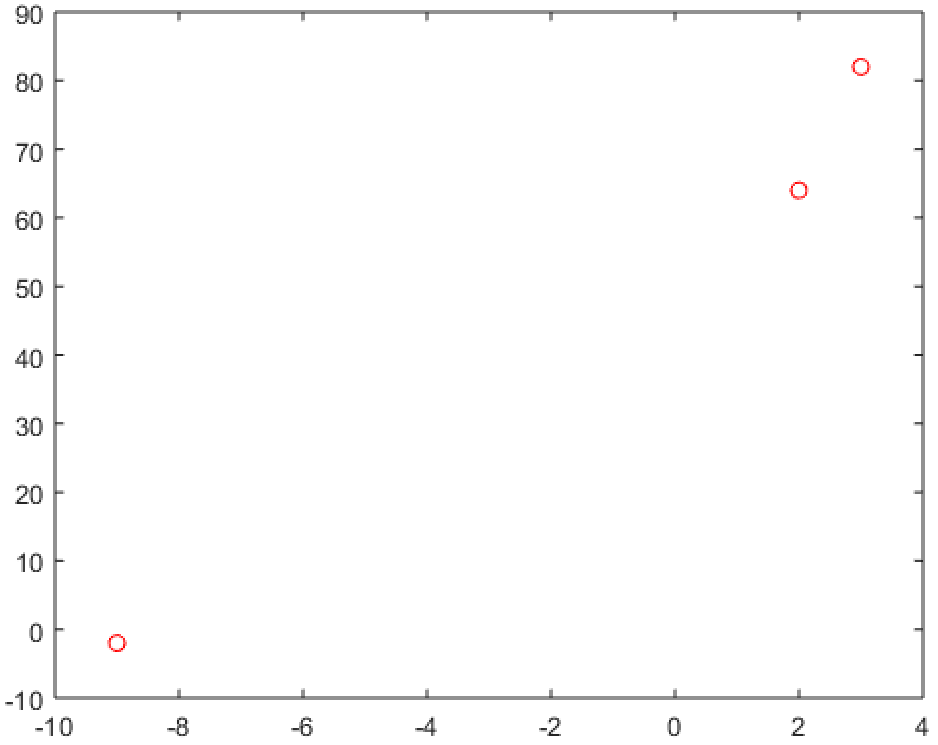

250
200
150
100
50
0
-50
-10
-8
-6
-4
-2
0
2
4
6
8

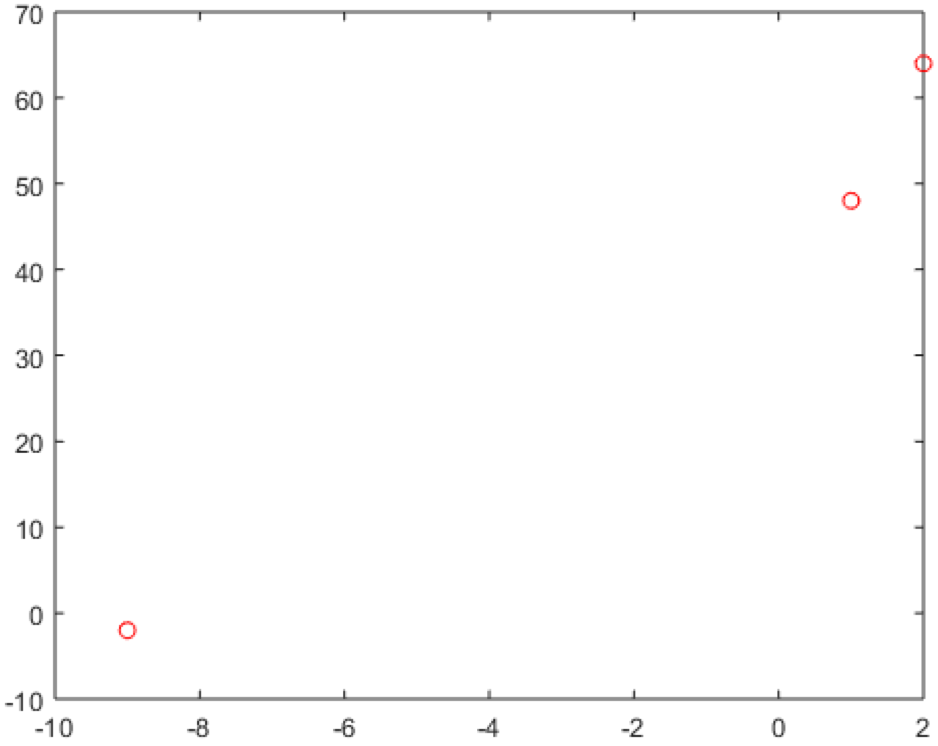

180
160
140
120
100
80
60
40
20
0
-20
