Tema 1
Nicoleta Radu

// Rezolvările au fost scrise în engleza din obisnuinta.
// În cazul în care nu sunt acceptate, le voi schimba de acum încolo 🙂

Exercitiul 2 / Pag 3
Rezolvare Script:

```
% Exercitiul 2: B
% REZOLVARE
clear, clc
format long
a = [3.14, 999996, 0.000009, 1.00345];
x = [3.141592, 1000000, 0.000012, 1.000145];
Measure(a,x);

% Exercitiul 2: C
% REZOLVARE
clear, clc
a = [52,101,297];
x = [50,100,300];
Measure(a,x;
% Masurarea mai exacta este: 0.00990099
```

Rezolvare Functie:

```
function [] = Measure(a,x)
%  Return absolute and relative errors
%  If "Delta(a) > 0", we say that "a" approximates "x" by shortage.
%  If "Delta(a) < 0" we say that "a" approximates "x" by addition.
if nargin > 3
   error('Error! This Function Only Takes Three Arguments.');
End

sizeA = length(a);
sizeX = length(x);

if sizeA ~= sizeX
   error('Error! Invalid Size For Arrays')
End

 deltaA = x - a;

if all(deltaA == 0)
   fprintf('deltaA is 0\n')
   return
```

```
End

absolute = abs(deltaA);
relative = absolute ./ abs(a);

i = 1;
while i <= sizeA
   fprintf('a = %.8f \n',a(i))
   fprintf('x = %.8f \n',x(i))

   fprintf('absolute = %.8f\n',absolute(i))
   fprintf('relative = %.8f\n',relative(i))

   if deltaA(i) > 0
   fprintf('a approximates x by shortage\n')
   fprintf('\n')
   else
   fprintf('a approximates x by addition\n')
   fprintf('\n')
   end
   i = i + 1;
End

[minimumAbsolute, minimumAbsolutePos] = min(absolute);
[minimumRelative, minimumRelativePos] = min(relative);

fprintf('ErrAbs = %.8f , position = %d\n',minimumAbsolute,
minimumAbsolutePos)
fprintf('ErrRel = %.8f , position = %d\n',minimumRelative,
minimumRelativePos)
End
```

Exercitiul 6 / Pag 5
Rezolvare Script:

```
% Exercitiul 6
% REZOLVARE
clear, clc
% Valorile
values = [2.416752; 6.216253; 3.454650];
% Decimalele cerute
n = [2, 3, 4, 5];
for i = 1:length(n)
   rounded_values(:, i) = roundn(values, -n(i));
```

```
   rounded_values(:, i) = round(rounded_values(:, i) * 10^n(i)) /
10^n(i);
   if mod(rounded_values(:, i) * 10^n(i), 2) == 1
       rounded_values(:, i) = rounded_values(:, i) - 1 / 10^n(i)
   end
end
```

Exercitiul 7 / Pag 5
Pe 'Hartie'

2.456750 => 2.457 (prima cifra neglijata: 7, ultima cifra pastrata: 6);
2.42629 => 2.426 (prima cifra neglijata: 2, ultima cifra pastrata: 6);
2.456752 => 2.457 (prima cifra neglijata: 7, ultima cifra pastrata: 6);
2.416512; => 2.417 (prima cifra neglijata: 5, ultima cifra pastrata:6);
2.45350 => 2.454 (prima cifra neglijata: 5, ultima cifra pastrata: 3);

Exercitiul 10 / Pag 6
// Am folosit funcția definită anterior ca sa aflam eroarea relativă pentru fiecare aproximare
// Mai departe, din pacate, nu am înțeles calcul necesar pentru aflarea cifrelor semnificative

```
clear, clc
a = [500, 499.992, 500.02, 499.979, 499.989];
x = [499.987, 499.987, 499.987, 499.987, 499.987];
Measure(a,x)
```

Output:
a = 500.00000000
x = 499.98700000
absolute = 0.01300000
relative = 0.00002600
a approximates x by addition

a = 499.99200000
x = 499.98700000
absolute = 0.00500000
relative = 0.00001000
a approximates x by addition

a = 500.02000000

x = 499.98700000
absolute = 0.03300000
relative = 0.00006600
a approximates x by addition

a = 499.97900000
x = 499.98700000
absolute = 0.00800000
relative = 0.00001600
a approximates x by shortage

a = 499.98900000
x = 499.98700000
absolute = 0.00200000
relative = 0.00000400
a approximates x by addition