ESCUELA POLITÉCNICA NACIONAL
MÉTODOS NUMÉRICOS

Leandro Bravo GR1CC

*** Resuelva los siguientes ejercicios, tome en cuenta que debe mostrar el desarrollo completo del ejercicio**

1. Calcule los errores absoluto y relativo en las aproximaciones de p por p^*

   a. $p = \pi, p* = 22/7$ b. $p = \pi, p* = 3.1416$

   c. $p = e, p* = 2.718$ d. $p = \sqrt{2}, p* = 1.414$

- Calcule los errores absoluto y relativo en las aproximaciones de p por p^*

a. $p = e^{10}, p* = 22000$ b. $p = 10^{\pi}, p* = 1400$

c. $p = 8!, p* = 39900$ d. $p = 9!, p* = \sqrt{18\pi}(9/e)^9$

```python
def errorAbsoluto(valores):
    errores = []
    for p, p_aprox in valores:
        errores.append(abs(p - p_aprox))
    return errores
```

```python
def errorRelativo(valores):
    errores = []
    for p, p_aprox in valores:
        errores.append(abs((p - p_aprox) / p))
    return errores
```

```python
import math
valores = [(math.pi, 22/7), (math.pi, 3.1416), (math.e, 2.718), (math.sqrt(2), 1.414),
           (math.e**10, 22000), (10**math.pi, 1400), (math.factorial(8), 39900),
           (math.factorial(9), math.sqrt(18*math.pi)*(9/math.e)**9)]

errores_absolutos = errorAbsoluto(valores)
errores_relativos = errorRelativo(valores)

for i in range(len(valores)):
    print(f"Para p = {valores[i][0]}, p* = {valores[i][1]}:")
    print(f"El error absoluto es {errores_absolutos[i]}")
    print(f"El error relativo es {errores_relativos[i]}")
```

```
Para p = 3.141592653589793, p* = 3.142857142857143:
El error absoluto es 0.0012644892673496777
El error relativo es 0.0004024994347707008
Para p = 3.141592653589793, p* = 3.1416:
El error absoluto es 7.346410206832132e-06
El error relativo es 2.3384349967961744e-06
Para p = 2.718281828459045, p* = 2.718:
El error absoluto es 0.0002818284590451192
El error relativo es 0.00010367889601972718
Para p = 1.4142135623730951, p* = 1.414:
El error absoluto es 0.00021356237309522186
El error relativo es 0.00015101140222192286
Para p = 22026.465794806703, p* = 22000:
El error absoluto es 26.465794806703343
El error relativo es 0.0012015452253326688
Para p = 1385.4557313670107, p* = 1400:
El error absoluto es 14.544268632989315
El error relativo es 0.010497822704619136
Para p = 40320, p* = 39900:
El error absoluto es 420
El error relativo es 0.010416666666666666
Para p = 362880, p* = 359536.87284194835:
El error absoluto es 3343.1271580516477
El error relativo es 0.00921276223080598
```

3. Encuentre el intervalo más largo en el que se debe encontrar $p*$ para aproximarse a p con error relativo máximo de 10^{-4} para cada valor de p

$$\text{a.}\pi \qquad \text{b. } e$$
$$\text{c.}\sqrt{2} \qquad \text{d. } \sqrt[3]{7}$$

```
def calcular_intervaloMasLargo(p):
    error_maximo = 10**-4
    delta = p * error_maximo
    return (p - delta, p + delta)
```

```
import math
valores_p = [math.pi, math.e, math.sqrt(2), 7**(1/3)]
intervalos = {p: calcular_intervaloMasLargo(p) for p in valores_p}
for p, intervalo in intervalos.items():
    print(f"Para p = {p}, el intervalo es {intervalo}")
```

```
Para p = 3.141592653589793, el intervalo es (3.141278494324434, 3.141906812855152)
Para p = 2.718281828459045, el intervalo es (2.718010000276199, 2.718553656641891)
Para p = 1.4142135623730951, el intervalo es (1.4140721410168577, 1.4143549837293325)
Para p = 1.912931182772389, el intervalo es (1.9127398896541117, 1.9131224758906662)
```