-10
-8
-6
-4
-2
0
2
4
6
8

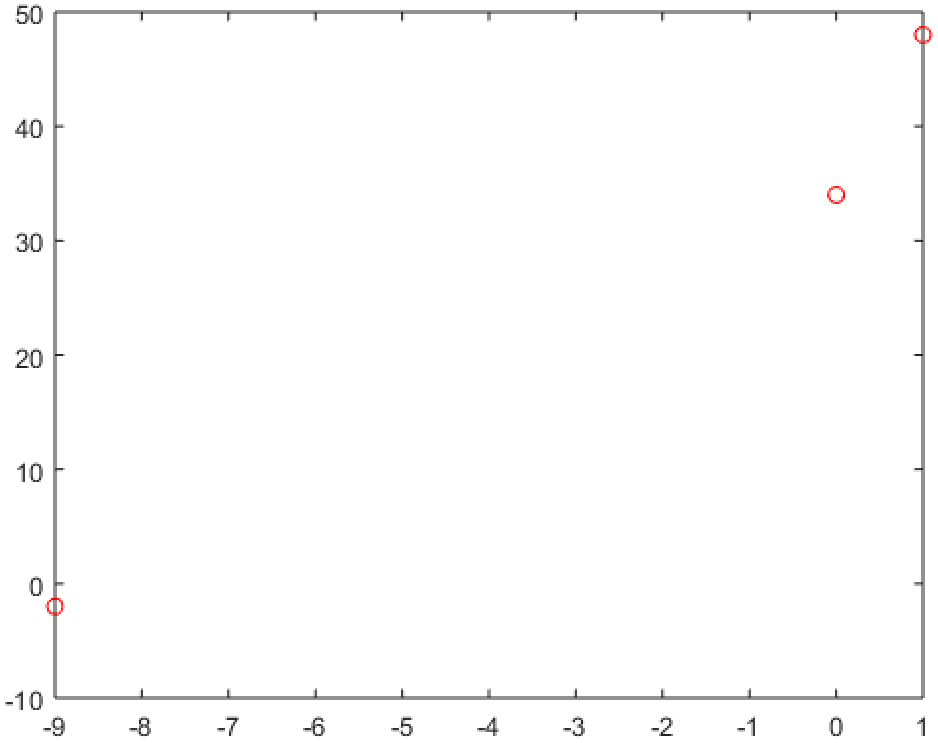

160
140
120
100
80
60
40
20
0
-20
-10
-8
-6
-4
-2
0
2
4
6

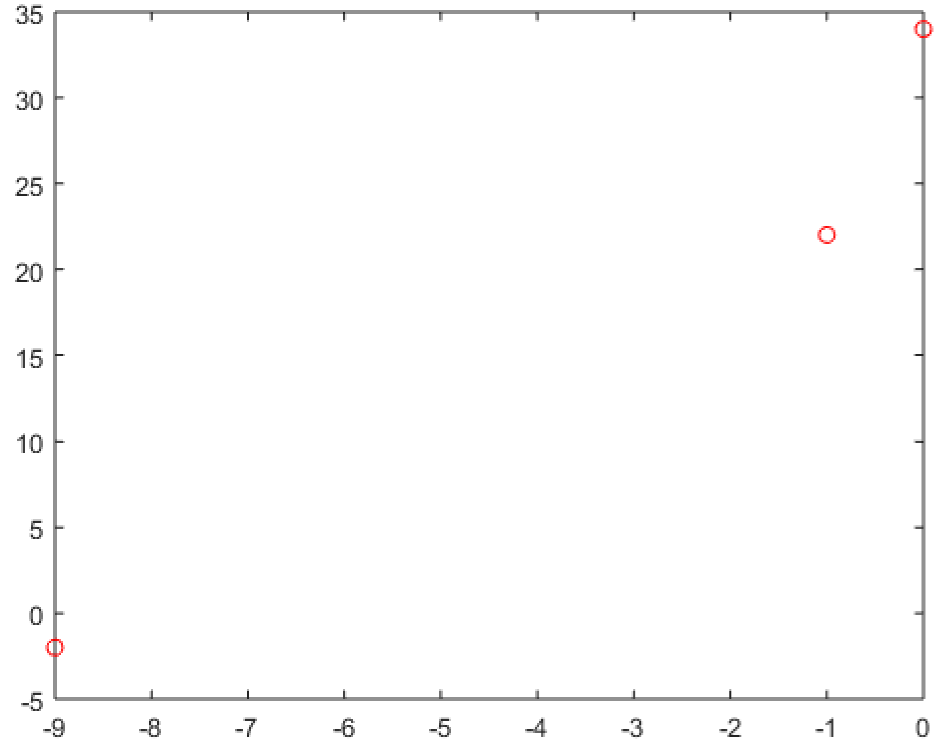

140
