Tema nr. 1

1. Să se găsească cel mai mic număr pozitiv $u > 0$, de forma $u = 10^{-m}$ care satisface proprietatea:

$$1.0 +_c u \neq 1.0$$

unde prin $+_c$ am notat operaţia de adunare efectuată de calculator. Numărul u se numeşte ***precizia maşină***.

2. Operaţia $+_c$ este ***neasociativă***: fie numerele $x=1.0$, $y = u$, $z = u$, unde u este precizia maşină calculată anterior. Să se verifice că operaţia de adunare efectuată de calculator este neasociativă:

$$(x +_c y) +_c z \neq x +_c (y +_c z)$$

Să se găsească un exemplu pentru care operația de înmulțire $\times_c$ este neasociativă.

3. **Aproximări polinomiale ale funcţiei *sin***

Fie polinoamele:

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

unde constantele c_i au următoarele valori:

$$c_1 = \frac{1}{3!} = 0.16666666666666666666666666666667$$
$$c_2 = \frac{1}{5!} = 0.00833333333333333333333333333333$$
$$c_3 = \frac{1}{7!} = 1.984126984126984126984126984127e\text{-}4$$
$$c_4 = \frac{1}{9!} = 2.7557319223985890652557319223986e\text{-}6$$
$$c_5 = \frac{1}{11!} = 2.5052108385441718775052108385442e\text{-}8$$
$$c_6 = \frac{1}{13!} = 1.6059043836821614599392377170155e\text{-}10$$

Toate polinoamele de mai sus sunt aproximări ale funcției ***sin*** pentru ***x∈[-π/2,π/2]***

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

Să se genereze 10.000 de numere aleatoare din intervalul $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ şi să se calculeze valorile celor 6 polinoame de mai sus în aceste puncte. Se consideră că valoarea funcției ***sin*** calculată de biblioteca matematică a programului cu care lucrați (Math.sin – Java , Math.Sin – C#, math.sin - Python) este valoarea exactă a funcției ***sin***.

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

Pentru fiecare din cele 10.000 de numere generate să se memoreze cele trei polinoame care au furnizat cele mai bune aproximări (acele polinoame care furnizează cele mai mici erori).

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

În funcție de aceste rezultate, să se facă o ierarhie a celor 6 polinoame.

Să se implementeze modul de calcul al celor 6 polinoame astfel încât să se facă un număr minim/cât mai mic de operații elementare (adunări, scăderi, înmulțiri, împărțiri). De exemplu, pentru polinomul $\boldsymbol{P_2}$ putem folosi următoarea grupare a termenilor:

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

Cele 6 constante $\boldsymbol{c_i}$ vor fi declarate ca atare în program, fie vor fi calculate o singură dată.

Bonus: să se calculeze timpii de lucru ale aceloraşi 100.000 de valori din $\left[-\frac{\pi}{2}, \frac{\pi}{2}\right]$ pentru fiecare din cele 6 polinoame.