Lab #6

Contents

1 Matrices

Matrices are two-dimensional arrays. Each value in a matrix is identified by a pair of numbers: row and column.

1.1 Working with matrices

Matlab syntax for an element at row `r` and column `c` of the matrix `M` is `M(r, c)`.

Examples:

```
a = [10, 20.11; 3.18, pi];

a
a(1, 1) % element at the first row and first column
a(2, 1) % element at the second row and first column
a(end, 1) % element at the last row and first column
```

```
a(1:2, 1) % first and second rows of the matrix a and the first column
a(1, :) % first row of the matrix a
a(:, 2) % second column of the matrix a

a(2, 2) = -2; % changing value at the second row and second column
              % to -2
a(2, :) = a(2, :) + 3; % add 3 to all elements in the second row
a(:, 1) = a(:, 1) - 1; % add -1 to all elements in the first column
a

size(a) % returns size of the matrix
size(a, 1) % returns the number of rows in the matrix
size(a, 2) % returns the number of columns in the matrix
```

1.2 Matrix operations

The same as for vectors, see Lab 2.

```
% change all elements of the matrix a
a = a + 10
a = a - 10
a = a * 10
a = a / 10

matrix1 = [10, 20; 30, 40];
matrix2 = [30, 20; 10, 66];

% element-wise operations
matrix1 + matrix2
matrix1 - matrix2
matrix1 .* matrix2
matrix1 ./ matrix2

% matrix concatenation
[matrix1, matrix2]
```

```
18
19 % concatenation again
20 [matrix1; matrix2]
21
22 % transpose
23 a'
24 a''
```

1.3 Matrix creation

```
1 zeros(5, 10)
2 ones(6, 10)
3 rand(7, 10)
```

1.4 Iterating through all elements of the matrix

In order to work with elements of the matrix one-by-one, we need to use nested loops.

1.4.1 Print each element separately

In the following example we print each element separately:

```
1  clear;
2
3  M = rand(4, 4);
4
5  number_of_rows = size(M, 1);
6  number_of_columns = size(M, 2);
7
8  % for each row
9  for row = 1:number_of_rows
10     % for each column
11     for column = 1:number_of_columns
12         disp(M(row, column)); % print value
13     end
14 end
```

1.4.2 Adding 2 to each element

In the following example we add 2 to each element:

```
clear;

M = zeros(4, 4);

number_of_rows = size(M, 1);
number_of_columns = size(M, 2);

% for each row
for row = 1:number_of_rows
    % for each column
    for column = 1:number_of_columns
        M(row, column) = M(row, column) + 2; % add 2
    end
end
```

1.4.3 Finding the sum of all elements

In the following example we are calculating the sum of all elements in the matrix:

```
clear;

M = [1, 2, 1; 4, 5, 2; 1, 3, 2];
sum_of_elements = 0;

number_of_rows = size(M, 1);
number_of_columns = size(M, 2);

% for each row
for row = 1:number_of_rows
    % for each column
    for column = 1:number_of_columns
        sum_of_elements = sum_of_elements + M(row,
           column);
    end
end

```

```
disp(sum_of_elements);
```

1.4.4 Finding max element

In the following example we are looking for the largest element:

```
clear;

M = [1, 2, 1; 4, 5, 2; 1, 3, 2];
maximum = -9999999999; % we can use -Inf instead

number_of_rows = size(M, 1);
number_of_columns = size(M, 2);

% for each row
for row = 1:number_of_rows
    % for each column
    for column = 1:number_of_columns
        if M(row, column) > maximum
            maximum = M(row, column);
        end
    end
end

disp(maximum);
```

1.4.5 Print matrix line by line

In the following example we are printing the matrix one row at a time:

```
clear;

M = rand(4, 4) ;
number_of_rows = size(M, 1) ;

% for each row
for row = 1:number_of_rows
    disp(M(row, :)) ; % print value
end
```

2 Task 11

- Create a new 3x3 matrix `M` filled with random numbers (use `rand(3, 3)` function).
- Display the element at row 2 column 3
- Display the element at row 3 column 1
- Display the whole row 2
- Display the whole column 3
- Change the value of the element at row 2 column 3 to 100
- Change the value of the element at row 3 column 3 to 200
- Print each column of the matrix separately (Hint: try to modify the "Print matrix line by line" example. The number of columns in the matrix is `size(M, 2)`).
- Add a random number to each element of the matrix (Hint: use nested for loops. The number of rows is `size(M, 1)`, the number of columns is `size(M, 2)`. The random number can be generated by using `rand()` function without any parameters, i.e. nothing inside parenthesis.)