120
100
80
60
40
20
0
-20
-10
-5
0
5

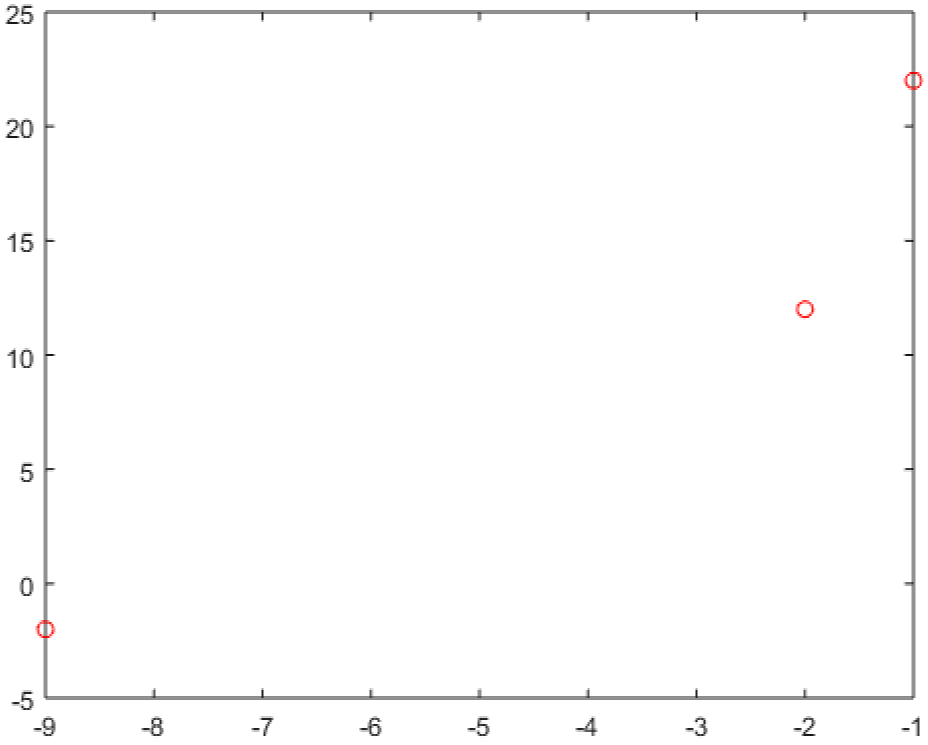

120
100
80
60
40
20
0
-20
-10
-8
-6
-4
-2
0
2
4

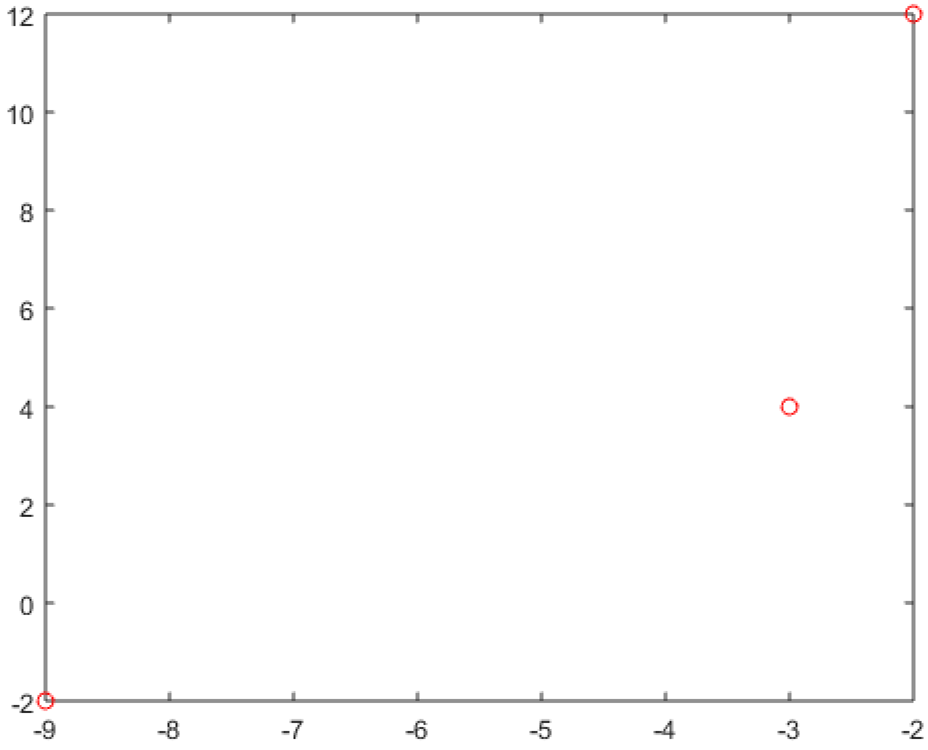

