1 Matrix

1.1 Make a matrix $A$, which exists of three rows of length 10. The first row goes from 1 to 10 with a step of 1. The second row is filled with random values. The third row is equally distributed from value 1 to 3.

$$A =
\begin{bmatrix}
1.0000 & 2.0000 & 3.0000 & 4.0000 & 5.0000 & 6.0000 & 7.0000 & 8.0000 & 9.0000 & 10.0000 \\
0.0596 & 0.6820 & 0.0424 & 0.0714 & 0.5216 & 0.0967 & 0.8181 & 0.8175 & 0.7224 & 0.1499 \\
1.0000 & 1.2222 & 1.4444 & 1.6667 & 1.8889 & 2.1111 & 2.3333 & 2.5556 & 2.7778 & 3.0000
\end{bmatrix}$$

1.2 Make a matrix $A$ (5x5) with random integers between 1 and 10 (inclusive). Make a matrix $B$ which shows which values of $A$ are greater than 5. Make a matrix $C$ which has 0 value for all values of matrix $A$ that are equal or smaller than 5 by using matrix $B$.

$$A =
\begin{bmatrix}
3 & 4 & 5 & 1 & 4 \\
5 & 9 & 9 & 7 & 1 \\
6 & 5 & 5 & 1 & 1 \\
8 & 9 & 10 & 2 & 2
\end{bmatrix}$$

$$B =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 1 & 1 & 0 & 0
\end{bmatrix}$$

$$C =
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 \\
0 & 9 & 9 & 7 & 0 \\
6 & 0 & 0 & 0 & 0 \\
8 & 9 & 10 & 0 & 0
\end{bmatrix}$$
1.3 Make a matrix $A$ (8x8) filled with zeros. Make a matrix $B$ (4x4) filled with random integer of 1 up to 3. Copy $B$ into $A$. $B$ must start at the second row and third column of $A$. 

$A =$

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]

$B =$

\[
\begin{array}{cccc}
2 & 2 & 1 & 2 \\
2 & 2 & 1 & 3 \\
3 & 1 & 3 & 2 \\
3 & 2 & 1 & 3 \\
\end{array}
\]

$A =$

\[
\begin{array}{cccccccc}
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 2 & 2 & 1 & 2 & 0 & 0 \\
0 & 0 & 2 & 2 & 1 & 3 & 0 & 0 \\
0 & 0 & 3 & 1 & 3 & 2 & 0 & 0 \\
0 & 0 & 3 & 2 & 1 & 3 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\end{array}
\]
1.4 Make a matrix $A$ (5x5) filled with random integers from 1 up to 99. Find the index of even numbers. Find the values on even position.

$$A = \begin{bmatrix}
77 & 88 & 75 & 9 & 19 \\
17 & 59 & 82 & 12 & 50 \\
86 & 16 & 79 & 14 & 15 \\
98 & 20 & 32 & 68 & 6 \\
51 & 41 & 53 & 50 & 85
\end{bmatrix}$$

Index of even numbers =

2 4 6 8 10 12 14 16 18 20 22 24

Numbers on even index =

17 98 88 16 41 82 32 9 14 50 50 6
2 Functions

2.1 Pythagoras

A) Write a function that calculates the oblique of Pythagoras. The arguments are side A and B.

2.2 Percentage

A) Write a function that calculates the percentage of a number. The arguments are the number and the percentage of it to be found.

2.1 Hailstone

A) Implement a function that takes one argument and returns one vector.

B) Inside the function, create a while loop which continues if the argument (here called n) is greater than 1.

C) At each iteration if n is odd n becomes 3 * n + 1. If it even n becomes n / 2. Each iteration adds n to the vector.

2.2 Function parameters

A) Make a function that takes 3 arguments. The first two arguments are numbers. The third argument is a character representing multiplication, addition, subtraction or division (*, +, -, /). The operation applied to the two numbers depends on the third argument. If the third argument does not match our expectations (e.g is @, è, p, ...) then an error is thrown.

2.3 Pi

If a circle has the same diameter than the length of a square, then the ratio of the area of the circle to the area of the square is PI/4. If you put the circle inside the square and select random points inside the square, the number of points inside the circle divided the number of points inside the square would approximate PI/4. This technique of calculating PI is called Monte Carlo Simulation.

A) Create a function that implements Monte Carlo. The function should have one argument which is the number of points.
B) Make a meaningful plot of the evolution of PI in function of the number of points. Preferably $10^1$, $10^2$, $10^3$, $10^4$ and $10^5$ points.

2.4 Geometry

A) Create a function that calculates the area of a square. This function has one argument.

B) Create a function that calculates the area of a rectangle. This function has two arguments.

B) Create a function that encapsulates both functions. This super function should be able to work with one or two arguments. If this function in called with one argument it will return the area of a square. If this function is called with two arguments it will return the area of a rectangle.

2.5 Benchmark

A) Compare the execution time of functions with and without pre-allocation. Start by comparing the speed of two quadratic functions by generating $10^1$, $10^2$, $10^3$, $10^4$ and $10^5$ numbers. Make a meaningful plot as shown in the figure below. Do not forget that a good benchmark takes the average of multiple samples.

B) Ensure that the code is well written. This means that you must have 4 different scripts: quadratic function, quadratic function with pre-allocation, benchmark function and the main script file in which the data is plotted.
C) Repeat the first step by replacing quadratic with fibonacci.

C) Repeat the first step by replacing quadratic with hailstone.

D) Make one figure which has 3 separated plots. The first plot is quadratic, second is fibonacci and the third one is hailstone.

2.6 Circle
A) Make a function that draws a circle. The function has 4 arguments: x position and y position of the center of the circle, radius r and precision p.