Exercise 1: **Probability Density Functions and Monte Carlo generators**

Toni Šćulac (Faculty of Science, University of Split, Croatia)

Problem 1

Draw the Gaussian distribution for three different sets of values for the mean and standard deviation parameters (μ, σ) and explain how μ and σ parameters affect its shape.
If a newly discovered particle X is described by a Gaussian distribution $Gauss(\mu = 200\text{GeV}, \sigma = 2\text{GeV})$, calculate the following probabilities:

- to produce particle X with mass of 205 GeV or more;
- to independently produce two particles X with masses above 203 GeV;

Problem 2

Let's make sure that the Central Limit Theorem works even with the exponential PDF $E(x;\lambda) = \lambda e^{-\lambda x}$. Generate a random number x which is defined as $x = \frac{1}{3}(x_1 + x_2 + x_3)$, where x_i is a random number drawn from $E(x;\lambda_i)$. Check the distribution of x after generating a large number of random events. Does it look like a Gaussian? Now redo the Monte Carlo experiment by defining x as $x = \frac{1}{N}\sum_{i=1}^{N} x_i$ and find N for which the resulting distribution looks like a Gaussian.

Problem 3

Transform already available uniform random number generator to draw random numbers according to a given PDF distribution. Let the user define a PDF function it wants random numbers to be generated according to. Generate a pair of random numbers (x, y) and if the value $PDF(x) > y$ keep the value x as the generated random number, otherwise discard it. This method is called the acceptance-rejection method.

Problem 4

Test your random number generator from Problem 3 to generate random numbers according to the normal distribution $Gauss(\mu = 0, \sigma = 1)$. Generate 10^4 events and draw them in a histogram. Draw a normal distribution on top of the histogram. How many random numbers did you have to generate to obtain the 10^4 random numbers distributed according to the given PDF?

Problem 5*

Start by writing a code that derives and draws a Cumulative Density Function (CDF) for any PDF that user gives it. Test the funcionality of your code on the normal distribution $Gauss(\mu = 0, \sigma = 1)$. Now that the CDF derivation is ready, the next step is to implement the inversion method to generate random numbers according to the given PDF. Generate a random number u, then compute the value x for which $CDF(x) = u$. Take the calculated number x to be randomly drawn from the given PDF distribution. Test your random number generator using the normal distribution $Gauss(\mu = 0, \sigma = 1)$.