4. Use la aritmética de redondeo de tres dígitos para realizar lo siguiente. Calcule los errores absoluto y relativo con el valor exacto determinado para por lo menos cinco dígitos

$$\text{a.}\frac{\frac{13}{14}-\frac{5}{7}}{2e-5.4} \qquad \text{b. } -10\pi+6e-\frac{3}{61}$$
$$\text{c.}\left(\frac{2}{9}\right)\cdot\left(\frac{9}{11}\right) \qquad \text{d. } \frac{\sqrt{13}+\sqrt{11}}{\sqrt{13}-\sqrt{11}}$$

```
def redondeo(num:float, digitos:int)->float:
    return float(f"{num:.{digitos - 1}e}")
```

```
def err_Abs_5dig(valores):
    return list(map(lambda x: abs(redondeo(x[0]-x[1],5)),valores))
```

```
def err_Rel_5dig(valores):
    return list(map(lambda x: redondeo(abs(redondeo(x[0]-x[1],5))/redondeo(x[0],5),5),valores))
```

```
a_exp = redondeo(redondeo((redondeo(13/14,3)-redondeo(5/7,3)),3)/(redondeo(2*math.e,3)-5.4),3)
b_exp = redondeo(redondeo(-10*math.pi,3)+redondeo(6*math.e,3)-redondeo(3/61,3),3)
c_exp = redondeo(redondeo(2/9,3)*redondeo(9/11,3),3)
d_exp = redondeo(redondeo(redondeo(13**0.5,3)+
                          redondeo(11**0.5,3),3)/redondeo(redondeo(13**0.5,3)-redondeo(11**0.5,3),3),3)
valores = [((((13/14)-(5/7))/(2*math.e-5.4)),a_exp),(-10*math.pi+6*math.e-(3/61),b_exp),
           ((2/9)*(9/11),c_exp), ((13**0.5+11**0.5)/(13**0.5-11**0.5),d_exp)]
literal = 97
errores_absolutos = err_Abs_5dig(valores)
print("\nERRORES ABSOLUTOS:\n")
for errAbs in errores_absolutos:
    print("Ejercicio " + chr(literal) + ": " + str(errAbs))
    literal += 1
literal=97
print("\nERRORES RELATIVOS:\n")
errores_relativos = err_Rel_5dig(valores)
for errRel in errores_relativos:
    print("Ejercicio " + chr(literal) + ": "+ str(errRel))
    literal += 1
```

```
ERRORES ABSOLUTOS:

Ejercicio a: 0.49062
Ejercicio b: 0.055416
Ejercicio c: 0.00018182
Ejercicio d: 0.058261

ERRORES RELATIVOS:

Ejercicio a: 0.083715
Ejercicio b: -0.0036566
Ejercicio c: 0.001
Ejercicio d: 0.0024318
```

5. Los primeros tres términos diferentes a cero de la serie de Maclaurin para la función arcotangente son: $x - \left(\frac{1}{3}\right)x^3 + \left(\frac{1}{5}\right)x^5$. Calcule los errores absoluto y relativo en las siguientes aproximaciones de π mediante el polinomio en lugar del arcotangente:

$$\text{a.}4\left[arctan\left(\frac{1}{2}\right) + arctan\left(\frac{1}{3}\right)\right]$$