90
80
70
60
50
40
30
20
10
0
-10
-10
-8
-6
-4
-2
0
2
4

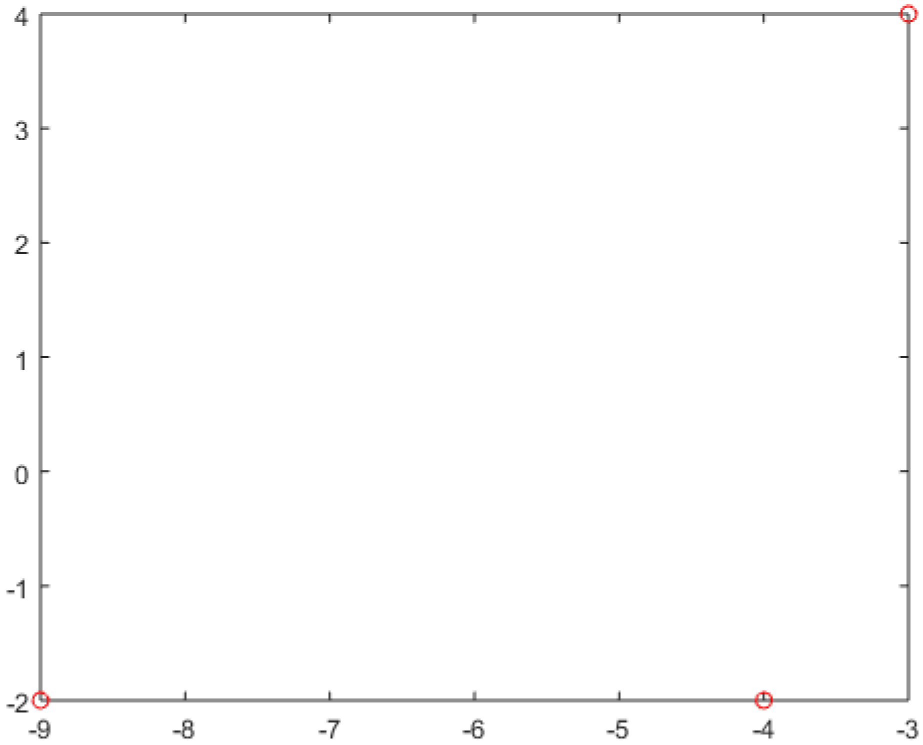

70
60
50
40
30
20
10
0
-10
-10
-8
-6
-4
-2
0
2

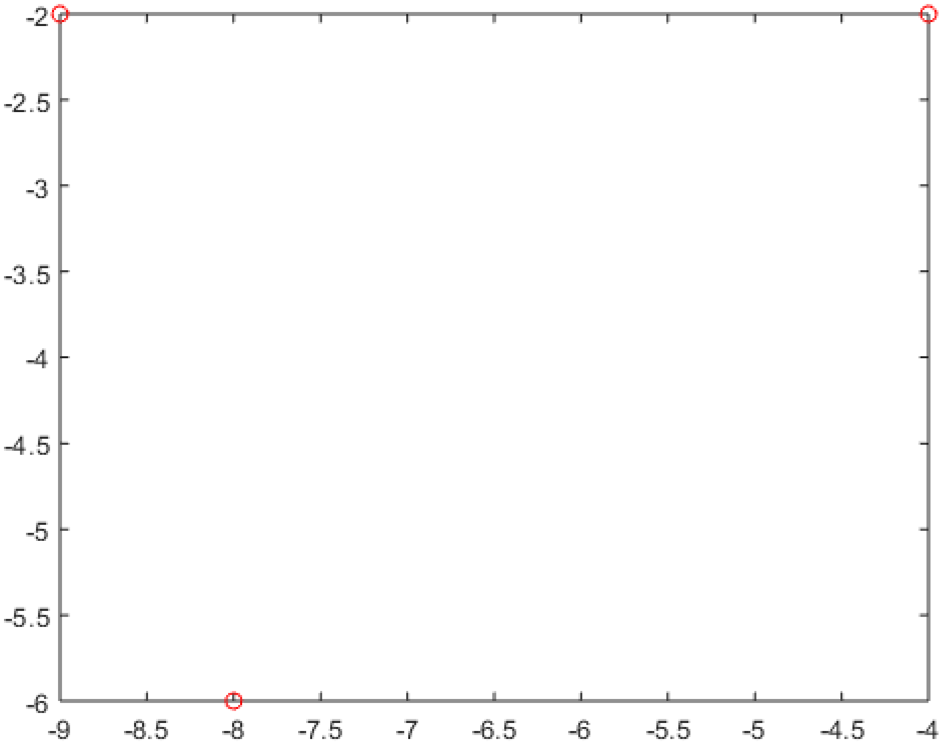

