Homework no. 1

1. Find the smallest positive number $\boldsymbol{u > 0}$, written as a negative power of 10, $\boldsymbol{u = 10^{-m}}$, where $\boldsymbol{m} \in \mathbb{N}$, which satisfies the property:

   $$\mathbf{1} +_c \boldsymbol{u} \neq \mathbf{1}$$

   In the above relation, $+_c$ denotes the computer-implemented addition operation. The number $\boldsymbol{u}$ is known as ***machine precision***.

2. Operation $+_c$ is ***non-associative***: consider the numbers $\boldsymbol{x=1.0}$, $\boldsymbol{y = u/10}$, $\boldsymbol{z = u/10}$, where $\boldsymbol{u}$ is the above-computed machine precision (the value that satisfies the relations $\mathbf{1} +_c \boldsymbol{u} \neq \mathbf{1}$ and $\mathbf{1} +_c \boldsymbol{u}/\mathbf{10} = \mathbf{1}$). Verify that the computer addition operation is non-associative.

   $$(\boldsymbol{x} +_c \boldsymbol{y}) +_c \boldsymbol{z} \neq \boldsymbol{x} +_c (\boldsymbol{y} +_c \boldsymbol{z}).$$

   Find an example that shows the computer multiplication operation is also non-associative.

3. **Polynomial approximations for the *sin* function**
   Consider the polynomials:

   $$P_1(x) = x - c_1x^3 + c_2x^5$$
   $$P_2(x) = x - c_1x^3 + c_2x^5 - c_3x^7$$
   $$P_3(x) = x - c_1x^3 + c_2x^5 - c_3x^7 + c_4x^9$$
   $$P_4(x) = x - 0.166x^3 + 0.00833x^5 - c_3x^7 + c_4x^9$$
   $$P_5(x) = x - 0.1666x^3 + 0.008333x^5 - c_3x^7 + c_4x^9$$
   $$P_6(x) = x - 0.16666x^3 + 0.0083333x^5 - c_3x^7 + c_4x^9$$
   $$P_7(x) = x - c_1x^3 + c_2x^5 - c_3x^7 + c_4x^9 - c_5x^{11}$$
   $$P_8(x) = x - c_1x^3 + c_2x^5 - c_3x^7 + c_4x^9 - c_5x^{11} + c_6x^{13}$$

   where the constants $\boldsymbol{c_i}$ have the following values:

$$c_1 = \frac{1}{3!} = 0.16666666666666666666666666666667$$

$$c_2 = \frac{1}{5!} = 0.0083333333333333333333333333333333$$

$$c_3 = \frac{1}{7!} = 1.984126984126984126984126984127e\text{-}4$$

$$c_4 = \frac{1}{9!} = 2.7557319223985890652557319223986e\text{-}6$$

$$c_5 = \frac{1}{11!} = 2.5052108385441718775052108385442e\text{-}8$$

$$c_6 = \frac{1}{13!} = 1.6059043836821614599392377170155e\text{-}10$$

All the above polynomials can be used to approximate the ***sin*** function for $x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$***:***

$$\sin(x) \approx P_i(x) \quad , \quad \forall\, x \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right].$$

Generate 10.000 random numbers in interval $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ and compute the values of those 8 above polynomials in these points. Consider that the value of the ***sin*** function computed using the mathematical library of the programming language you are employing (math.sin – Python, Math.sin – Java , Math.Sin – C#) is the exact value of the ***sin*** function, i.e.

$$v_{exact} = \sin(x) = Math.\sin(x).$$

For each of the 10.000 generated numbers save three polynomials that provided the best approximations (those polynomials that provided the smallest approximation errors).

$$\boldsymbol{error_i\,(x) = |\,P_i(x) - v_{exact}\,|.}$$

Compute a top for the 8 polynomials, taking into account these results.

Implement the computations of the 8 polynomials such that it minimizes the number of elementary operations (additions, subtractions, multiplications, divisions). For example, for polynomial $\boldsymbol{P_2}$ we can use the following relation in order to make as few elementary operations as possible:

$$P_2(x) = x\left(1 + y\left(-c_1 + y\left(c_2 - c_3 y\right)\right)\right) \text{ where } y = x^2$$

The 6 constants c_i will be declared as such in your program, or they will be computed only once.

Bonus 5pt: display the computing time for each of the 8 polynomials using the same 10.000 values generated from $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Display in increasing order these 8 computing times (and the number of the corresponding polynomial).