TNSL20 - Matlab Test

This is the final test for the MATLAB part of TNSL20 course.

You are allowed to use:

- All lab material

- Previously written code

- Internet

Not allowed:

- Communication of any kind (email, chat, discussion, phones or other mobile devices, etc.)

- Information from your neighbour's display

The exam goes during the whole class without the break. Late submissions will not be graded.
All mobile devices should be off and away.
To pass the Matlab part of the course you need to correctly solve at least 2 tasks.
Please, submit all your MATLAB codes (.m files) to your teacher in a single email.

Teachers' emails:
Leonid Sedov: leonid.sedov@liu.se
Anastasia Lemetti: anastasia.lemetti@liu.se

Good luck!

1 Task 1

Create a program that for a given vector `v` computes the product of all negative numbers of `v`. Make sure that your code works with any `v`.

For example, for the following vector `v`:

```
1  v = [-65, 10, 101, 0, -1, 55, -21, -200];
```

The product should be 273000.

2 Task 2

Create a program that for a given vector `v` prints all pairs of elements from this vector where the first number is greater than the second number. Make sure that your program works with any `v`.

For example, for the vector `v = [1, 4, 2, 9]` the program output should be (order of the rows does not matter):

```
1  4 1
2  4 2
3  2 1
4  9 1
5  9 4
6  9 2
```

3 Task 3

Write a program that creates a vector with 10 random numbers (using `rand()` or `randi()` function) and finds the maximum element in the vector.

4 Task 4

Write a program that for a given matrix `A` creates a new matrix `B` of the same size with elements from `A` divided by 10. Make sure that your code works with any matrix `A`.

For example, for the input matrix

```
1  A = [
2      10, 20, 40;
```

```
3        30, 10, 50;
4        90, 60, 70
5    ];
```

The resulting matrix `B` should be:

```
1    B = [
2        1, 2, 4;
3        3, 1, 5;
4        9, 6, 7
5    ];
```

5   Task 5

Create a program that for a given matrix `A` prints all columns of `A`, separating columns with a line of symbols "#". Make sure that your program works with any matrix `A`.

For example, for the input matrix

```
1    A = [
2        10, 20, 21;
3        30, 10, 56;
4        50, 60, 75
5    ];
```

The program output should be:

```
1        10
2        30
3        50
4
5    ################
6        20
7        10
8        60
9
10   ################
11       21
12       56
13       75
```