50
40
30
20
10
0
-10
-9
-8
-7
-6
-5
-4
-3
-2
-1
0
1

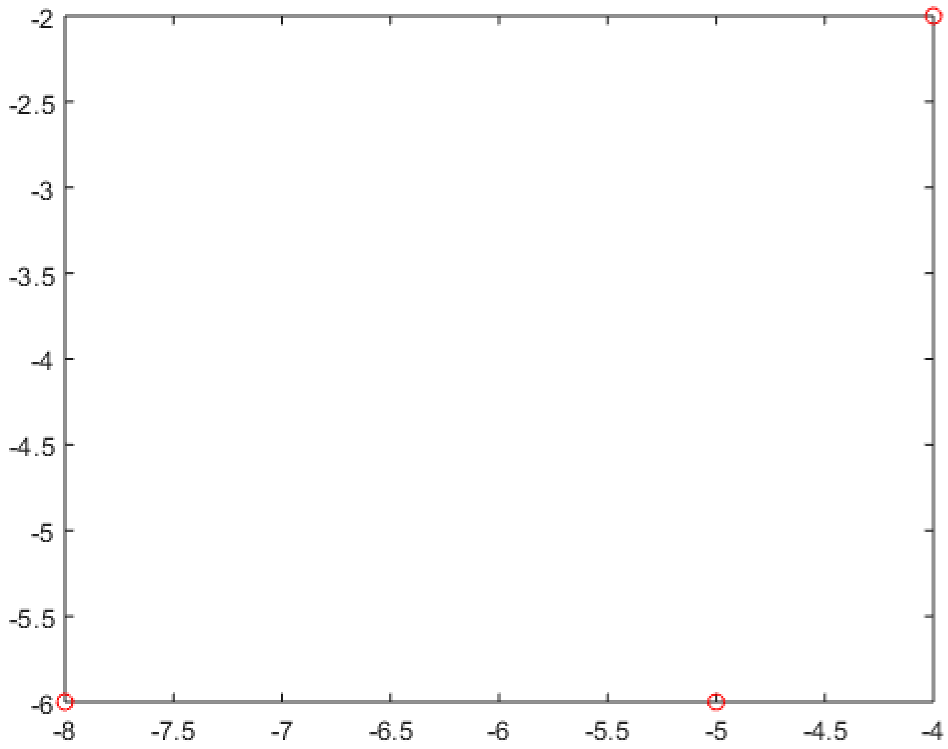

35
30
25
20
15
10
5
0
-5
-9
-8
-7
-6
-5
-4
-3
-2
-1
0