$$\text{b.}16 \cdot arctan\left(\frac{1}{5}\right) - 4 \cdot arctan\left(\frac{1}{239}\right)$$

```
def arctan_Serie(x):
return x-(1/3)*x**3+(1/5)*x**5
```

```
a_teo = 4*(math.atan(1/2)+math.atan(1/3))
b_teo = 16*math.atan(1/5)-4*math.atan(1/239)
a_exp = 4*(arctan_Serie(1/2)+arctan_Serie(1/3))
b_exp = 16*arctan_Serie(1/5)-4*arctan_Serie(1/239)
valores = [(a_teo,a_exp),(b_teo,b_exp)]
literal = 97
error_absoluto = errorAbsoluto(valores)
print("\nERRORES ABSOLUTOS:\n")
for errAbs in errores_absolutos:
print("Ejercicio " + chr(literal) + ": " + str(errAbs))
literal += 1
literal=97
print("\nERRORES RELATIVOS:\n")
errores_relativos = errorRelativo(valores)
for errRel in errores_relativos:
print("Ejercicio " + chr(literal) + ": "+ str(errRel))
literal += 1
```

```
ERRORES ABSOLUTOS:

Ejercicio a: 0.49062
Ejercicio b: 0.055416
Ejercicio c: 0.00018182
Ejercicio d: 0.058261

ERRORES RELATIVOS:

Ejercicio a: 0.0012679804598147663
Ejercicio b: 9.032277054963067e-06
```

6. El número e se puede definir por medio e $e = \sum_{n=0}^{\infty}\left(\frac{1}{n!}\right)$, donde $n! = n(n-1)...2 \cdot 1$ para $n \neq 0$ y $0! = 1$.

Calcule los errores absoluto y relativo en la siguiente aproximacion de e:

a. $\sum_{n=0}^{5}\left(\frac{1}{n!}\right)$ b. $\sum_{n=0}^{10}\left(\frac{1}{n!}\right)$

```
def e_serie(n):
sum=0
for i in range(n):
sum += (1/math.factorial(i))
return sum
```

```
valor_teo = math.e
a_exp = e_serie(5)
b_exp = e_serie(10)
valores = [(valor_teo,a_exp),(valor_teo,b_exp)]
literal = 97
errores_absolutos = errorAbsoluto(valores)
print("\nERRORES ABSOLUTOS:\n")
for errAbs in errores_absolutos:
print("Ejercicio " + chr(literal) + ": " + str(errAbs))
literal += 1
literal=97
print("\nERRORES RELATIVOS:\n")
errores_relativos = errorRelativo(valores)
for errRel in errores_relativos:
print("Ejercicio " + chr(literal) + ": "+ str(errRel))
literal += 1
```

```
ERRORES ABSOLUTOS:

Ejercicio a: 0.009948495125712054
Ejercicio b: 3.0288585284310443e-07

ERRORES RELATIVOS:

Ejercicio a: 0.003659846827343768
Ejercicio b: 1.1142547828265698e-07
```

7. Suponga que dos puntos (x_0, y_0) y (x_1, y_1) se encuentran en línea recta con $y_1 \neq y_0$. Existen dos fórmulas para encontrar la intersección x de la línea:

$$x = \frac{x_0 y_1 - x_1 y_0}{y_1 - y_0}$$ y $ x=x_{0}-$ |

a. Use los datos $(x_0, y_0) = (1.31, 3.24)$ y $(x_1, y_1) = (1.93, 5.76)$ y la aritmética de redondeo de tres dígitos para calcular la intersección con x de ambas maneras. ¿Cuál método es mejor y por qué?

```
def x_1ra_Aprox(P_0, P_1):
num = redondeo(P_0[0]*P_1[1],3)-redondeo(P_1[0]*P_0[1],3)
den = redondeo(P_1[1]-P_0[1],3)
return num/den
```

```
def x_2da_Aprox(P_0, P_1):
x_0 = redondeo(P_0[0],3)
num = redondeo(redondeo(P_1[0]-P_0[0],3)*redondeo(P_0[1],3),3)
den = redondeo(P_1[1]-P_0[1],3)
return x_0-num/den
```

```
P_0 = (1.31, 3.24)
P_1 = (1.93, 5.76)
print("INTERSECCIÓN CON X:\n")
print("Primera aproximación: "+str(redondeo(x_1ra_Aprox(P_0, P_1),3)))
print("Segunda aproximación: "+str(redondeo(x_2da_Aprox(P_0, P_1),3)))
```

```
INTERSECCIÓN CON X:

Primera aproximación: 0.516
Segunda aproximación: 0.512
```