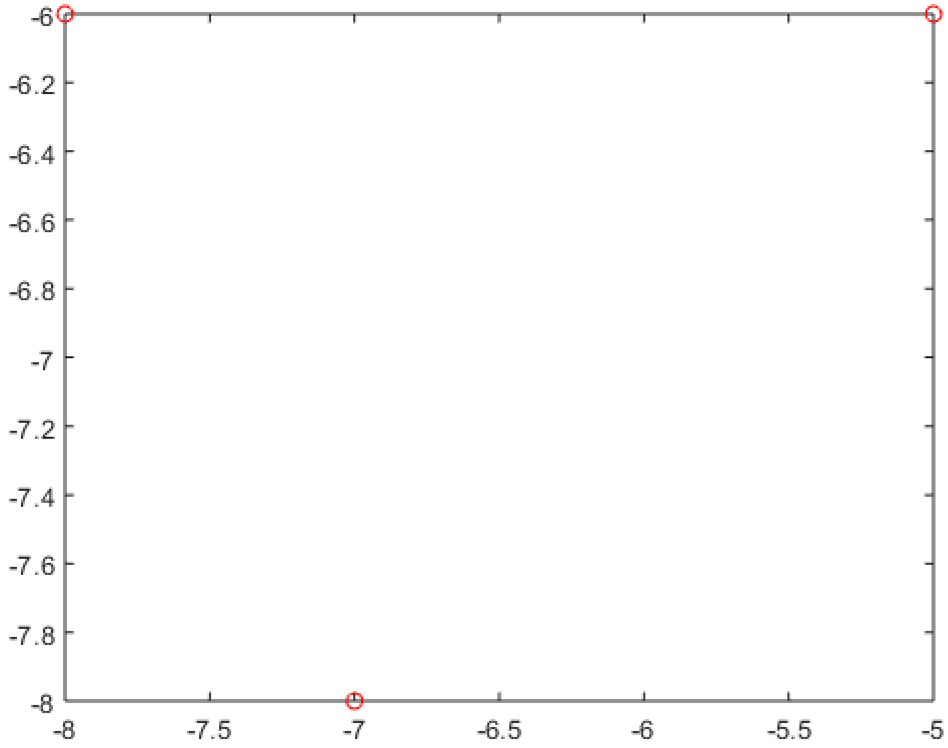

25
20
15
10
5
0
-5
-9
-8
-7
-6
-5
-4
-3
-2
-1

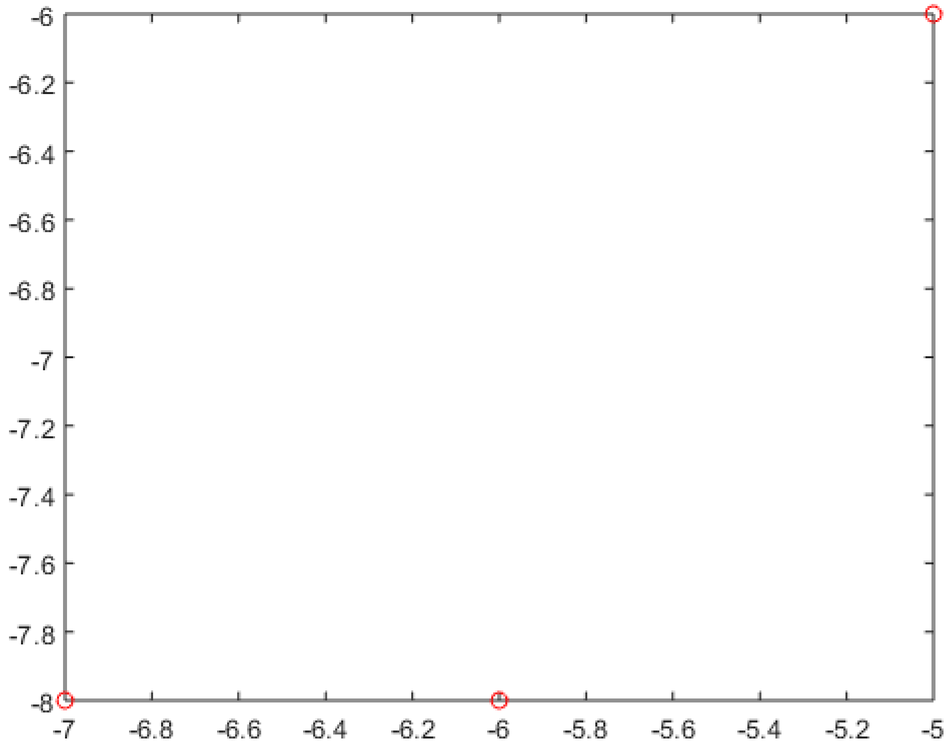

12
10
8
6
4
2
0
-2
-9
-8
-7
-6
-5
-4
-3
-2

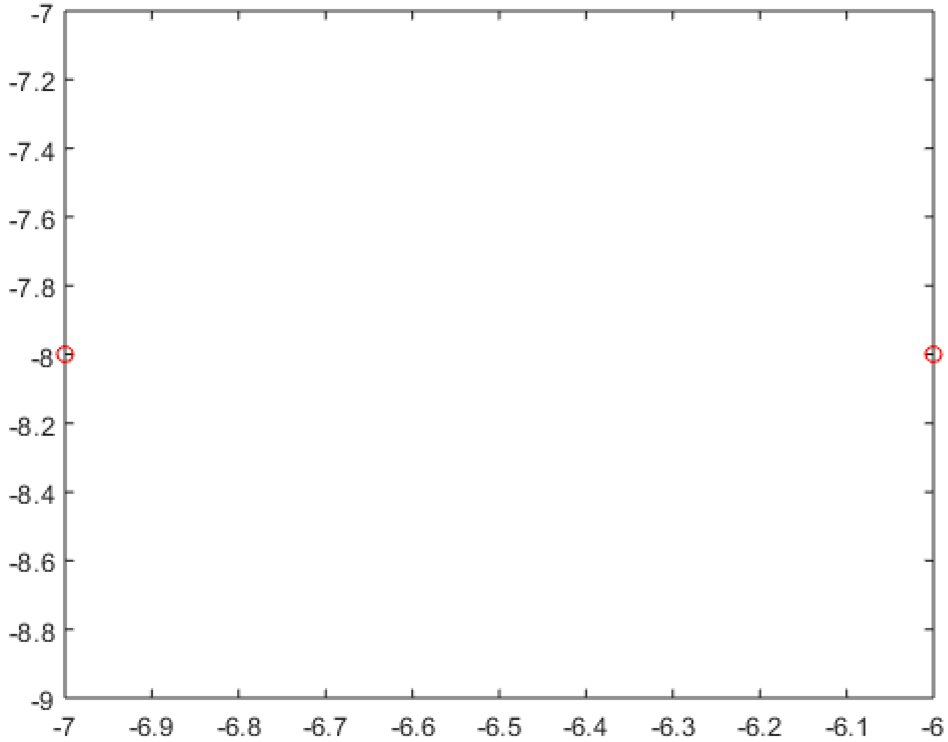

4
3
2
1
0
-1
-2
-9
-8
-7
-6
-5
-4
-3


-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
-9
-8.5
-8
-7.5
-7
-6.5
-6
-5.5
-5
-4.5
-4


-2
-2.5
-3
-3.5
-4
-4.5
-5
-5.5
-6
-8
-7.5
-7
-6.5
-6
-5.5
-5
-4.5
-4


-6
-6.2
-6.4
-6.6
-6.8
-7
-7.2
-7.4
-7.6
-7.8
-8
-8
-7.5
-7
-6.5
-6
-5.5
-5


-6
-6.2
-6.4
-6.6
-6.8
-7
-7.2
-7.4
-7.6
-7.8
-8
-7
-6.8
-6.6
-6.4
-6.2
-6
-5.8
-5.6
-5.4
-5.2
-5


-7
-7.2
-7.4
-7.6
-7.8
-8
-8.2
-8.4
-8.6
-8.8
-9
-7
-6.9
-6.8
-6.7
-6.6
-6.5
-6.4
-6.3
-6.2
-6.1
-6

COMMAND WINDOW OUTPUT

```
% % for known function y_value = 2 * (x_value - 4)^2 - 5;
% % in my sub function
% PS08_simplex_koike()
% iteration 1
% x4 - x1 = 17.000000
% iteration 2
% x4 - x1 = 16.000000
% iteration 3
% x4 - x1 = 15.000000
% iteration 4
% x4 - x1 = 14.000000
% iteration 5
% x4 - x1 = 13.000000
% iteration 6
% x4 - x1 = 12.000000
% iteration 7
% x4 - x1 = 11.000000
% iteration 8
% x4 - x1 = 10.000000
% iteration 9
% x4 - x1 = 9.000000
% iteration 10
% x4 - x1 = 8.000000
% iteration 11
% x4 - x1 = 7.000000
% iteration 12
% x4 - x1 = 6.000000
% iteration 13
% x4 - x1 = 5.000000
% iteration 14
% x4 - x1 = 4.000000
% iteration 15
% x4 - x1 = 3.000000
% iteration 16
% x4 - x1 = 2.000000
% iteration 17
% x4 - x1 = 1.000000
% iteration 18
% x4 - x1 = 0.000000
% the location of the minimum value is (4.0000, -5.0000)
%
% I am submitting code that is my own original work. I have not used
% source code, either modified or unmodified, obtained from any
% unauthorized source. Neither have I provided access to my code to
 any
% peer or unauthorized source. Signed,
% <Tomoki Koike>
%
```

```
% ans =
%
%     -5

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%

% for the unknown function in PS08_funeval.p
% PS08_simplex_koike()
% iteration 1
% x4 - x1 = 17.000000
% iteration 2
% x4 - x1 = 16.000000
% iteration 3
% x4 - x1 = 15.000000
% iteration 4
% x4 - x1 = 14.000000
% iteration 5
% x4 - x1 = 13.000000
% iteration 6
% x4 - x1 = 12.000000
% iteration 7
% x4 - x1 = 11.000000
% iteration 8
% x4 - x1 = 10.000000
% iteration 9
% x4 - x1 = 9.000000
% iteration 10
% x4 - x1 = 8.000000
% iteration 11
% x4 - x1 = 7.000000
% iteration 12
% x4 - x1 = 6.000000
% iteration 13
% x4 - x1 = 5.000000
% iteration 14
% x4 - x1 = 4.000000
% iteration 15
% x4 - x1 = 3.000000
% iteration 16
% x4 - x1 = 2.000000
% iteration 17
% x4 - x1 = 1.000000
% iteration 18
% x4 - x1 = 0.000000
% the location of the minimum value is (-6.0000, -8.0000)
%
% I am submitting code that is my own original work. I have not used
% source code, either modified or unmodified, obtained from any
% unauthorized source. Neither have I provided access to my code to
 any
% peer or unauthorized source. Signed,
% <Tomoki Koike>
%
```

```
% ans =
%
%     -8
```

ACADEMIC INTEGRITY STATEMENT

```
% Call your academic integrity statement here
PS07_academic_integrity_koike(nameArray);
```

```
I am submitting code that is my own original work. I have not used
source code, either modified or unmodified, obtained from any
unauthorized source. Neither have I provided access to my code to any
peer or unauthorized source. Signed,
<Tomoki Koike>

ans =

    -8
```

Published with MATLAB® R2018a